

Todor Dimov



INFORMATIKA për vitin II (të dytë) të arsimit të mesëm - gjimnaz

Me Vendimin për miratimin dhe përdorimin e tekstit shkollor për lëndën mësimore Informatikë për vitin II (të dytë) të arsimit të mesëm gjimnaz nr. 26-843/1 të datës 22.07.2026, miratuar nga Komisioni Nacional për Tekste Shkollore.

Shkup, 2026

INFORMATIKË

për vitin II (të dytë) të arsimit të mesëm - gjimnaz

назив на оригинал:

Информатика

за II (втора) година гимназиско образование

Botues:

DPTU „STUDENTSKI SERVIS“ SH.P.K. – Shkup

Autor:

Todor Dimov

Recensentë

Prof. Dr. Marija Kalendar

Prof. Dr. Goran Jakimovski

Jaklina Miloševa

Redaktore:

Simona Ristovska

Lektor:

Elizabeta Poljoska

Përkthim nga gjuha maqedonase në gjuhën shqipe:

Ajnur Metaj

Dizajni dhe përgatitja teknike:

SHPTSH „STUDENTSKI SERVIS“ SHPK – Shkup

Shtypur nga:

SHPTSH „STUDENTSKI SERVIS“ SHPK

Rr. „Pero Nakov“ nr. 112 – Shkup

Tirazhi:

2000 ekzemplarë

CIP - Каталогизација во публикација

Национална и универзитетска библиотека «Св. Климент Охридски», Скопје

004(075.3)=18

DIMOV, Todor

Informatika për vitin II (të dytë) të arsimit të mesëm - gjimnaz /
Todor Dimov ; [përkthim nga gjuha maqedonase në gjuhën shqipe Ajnur
Metaj]. - Shkup : Studentski servis, 2026. - 330 стр. : илустр. ; 28 см

Превод на делото: Информатика за II (втора) година гимназиско
образование

ISBN 978-608-4777-55-7

COBISS.MK-ID 69471493

PËRMBAJTJA

TEMA 1.....7

TEKNOLOGJITË BASHKËKOHORE NË INFORMATIKË DHE TRANSFORMIMI DIGJITAL	8
SIGURIA KIBERNETIKE DHE MBROJTJA E TË DHËNAVE PERSONALE	12
KRIPTOGRAFIA	19
TEKNOLOGJIA BLOCKCHAIN – REGJISTRIME DIGJITALE	
TË BESUESHME NË BOTËN BASHKËKOHORE.....	26
ROBOTIKA DHE AUTOMATIZIMI.....	32
INTELIGJENCA ARTIFICIALE DHE MËSIMI MAKINERIK.....	38
MODELE TË MËDHA GJUHËSORE DHE INTELIGJENCA ARTIFICIALE GJENERATIVE	46
PROGRAMIMI NË C++	52

TEMA 2.....53

PARAQITJA DIGJITALE E TË DHËNAVE NË KOMPJUTER.....	54
SISTEMET E NUMËRIMIT DHE KONVERTIMET	58
PARAQITJA E NUMRAVE TË PLOTË DHE GAMA E TË DHËNAVE	64
OPTIMIZIMI DHE EFIKASITETI I ALGORITMEVE	68
KOMBINIMI I KUSHTEVE DHE CIKLEVE	73
PROGRAMIMI MODULAR DHE KONCEPTI „PËRÇAJ DHE SUNDË“	83
DEFINIMI I FUNKSIONEVE NË C++	88
PARAMETRAT DHE KTHIMI I VLERËS	94
VARIABLAT LOKALE DHE GLOBALE NË FUNKSIONE	100
FUNKSIONET THEMELORE MATEMATIKORE NGA BIBLIOTEKA CMATH (PJESA 1).....	106
ZGJIDHJA E DETYRAVE TEKSTUALE ME ZBATIMIN E FUNKSIONEVE.....	112
DETYRA PËR USHTRIMIN E FUNKSIONEVE	118
VARGU NJËDIMENSIONAL SI STRUKTURË E TË DHËNAVE	121
DEKLARIMI DHE INICIALIZIMI I VARGJEVE NJËDIMENSIONALE	127
INDEKSI I ELEMENTIT DHE QASJA NË ELEMENTET E VARGUT.....	132
FUTJA DHE SHËTYPJA E ELEMENTEVE TË VARGUT	137
OPERACIONET THEMELORE ME ELEMENTET E VARGUT	142
FUTJA E ELEMENTEVE NË NJË VARG DHE LLOGARITJET THEMELORE.....	148

PËRCAKTIMI I ELEMENTIT MË TË MADH DHE MË TË VOGËL NË NJË VARG	153
NUMËRIMI I ELEMENTEVE NË NJË VARG	158
VARGJE ME KARAKTERE DHE KËRKIMI I SHENJAVE	163
ZBATIMI I FUNKSIONEVE MATEMATIKORE MBI ELEMENTET E VARGUT	168
ZBATIMI I KOMBINUAR I FUNKSIONEVE DHE VARGJEVE NJËDIMENSIONALE.....	173
DETYRA PËR USHTRIME ME VARGJE	179

TEMA 3.....183

KONCEPTET THEMELORE TË BAZAVE TË TË DHËNAVE	184
KRIJIMI I TABELAVE NË BAZËN E TË DHËNAVE.....	189
LLOJET E TË DHËNAVE.....	196
ÇELËSI PRIMAR DHE LIDHJA E TABELAVE	204
FORMA PËR FUTJEN DHE SHQYRTIMIN E TË DHËNAVE	210
PYETËSORËT (QUERIES) – KËRKIMI DHE PËRZGJEDHJA E TË DHËNAVE.....	217
RAPORTET (REPORTS) – SHFAQJA DHE PRINTIMI I TË DHËNAVE	226
KRIJIMI I LETRAVE QARKORE ME BAZË TË TË DHËNAVE.....	232
SQL – GJUHË THEMELORE PËR PUNË ME BAZA TË TË DHËNAVE	239

TEMA 4.....245

MULTIMEDIA	247
GRAFIKA VEKTORIALE DHE RASTER	250
BAZAT E PËRPUNIMIT TË IMAZHEVE.....	253
SHTRESAT (LAYERS) NË GRAFIKË	267
INTELIGJENCA ARTIFICIALE GJENERATIVE PËR GRAFIKËN.....	271

TEMA 5.....279

KONCEPTI I FAQES WEB.....	281
BAZAT E HTML-SË	285



Jetojmë në një kohë në të cilën teknologjia është pjesë e pandashme e jetës së përditshme. Kompjuterët, interneti, pajisjet mobile dhe shërbimet digjitale na mundësojnë të komunikojmë, të mësojmë, të punojmë dhe të krijojmë më shpejt. Pas të gjitha këtyre teknologjive qëndrojnë njohuritë dhe aftësitë nga fusha e informatikës. Informatika sot nuk nënkupton vetëm përdorimin e kompjuterit. Ajo përfshin të kuptuarit e mënyrës se si funksionojnë teknologjitë bashkëkohore, përpunimin dhe organizimin e të dhënave, zhvillimin e programeve, krijimin e përmbajtjeve digjitale dhe zgjidhjen e problemeve me ndihmën e sistemeve kompjuterike.

Ky tekst shkollor është hartuar në përputhje me programin mësimor të lëndës së Informatikës për vitin e dytë të arsimit gjimnazial dhe synon t'u ndihmojë nxënësve të fitojnë njohuri bashkëkohore dhe aftësi praktike të nevojshme për pjesëmarrje të suksesshme në shoqërinë digjitale. Përmbajtjet e tekstit shkollor përfshijnë teknologjitë bashkëkohore në informatikë, programimin, bazat e të dhënave, multimedien, grafikën kompjuterike dhe zhvillimin e uebit. Vëmendje e veçantë i kushtohet zbatimit praktik të njohurive përmes shembujve, detyrave dhe aktiviteteve që i nxisin nxënësit të mendojnë, të hulumtojnë dhe të krijojnë zgjidhje të tyre. Qëllimi i tekstit shkollor nuk është vetëm të ofrojë informacione të reja, por edhe të zhvillojë të menduarit logjik, kreativitetin, shkrim-leximin digjital dhe gatishmërinë për të nxënë gjatë gjithë jetës. Në një kohë kur teknologjia ndryshon vazhdimisht, aftësia për të mësuar, për t'u përshtatur dhe për t'i zbatuar njohuritë në situata të reja bëhet gjithnjë e më e rëndësishme.

Shpresojmë që ky tekst shkollor të jetë një udhërrëfyes i dobishëm në zbulimin e botës së informatikës dhe t'i motivojë nxënësit të vazhdojnë të hulumtojnë, të krijojnë dhe t'i zhvillojnë idetë e tyre.

Informatika nuk është vetëm një lëndë që mësohet në klasë. Ajo është gjuha e botës bashkëkohore dhe një mjet për krijimin e së ardhmes.



TEMA 1





1.1

TEKNOLOGJITË BASHKËKOHORE NË INFORMATIKË DHE TRANSFORMIMI DIGJITAL

Jetojmë në një epokë në të cilën ndryshimet ndodhin me një shpejtësi të jashtëzakonshme dhe një nga nxitësit kryesorë të këtyre ndryshimeve është informatika. Sot, termi „teknologji“ nuk i referohet më vetëm kompjuterëve të zakonshëm dhe programeve bazë, por një sistemi kompleks inovacionesh që ndryshojnë në mënyrë thelbësore mënyrën se si punojmë, komunikojmë dhe jetojmë.

Me termin teknologji bashkëkohore në informatikë nënkuptohen mjetet më të reja dhe të avancuara, pajisjet harduerike, zgjidhjet softuerike dhe sistemet e rrjeteve që mundësojnë përpunimin, ruajtjen dhe analizimin e shpejtë të sasive të mëdha të të dhënave. Këto teknologji përfaqësojnë kalimin nga menaxhimi i dikurshëm i thjeshtë dhe statik i bazave të të dhënave drejt një ekosistemi aktiv dhe inteligjent. Ato mundësojnë analizimin e sasive të mëdha të të dhënave dhe marrjen e vendimeve të mençura në kohë reale, duke lidhur botën digjitale me atë fizike.

Teknologjitë bashkëkohore evoluojnë dhe integrohen vazhdimisht në të gjitha fushat e jetës njerëzore, duke krijuar fenomenin e quajtur transformim digjital – një proces i integritit të teknologjisë digjitale në të gjitha fushat e një shoqërie, në arsim, ekonomi, prodhimtari apo në biznes, i cili ndryshon në mënyrë thelbësore mënyrën se si funksionojnë ato.

Bota e informatikës sot udhëhiqet nga disa shtylla kryesore:

- Inteligjenca artificiale (AI) dhe modelet e mëdha gjuhësore (LLM), të cilat përfaqësojnë sisteme të avancuara që mund të përpunojnë dhe të gjenerojnë tekst, kod dhe ide në një nivel të ngjashëm me atë njerëzor (p.sh. modelet GPT, Gemini, Claude), duke analizuar miliarda të dhëna brenda pak sekondash.
- Robotika e avancuar, e cila ndërthur informatikën, inteligjencën artificiale dhe inxhinierinë mekanike. Bëhet fjalë për makina autonome ose gjysmëautonome, të afta të kryejnë detyra fizike komplekse, të mësojnë nga mjedisi dhe të reagojnë ndaj ndryshimeve në kohë reale.
- Interneti i Gjërave (IoT) përfaqëson një rrjet pajisjesh fizike (telefona inteligjentë, pajisje shtëpiake, sensorë industrialë) që janë të lidhura me internetin dhe komunikojnë me njëra-tjetrën.
- Teknologjitë e kompjuterizimit në re (Cloud Computing), të cilat mundësojnë ruajtjen dhe përpunimin e të dhënave në serverë të largët, në vend të kompjuterit lokal (p.sh. Google Drive, Netflix).

Zbatimi i këtyre teknologjive në jetën tonë të përditshme është shumë i përhapur. Mundësitë për zbatimin e tyre janë praktikisht të pakufizuara. Zbatimi praktik i këtyre teknologjive është veçanërisht i pranishëm në:

- **Puna krijuese dhe asistentët personalë.** Me ndihmën e modeleve të mëdha gjuhësore, sot çdo person ka qasje në një „tru digjital“ të disponueshëm menjëherë. Këto modele ndihmojnë në shkrimin e eseve, përkthimin e gjuhëve në kohë reale, mësimin e aftësive të reja dhe gjenerimin e kodit kompjuterik.

- **Automatizimi inteligjent** në shtëpi dhe në industri. Përmes robotikës dhe IoT-së, në jetën e përditshme përdorim fshesa robotike që hartojnë vetë planimetrinë e shtëpisë. Në industri, robotët kryejnë operacione të sakta, paketimin e produkteve dhe detyra fizike të rrezikshme, duke eliminuar rrezikun për njerëzit.
- **Arsimi dhe mjekësia.** Modelet e mëdha gjuhësore shërbejnë si tutorë personale, të cilët e përshtatin shpjegimin sipas nivelit të njohurive të nxënësit. Në mjekësi, robotët kirurgjikë (si sistemi Da Vinci) u mundësojnë mjekëve të kryejnë operacione me saktësi jashtëzakonisht të lartë, edhe në distancë.
- **Shërbimet elektronike (E-shërbimet).** Pagesa e faturave me një klikim, bankingu online dhe mbështetja e përdoruesve përmes „chatbot“-ëve inteligjentë, të cilët funksionojnë 24/7 falë inteligjencës artificiale.
- **Transporti,** në të cilin përdoren sisteme navigimi që ndihmojnë në zgjedhjen e itinerarit më të përshtatshëm.
- **Bujqësia,** përmes zbatimit të sistemeve inteligjente për monitorimin e kushteve atmosferike dhe menaxhimin e ujitjes.

1.1.1 PËRPARËSITË, SFIDAT DHE NDIKIMI NË SHOQËRI

Ashtu si çdo revolucion i madh, edhe transformimi digjital sjell me vete përfitime të rëndësishme, por edhe sfida që shoqëria duhet t'i kapërcejë.

Aspekte pozitive	Aspekte negative
Efikasiteti dhe produktiviteti: Modelet e mëdha gjuhësore dhe robotët marrin përsipër detyrat manuale dhe të përsëritura, duke i liruar njerëzit për të menduar në mënyrë krijuese.	Humbja e vendeve të punës: Automatizimi, robotika dhe inteligjenca artificiale mund të zëvendësojnë profesione të caktuara (nga punëtorët e fabrikave deri te administratorët), duke e bërë të domosdoshëm rikualifikimin e shpejtë profesional.
Siguria në vendin e punës: Robotët mund të punojnë në mjedise të rrezikshme (miniera, thellësi detare, centrale bërthamore), ku jeta e njeriut është e rrezikuar.	Varësia dhe dembelizmi kognitiv: Mbështetja e tepruar të modelet gjuhësore për shkrim dhe të menduar mund të zvogëlojë aftësitë për të menduar në mënyrë kritike dhe krijuese, veçanërisht te të rinjtë.
Qasja në njohuri dhe gjithëpërfshirja: Barrierat gjuhësore po zhduken falë përkthyesve të avancuar të bazuar në modelet e mëdha gjuhësore (LLM), duke mundësuar që informacioni të jetë i qasshëm për të gjithë.	Përhapja e dezinformatave: Modelet e mëdha gjuhësore mund të gjenerojnë pa dashje informacione të rreme që duken bindëse (të ashtuquajturat „halucinacione“) ose të keqpërdoren për qëllime propagandistike.
Diagnostikim dhe kujdes më i mirë shëndetësor: Ndërthurja e inteligjencës artificiale me robotikën mjekësore mundëson jetëgjatësi më të madhe dhe një cilësi më të lartë të jetës për njerëzit.	Siguria kibernetike dhe çështjet etike: Kush mban përgjegjësi nëse roboti bën një gabim? Si mund të mbrohen të dhënat personale me të cilat trajnohen modelet gjuhësore?

Teknologjitë bashkëkohore duhet të përdoren në mënyrë të mençur në jetën e përditshme. Shembujt në vijim përshkruajnë mënyra se si teknologjitë mund të përdoren në mënyrë të dobishme.

- Të nxënit adaptiv për arsimim gjatë gjithë jetës. Në vend të kurseve klasike, është më praktike të përdoren platforma të hapura arsimore të bazuara në teknologjinë cloud, për rikualifikim ose për përvetësimin e aftësive të reja (gjuhë, programim), të cilat përshtaten me ritmin individual të të nxënit.

- Menaxhimi inteligjent i shëndetit (IoT dhe telemjekësia). Përdorimi i orëve inteligjente dhe aplikacioneve për monitorimin e gjumit, pulsit dhe aktivitetit fizik, me qëllim që këto të dhëna të ndahen me mjekun amë përmes platformave digjitale (në vend të vizitave fizike për kontrole rutinë).
- Automatizimi i detyrave rutinë me bashkëpunëtorë të bazuar në inteligjencën artificiale. Përdorimi i mjeteve të inteligjencës artificiale si „asistentë kognitivë“ për hartimin e shpejtë të e-maileve të biznesit, strukturimin e shënimeve nga takimet ose kërkimin në arkiva të mëdha tekstuale, me qëllim lirim të kohës për zgjidhje krijuese të problemeve.
- Ndërgjegjësimi për kufizimin e përdorimit të pajisjeve digjitale. Si përgjigje ndaj uljes së përqendrimit, është më mirë të përdoren aplikacione që kufizojnë qasjen në rrjetet sociale gjatë orarit të punës, me qëllim ruajtjen e aftësisë për punë të përqendruar.



PËRFUNDIM:

Teknologjitë bashkëkohore në informatikë, nga robotika që transformon botën fizike deri te modelet e mëdha gjuhësore që transformojnë punën intelektuale, janë mjete të fuqishme. Ndikimi i tyre në shoqëri varet tërësisht nga mënyra se si i përdorim. Potenciali i tyre i jashtëzakonshëm duhet të shfrytëzohet për lehtësimin e jetës, duke zhvilluar njëkohësisht dhe të menduarit kritik, si dhe ndërgjegjësimin për normat etike, për të mbrojtur shoqërinë nga pasojat negative.



A E DI?

A e di se sot një telefon inteligjent ka fuqi më të madhe përpunuese se kompjuterët që u përdorën gjatë misioneve të para hapësinore drejt Hënës?



PYETJE PËR VERIFIKIMIN E NJOHURIVE

1. Si përkufizohet termi „teknologjitë bashkëkohore“ në informatikë?
2. Rendit tri shembuj të teknologjive bashkëkohore që sot janë më të përhapura në botë.
3. Shpjego, me fjalët e tua, lidhjen ndërmjet teknologjive bashkëkohore dhe procesit të transformimit digjital.
4. Si funksionon Interneti i Gjërave (IoT) dhe si e lidh ai jetën tonë të përditshme me rrjetin?
5. Pse modelet e mëdha gjuhësore (LLM) ndonjëherë gjenerojnë „halucinacione“ (informacione të pasakta) dhe çfarë nënkupton kjo?
6. Përkruaj një situatë në të cilën ti personalisht ose shtëpia jote inteligjente përdor teknologjinë IoT gjatë ditës.
7. Krahaso anët pozitive dhe negative të përdorimit masiv të robotëve në fabrika. Kush përfiton dhe kush humbet?
8. Analizo ndikimin e teknologjive cloud në mënyrën se si sot ruhen dhe ndahen të dhënat, në krahasim me të kaluarën (hard disqet, memoriet USB).
9. Kush duhet të mbajë përgjegjësi etike dhe ligjore nëse një robot mjekësor bën një gabim gjatë një operacioni – programuesi, inxhinieri apo mjeku? Arsyejo qëndrimin tënd.
10. Vlerëso nëse zhvillimi i shpejtë i inteligjencës artificiale çon në izolim shoqëror dhe në uljen

e kreativitetit të njerëzit apo, përkundrazi, e rrit efikasitetin e tyre.

11. Propozo një ide për një „pajisje të mençur“ të re (bazuar në IoT ose inteligjencë artificiale) që do të mund të zgjidhte një problem real në shkollën ose qytetin tënd.

12. Shkruaj një plan të shkurtër veprimi (3–4 hapa) se si një biznes që tradicionalisht punon me dokumente në letër mund të realizojë në mënyrë të sigurt transformimin digjital.



DETYRA HULUMTUESE DHE PRAKTIKE

1. Përgatit një listë me dhjetë teknologji bashkëkohore që përdor gjatë një dite. Për secilën teknologji, shkruaj në cilën fushë përdoret dhe si ta lehtëson ajo jetën e përditshme.
2. Zgjidh një institucion publik (shkollë, bibliotekë, bankë, spital ose komunë) dhe hulumto se cilat shërbime digjitale u ofron qytetarëve. Përgatit një raport të shkurtër.
3. Mendo se si dukej një shkollë para njëzet vjetësh dhe krahasojë atë me shkollën e sotme. Rendit të paktën pesë ndryshime që janë rezultat i teknologjive bashkëkohore.
4. Në grup, së bashku me shokët dhe shoqet e klasës, diskutoni për përparësitë dhe mangësitë e përdorimit të telefonave inteligjentë në jetën e përditshme. Shënoni përfundimet.
5. Hulumto një teknologji bashkëkohore për të cilën mendon se do të ketë ndikim të madh në të ardhmen. Përgatit një prezantim të shkurtër, në të cilin do të shpjegosh se si funksionon dhe në cilat fusha mund të zbatohet.



PËRFUNDIM:

Teknologjitë bashkëkohore përfaqësojnë zgjidhje teknike dhe informatike që e lehtësojnë jetën e përditshme.

Transformimi digjital përfaqëson procesin e përfshirjes së teknologjive digjitale në fusha të ndryshme të shoqërisë.

Teknologjitë bashkëkohore gjejnë zbatim në arsim, shëndetësi, industri, transport, tregti dhe komunikim.

Përdorimi i teknologjive bashkëkohore sjell përfitime të shumta, por edhe sfida të caktuara.

Përdorimi i përgjegjshëm dhe i sigurt i teknologjive bashkëkohore kontribuon në mbrojtjen e të dhënave personale dhe të privatësisë digjitale.



PËR ATA QË DËSHIROJNË TË DINË MË SHUMË

Hulumto për të gjetur përgjigjen e pyetjes: „Si e di një aplikacion se çfarë dua të shoh në vazhdim?“ Gjetja e përgjigjes do të të hapë dyert drejt së ardhmes në botën e informatikës së avancuar.





Siguria kibernetike përbën një segment kyç të botës së sotme digjitale, ku mbrojtja e informacionit bëhet gjithnjë e më e rëndësishme. Me rritjen e vazhdueshme të përdorimit të internetit dhe shërbimeve digjitale, të dhënat personale ekspozohen ndaj rreziqeve dhe keqpërdorimeve të ndryshme. Prandaj, ruajtja e privatësisë dhe siguri i mbrojtjes së përshtatshme të të dhënave janë thelbësore për çdo individ dhe organizatë. Kërcënimet digjitale, si sulmet e hakerëve, viruset dhe phishing-u, paraqesin rrezik serioz për sigurinë e informacionit. Kuptimi i këtyre sfidave dhe zbatimi i parimeve themelore të mbrojtjes mundësojnë përdorim më të sigurt të teknologji digjitale. Kështu, siguria kibernetike bëhet një aftësi e domosdoshme për funksionimin në shoqërinë bashkëkohore.

- të përdorësh internetin për të kërkuar informacione;
- të komunikosh përmes platformave digjitale;
- të përdorësh llogari të ndryshme përdoruesi;
- të shpërndash përmbajtje digjitale;
- të dallos se cilat të dhëna personale duhet të mbrohen.

1. Sa llogari përdoruesi përdor gjatë një jave?
2. A ke menduar ndonjëherë se çfarë të dhënash personale lë pas kur përdor internetin?
3. Si do të veproje nëse një person i panjohur do të të kërkonte fjalëkalimin ose të dhëna të tjera personale?
4. A duhet të ndahen me njerëz të tjerë të gjitha informacionet që publikohen në internet?
5. Pse është e rëndësishme të jesh i kujdesshëm gjatë përdorimit të shërbimeve digjitale?

- të shpjegosh pse është e rëndësishme mbrojtja e të dhënave personale dhe digjitale.
- të përshkruash masat themelore të mbrojtjes (fjalëkalime të forta, programe antivirus, përdorim i kujdesshëm i internetit).
- të zbatosh rregullat për përdorim të sigurt të internetit në situata konkrete, për të mbrojtur të dhënat e tua gjatë përdorimit të rrjeteve sociale ose postës elektronike.
- të dallosh sjelljet e sigurta nga ato të pasigurta në internet dhe të analizosh pasojat e mundshme të mbrojtjes së pamjaftueshme të të dhënave.
- të propozosh mënyra për përmirësimin e sigurisë tënde digjitale.

12

postës elektronike, fjalëkalimet dhe informacione të tjera. Për këtë arsye, përdorimi i sigurt i teknologjive digjitale po bëhet gjithnjë e më i rëndësishëm për çdo përdorues.

Me zhvillimin e teknologjive bashkëkohore shfaqen edhe rrezeqe të ndryshme që mund të ndikojnë në sigurinë e të dhënave dhe në privatësinë e përdoruesve. Ekzistojnë përpjekje të vazhdueshme për vjedhjen e të dhënave personale, qasjen e paautorizuar në llogaritë e përdoruesve dhe përhapjen e programeve keqdashëse. Për të zvogëluar këto rrezeqe, është e nevojshme të njihen rregullat themelore të sigurisë kibernetike dhe të zbatohen masat për mbrojtjen e të dhënave personale gjatë përdorimit të përditshëm të teknologjive digjitale.

1.2.1 ÇFARË ËSHTË SIGURIA KIBERNETIKE DHE PSE ËSHTË E RËNDËSISHME?

Sa herë që lidheni me internetin, pavarësisht nëse shfletoni rrjetet sociale, dërgoni postë elektronike apo luani lojëra në internet, në të vërtetë po hapni derën drejt botës digjitale. Por, ashtu siç e mbyllni me çelës shtëpinë për t'u mbrojtur nga hajdutët, po ashtu duhet të kujdeseni edhe për sigurinë e hapësirës suaj digjitale.

Siguria kibernetike përfaqëson një tërësi masash, rregullash dhe procedurash që mundësojnë mbrojtjen e sistemeve kompjuterike, rrjeteve dhe të dhënave digjitale nga kërcënimet dhe keqpërdorimet e ndryshme. Në shoqërinë bashkëkohore, një numër i madh aktiviteteesh varen nga teknologjitë e informacionit dhe interneti. Përdorimi i postës elektronike, rrjeteve sociale, platformave digjitale për mësim, bankimit elektronik dhe aplikacioneve të ndryshme celulare nënkupton shkëmbim të vazhdueshëm të informacionit përmes rrjeteve digjitale. Për këtë arsye, siguria e këtyre sistemeve luan një rol të rëndësishëm në mbrojtjen e përdoruesve dhe të të dhënave të tyre.

Siguria kibernetike nuk lidhet vetëm me mbrojtjen e kompanive dhe institucioneve të mëdha, por edhe me çdo përdorues individual. Përdorimi i përditshëm i telefonave inteligjentë, tabletëve dhe kompjuterëve nënkupton një nivel të caktuar përgjegjësie për mbrojtjen e të dhënave personale. Me zbatimin e masave të përshtatshme mund të zvogëlohen rreziqet nga qasja e paautorizuar në të dhëna, vjedhja e identitetit ose keqpërdorimi i llogarive të përdoruesve. Njohja e parimeve themelore të sigurisë kibernetike kontribuon në përdorimin më të sigurt të teknologjive bashkëkohore.

1.2.2 TË DHËNAT PERSONALE DHE PRIVATËSIA DIGJITALE

Të dhënat personale përfaqësojnë informacione përmes të cilave mund të identifikohet një person. Në këtë grup bëjnë pjesë emri dhe mbiemri, adresa e banimit, numri i telefonit, adresa e postës elektronike, numri personal i identifikimit, fotografitë dhe shumë informacione të tjera që mund të lidhen me një përdorues të caktuar. Në shoqërinë digjitale, çdo ditë lihen gjurmë të ndryshme të të dhënave personale gjatë përdorimit të shërbimeve të internetit, aplikacioneve celulare dhe platformave elektronike. Për këtë arsye, mbrojtja e tyre përbën një pjesë të rëndësishme të sigurisë digjitale.

Privatësia digjitale përfaqëson të drejtën e çdo përdoruesi për të vendosur në mënyrë të pavarur se cilat informacione do të ndajë dhe me kë do t'i ndajë ato. Gjatë përdorimit të internetit, është e nevojshme të përzgjidhen me kujdes të dhënat që publikohen në rrjetet sociale ose në platforma të tjera digjitale. Ndarja e tepërt e informacioneve personale mund të çojë në keqpërdorime të ndryshme dhe të rrezikojë sigurinë e përdoruesit. Përdorimi i përgjegjshëm i shërbimeve digjitale kontribuon në mbrojtjen më të mirë të privatësisë dhe në zvogëlimin e mundësive për keqpërdorimin e të dhënave personale.

1.2.3 RREGULLAT PËR PËRDORIM TË SIGURT TË INTERNETIT

Për të qenë i sigurt në internet, nuk nevojiten njohuri të avancuara informatike. Mjafton të zbatosh tri rregullat themelore të higjienës digjitale:

- **Krijto fjalëkalime të forta:** Fjalëkalimi është linja e parë e mbrojtjes. Një fjalëkalim i dobët (si 123456 ose emri123) mund të zbulohet nga hakerët brenda pak sekondash. Një fjalëkalim i fortë duhet të ketë të paktën 12 karaktere dhe të përmbajë një kombinim të shkronjave të mëdha e të vogla, numrave dhe simboleve të veçanta (p.sh. @, #, \$, %). Gjithashtu, nuk duhet të përdoret i njëjti fjalëkalim për disa llogari të ndryshme.
- **Përdor dhe përditëso programin antivirus:** Programet antivirus veprojnë si mbrojtës digjitalë që skanojnë pajisjen, zbulojnë programe të dyshimta (viruse, trojanë, programe spiunazhi) dhe i bllokojnë para se të shkaktojnë dëm. Është e rëndësishme që ky program të përditësohet rregullisht, në mënyrë që të jetë në gjendje të identifikojë kërcënimet e reja.
- **Përdore internetin me kujdes:** Kjo nënkupton të jesh i vetëdijshëm se ku klikon. Shmang shkarkimin e skedarëve nga faqe interneti të panjohura, mos kliko reklama të dyshimta (që premtojnë fitime të shpejta ose çmime falas) dhe mos u lidh me rrjete publike e të pasigurta Wi-Fi (si në kafene ose parqe) kur vendos të dhëna të ndryshme.

1.2.4 KËRCËNIMET DHE RREZIQET MË TË SHPESHTA DIGJITALE

Me zhvillimin e internetit dhe të teknologjive bashkëkohore shfaqen kërcënime të ndryshme digjitale që mund të ndikojnë në sigurinë e përdoruesve. Ndër kërcënimet më të shpeshta janë mesazhet e rreme dhe mashtrimet në internet, programet keqdashëse, qasja e paautorizuar në llogaritë e përdoruesve dhe vjedhja e të dhënave personale. Në situata të caktuara, sulmuesit përpiqen të fitojnë besimin e përdoruesve dhe t'i bindin ata që t'i zbulojnë vullnetarisht fjalëkalimet ose informacione të tjera të ndryshme. Këto përpjekje zakonisht realizohen përmes postës elektronike, mesazheve ose faqeve të rreme të internetit.

Përveç sulmeve të drejtpërdrejta, ekzistojnë edhe rreziqe që vijnë si pasojë e përdorimit të pakujdesshëm të teknologjive digjitale. Përdorimi i fjalëkalimeve të dobëta, shkarkimi i skedarëve nga burime të paverifikuara ose hapja e mesazheve të dyshimta mund të çojë në rrezikimin e sigurisë së sistemeve kompjuterike dhe të të dhënave personale. Njohja e kërcënimeve të mundshme dhe përdorimi i kujdesshëm i shërbimeve të internetit përbëjnë një hap të rëndësishëm në mbrojtjen e identitetit digjital dhe në zvogëlimin e rreziqeve të mundshme.

1.2.5 PËRDORIMI I SIGURT DHE I PËRGJEGJSHËM I INTERNETIT

Përdorimi i sigurt i internetit nënkupton zbatimin e rregullave dhe krijimin e shprehive të ndryshme që kontribuojnë në mbrojtjen e përdoruesve dhe të të dhënave të tyre. Një nga masat themelore është përdorimi i fjalëkalimeve të sigurta, të cilat përmbajnë një kombinim shkronjash, numrash dhe simbolesh të veçanta. Gjithashtu, është e dobishme që i njëjti fjalëkalim të mos përdoret për disa shërbime të ndryshme dhe që fjalëkalimet të ndryshohen rregullisht. Mbrojtje shtesë ofron edhe përdorimi i autentifikimit me shumë faktorë, ku për qasjen në llogarinë e përdoruesit kërkohet një konfirmim shtesë.

Përdorimi i përgjegjshëm i internetit nënkupton edhe sjellje të kujdesshme gjatë shpërndarjes së informacioneve dhe përdorimit të shërbimeve digjitale. Është e nevojshme të kontrollohet besueshmëria e faqeve të internetit, të hapen me kujdes mesazhet elektronike dhe të mos ndahen të dhënat personale me persona të panjohur. Përditësimi i rregullt i sistemeve operative dhe programeve kontribuon në mbrojtjen më të mirë nga kërcënimet e reja të sigurisë. Duke zhvilluar shprehje të mira digjitale dhe duke menduar në mënyrë kritike, përdoruesit mund ta përmirësojnë ndryshëm sigurinë e tyre gjatë përdorimit të teknologjive bashkëkohore të informacionit.

Mënyra më e mirë për ta kuptuar sigurinë kibernetike është përmes zbatimit të saj në situatat e përditshme me të cilat përballemi.

Shembulli 1: Përdorimi i rrjeteve sociale (TikTok, Instagram, Snapchat...)

Rrjetet sociale janë platforma ku mund të shpërndahen lehtësisht shumë informacione. Për mbrojtjen e të dhënave në këtë situatë, duhet të zbatohen rregullat e mëposhtme:

- **Profili privat (Private Account):** Ndrysho cilësimet e privatësisë në mënyrë që vetëm miqtë e pranuar të mund t'i shohin publikimet, fotografitë dhe vendndodhjen tënde.
- **Kujdes me publikimin e vendndodhjes („check-in“):** Mos e shpërnda vendndodhjen tënde të saktë në kohë reale. Kjo mund t'u tregojë personave keqdashës se ku ndodhesh saktësisht ose se nuk je në shtëpi.
- **Mendo para se të publikosh:** Çdo fotografi ose informacion, për shembull fotografia e dokumentit të identifikimit, adresa e shtëpisë ose emri i shkollës, mund të mbetet në internet edhe pasi ta fshish.

Shembulli 2: Përdorimi i postës elektronike (e-mail)

Posta elektronike është një nga mjetet më të shpeshta që përdoren për sulme të ashtuquajtura **phishing** (mashtime përmes të cilave sulmuesit përpiqen të vjedhin fjalëkalimet ose të dhënat e kartelave bankare, duke u paraqitur rrejtshëm si kompani ose institucione të ligjshme, si banka, rrjete sociale apo shërbime postare). Për t'u mbrojtur në këtë situatë, zbatohen rregullat e mëposhtme:

- **Mos iu përgjigj dërguesve të panjohur:** Nëse merr një e-mail nga një adresë e panjohur që kërkon përgjigje urgjente, për shembull: „Profili yt do të fshihet, kliko këtu për të konfirmuar fjalëkalimin“, fshije menjëherë.
- **Kontrollo lidhjet dhe skedarët bashkëngjitur:** Mos hap kurrë skedarë bashkëngjitur me prapashtesa si .exe ose .zip nga persona të panjohur, sepse mund të përmbajnë programe keqdashëse që mund të dëmtojnë kompjuterin ose të komprometojnë të dhënat e tua.

Shembulli 3: Aktivizimi i autentifikimit me dy faktorë (2FA): Kjo paraqet një shtresë shtesë mbrojtjeje. Gjatë hyrjes në postën elektronike, përveç fjalëkalimit, sistemi kërkon edhe një kod që mund të dërgohet, për shembull, me SMS në telefonin tënd. Në këtë mënyrë, edhe nëse dikush e zbulon fjalëkalimin, nuk do të mund të hyjë në llogarinë tënde pa pasur qasje edhe në telefonin tënd.

1.2.6 NXËNËSI DIGJITAL DHE SIGURIA KIBERNETIKE

Gjatë një dite të zakonshme shkollore, nxënësi përdor një numër të madh shërbimesh digjitale. Ai hyn në platforma elektronike për mësim, kërkon informacione në internet, komunikon përmes aplikacioneve të ndryshme dhe përdor postën elektronike. Gjatë të gjitha këtyre aktiviteteve shkëmbehen të dhëna të ndryshme, ndërsa për qasje në këto shërbime zakonisht përdoren llogari përdoruesi dhe fjalëkalime. Përdorimi i sigurt i këtyre shërbimeve kontribuon në mbrojtjen e të dhënave personale dhe në zvogëlimin e mundësive për keqpërdorim.

Mendo për një situatë në të cilën një nxënës merr një mesazh se ka fituar një çmim dhe se duhet të vendosë emrin e përdoruesit, fjalëkalimin dhe të dhënat e llogarisë bankare në një faqe interneti të panjohur. Nëse nuk kontrollohen dërguesi dhe adresa e faqes së internetit, mund të ndodhë keqpërdorimi i llogarisë së përdoruesit. Sjellja e kujdesshme, kontrollimi i burimit të informacionit dhe shmangia e shpërndarjes së të dhënave personale me persona të panjohur përbëjnë masa të thjeshta, por të rëndësishme për përmirësimin e sigurisë digjitale.

Sjellja në internet ndikon në masë të madhe në pasojat me të cilat mund të përballesh në hapësirën e pasigurt të internetit. Çdo veprim që kryen në botën digjitale lë gjurmë. Shpesh, për shkak të ritmit të shpejtë të jetës së përditshme, harrojmë se interneti nuk është vetëm një hapësirë për argëtim, por edhe një mjedis i mbushur me rreziqe të fshehura. Dallimi midis shfletimit të sigurt dhe një problemi serioz digjital shpesh varet nga vendimet dhe shprehitë tona.

1.2.7 SJELLJE TË SIGURTA DHE TË PASIGURTA NË INTERNET

Për t'u mbrojtur, së pari duhet të dish t'i dallosh situatat e rrezikshme. Sjellja në internet mund të ndahet në përgjithësi në dy kategori:

SJELLJE TË PASIGURTA (RREZIK)	SJELLJE TË SIGURTA (MBROJTJE)
Përdorimi i rrjeteve publike dhe të pasigurta Wi-Fi (pa fjalëkalim) për t'u identifikuar në rrjetet sociale ose për të përdorur shërbimet e bankimit elektronik.	Përdorimi i internetit celular (4G/5G) ose i një rrjeti shtëpiak të mbrojtur gjatë qasjes në llogari që përmbajnë të dhëna të ndjeshme.
Pranimi i kërkesave për miqësi nga persona të panjohur dhe ndarja e të dhënave private me ta.	Komunikimi vetëm me persona që i njeh në jetën reale dhe ruajtja e privatësisë.
Klikimi i lidhjeve joshëse që premtojnë shpërblime falas, telefona ose pjesëmarrje në „lojëra shpërblyese“ (phishing).	Injorimi i mesazheve të dyshimta dhe kontrollimi i faqeve zyrtare të internetit për të verifikuar nëse janë të vërteta ofertat ose shpërblimet.
Shkarkimi dhe instalimi i lojërave, filmave ose programeve pirate nga burime të pasigurta dhe faqe torrent.	Shkarkimi i programeve vetëm nga faqet zyrtare të internetit dhe dyqanet zyrtare të aplikacioneve (Google Play, App Store ose faqet zyrtare të prodhuesve).

Pasojat e sjelljes së pakujdesshme dhe të mbrojtjes së pamjaftueshme mund të jenë serioze dhe afatgjata. Nëse të dhënat e tua personale (fjalëkalimet, fotografitë, numrat personalë të identifikimit) bien në duart e personave të tjerë, mund të ndodhin skenarët e mëposhtëm:

- **Vjedhja e identitetit:** Hakerët mund t'i keqpërdorin të dhënat dhe fotografitë e tua për të krijuar profile të rreme, për të kryer mashtrime në emrin tënd ose për të shantazhuar miqtë dhe familjarët e tu duke u kërkuar para.
- **Dëmi financiar:** Përmes mesazheve phishing ose pagesave të pasigurta, sulmuesit mund të fitojnë qasje në llogaritë bankare ose në kartelat e kreditit të prindërve të tu dhe t'i tërheqin ose transferojnë paratë nga llogaria brenda pak minutash.
- **Infektimi me softuer shantazhues (ransomware):** Në vend të një virusi të zakonshëm, në pajisjen tënde mund të instalohet një softuer keqdashës që i bllokoi të gjitha dokumentet, fotografitë dhe projektet e tua, ndërsa hackerët mund të kërkojnë pagesën e një shpërblimi për t'i zhbllokuar dhe rikthyer ato.
- **Dëmtimi i mirëqenies emocionale dhe i reputacionit:** Shkelja e privatësisë dhe publikimi i paautorizuar i bisedave ose fotografive private (bullizmi kibernetik) mund të shkaktojnë stres të madh, ndjenjë turpi dhe pasoja afatgjata në jetën shoqërore të personit.

Përmirësimi i sigurisë nuk kërkon aftësi të ndërlikuara teknike, por zbatimin e shprehive të mira digjitale. Më poshtë janë disa këshilla konkrete që mund t'i zbatosh menjëherë:

1. **Aktivizo autentifikimin me dy faktorë (2FA):** Ky është një nga hapat më të rëndësishëm për mbrojtjen e llogarisë. Përveç fjalëkalimit, gjatë hyrjes në profilin tënd (Instagram, TikTok, Google etj.) kërkohet edhe një kod shtesë verifikimi, i cili mund të dërgohet në telefonin tënd. Në këtë mënyrë, edhe nëse dikush e di fjalëkalimin tënd, nuk do të mund të hyjë në llogarinë tënde pa kryer edhe verifikimin shtesë.
2. **Përdor një menaxher fjalëkalimesh (password manager):** Në vend që të përdorësh të njëjtin fjalëkalim të thjeshtë për të gjitha llogaritë, gjë që përbën një rrezik të madh sigurie, përdor një aplikacion që gjeneron dhe ruan në mënyrë të sigurt fjalëkalime të ndryshme dhe të forta për secilën nga llogaritë e tua.
3. **Bëj rregullisht kopje rezervë (Backup):** Një herë në javë ose një herë në muaj, transfero dokumentet dhe fotografitë e tua të rëndësishme në një disk të jashtëm, në një USB ose në një cloud të sigurt. Në këtë mënyrë, edhe nëse pajisja vidhet ose dëmtohet, të dhënat do të mbeten të sigurta dhe do të mund të përdoren përsëri.

4. **Përditësimi i rregullt i sistemit:** Mos i shty njoftimet për përditësimin e sistemit operativ të telefonit ose kompjuterit. Këto përditësime shpesh përmbajnë „arnime“ sigurie për dobësitë e zbuluara së fundmi, të cilat hakerët mund t'i shfrytëzojnë për sulme.

A E DI?



A e di se një fjalëkalim i fortë duhet të jetë mjaftueshëm i gjatë dhe të përmbajë një kombinim të shkronjave të mëdha dhe të vogla, numrave dhe simboleve speciale? Përdorimi i fjalëkalimeve të ndryshme për shërbime të ndryshme digjitale rrit ndjeshëm sigurinë e llogarive të përdoruesve.



HULUMTO

Hulumto për të zbuluar se cilat rregulla për përdorim të sigurt të internetit zbatohen në shkollën tënde, nga shokët e tu ose në familjen tënde. Bazuar në rezultatet e hulumtimit, përpilo një listë me të paktën dhjetë rregulla që kontribuojnë në mbrojtjen më të mirë të të dhënave personale dhe të privatësisë digjitale.



PËRFUNDIM:

Siguria kibernetike nuk varet nga kompleksiteti i kompjuterëve, por nga mënyra se si sillemi dhe veprimet në botën digjitale. Me përdorimin e fjalëkalimeve të forta, programeve antivirus dhe me pak kujdes në çdo klikim, bëhemi përdorues më të mbrojtur dhe më të përgjegjshëm në botën digjitale. Hapësira e internetit është po aq e sigurt sa jemi të kujdesshëm ne. Duke zhvilluar ndërgjegjësimin për rreziqet dhe duke zbatuar masa të thjeshta mbrojtëse, bëhemi krijues të sigurisë sonë dhe marrim kontrollin mbi jetën tonë digjitale.



PYETJE PËR PËRSËRITJEN E NJOHURIVE

1. Pse siguria kibernetike ka një rol të rëndësishëm në shoqërinë bashkëkohore?
2. Cilat informacione konsiderohen të dhëna personale?
3. Si mund të rrezikohet privatësia digjitale e përdoruesit?
4. Cilat janë rreziqet më të zakonshme gjatë përdorimit të internetit dhe shërbimeve digjitale?
5. Në ç'mënyrë shprehinë e mira digjitale kontribuojnë në mbrojtjen më të mirë të të dhënave personale?



DETYRA HULUMTUESE DHE PRAKTIKE

1. Përpilo një listë të të dhënave personale që përdoren më shpesh gjatë krijimit të një llogarie në internet. Shpjego pse është e rëndësishme që këto të dhëna të mbrohen.
2. Hulumto se cilat rregulla zbatohen platformat më të njohura të internetit për mbrojtjen e llogarive të përdoruesve dhe cilat të dhëna personale mbledhin dhe përpunojnë. Përgatit një raport të shkurtër.
3. Mendo dhe përshkruaj tri situata në të cilat përdorimi i pakujdesshëm i internetit mund të çojë në keqpërdorimin e të dhënave personale.

4. Në grup, së bashku me shokët e klasës, diskutoni për përparësitë dhe rreziqet e shpërndarjes së fotografive dhe të informacioneve personale në rrjetet sociale. Shënoni përfundimet.
5. Përgatit një udhëzues të shkurtër me dhjetë këshilla për përdorim të sigurt dhe të përgjegjshëm të internetit, i cili do të ishte i dobishëm për nxënësit e shkollës tënde.



PËR ATA QË DËSHIROJNË TË DINË MË SHUMË

Siguria kibernetike është një nga fushat me zhvillimin më të shpejtë në informatikë. Një numër i madh institucioneve shtetërore, kompanish, bankash, spitalesh dhe institucioneve arsimore përdorin çdo ditë sisteme moderne të sigurisë për mbrojtjen e të dhënave dhe shërbimeve të tyre. Ekzistojnë profesione të ndryshme që lidhen me këtë fushë, si: analist i sigurisë kibernetike, ekspert për mbrojtjen e rrjeteve, studiues i kërcënimeve digjitale dhe specialist i forenzikës digjitale. Me zhvillimin e teknologjive moderne, nevoja për profesionistë që do të kontribuojnë në mbrojtjen e hapësirës digjitale vazhdimisht rritet.



PYETJE PËR VERIFIKIMIN E NJOHURIVE

Rretho përgjigjen e saktë.

1. Siguria kibernetike ka të bëjë me:

- a) prodhimin e kompjuterëve;
- b) mbrojtjen e sistemeve dhe të dhënave digjitale;
- c) riparimin e telefonave celularë;
- ç) prodhimin e lojërave kompjuterike.

2. Shpjego termat.



Term	Shpjegim
Të dhëna personale	_____
Privatësia digjitale	_____
Siguria kibernetike	_____

3. Plotëso.

Fjalëkalimi i sigurt duhet të jetë _____.

4. Cilat janë tri karakteristikat kryesore që duhet të përmbajë një fjalëkalim i fortë dhe i sigurt?

5. Shpjego me fjalët e tua se si funksionon një sulm „phishing“ dhe cili është qëllimi i tij kryesor?

6. Nëse do të merrje një e-mail nga një dërgues i panjohur me një skedar bashkëngjitur me prapashtesën .exe, ku thuhet se është një video lojë falas, cilët hapa konkretë do të ndërmerre për t'u mbrojtur?

7. Sipas teje, cili është më i sigurt: përdorimi i një rrjeti publik Wi-Fi në një kafene apo përdorimi i internetit tënd celular (4G/5G)? Si ndikon zgjedhja e rrjetit në sigurinë e të dhënave të tua personale?

8. Vlerëso pasojat e mundshme të vjedhjes së identitetit në rrjetet sociale. Sipas mendimit tënd, a janë pasojat më të rënda në aspektin financiar apo në aspektin emocional/social për një adoleshent? Arsyejo qëndrimin tënd.

9. Propozo tri rregulla themelore që duhet të përmbajë një „Kod për sjellje të sigurt në internet“ i shkollës, me qëllim që të mbrohen të dhënat e nxënësve në shkollën tënde.

1.3



KRIPTOGRAFIA

Çdo ditë dërgojmë mesazhe të shumta, blejmë online dhe identifikohemi në platforma të ndryshme. Por, a ke pyetur ndonjëherë veten se si «udhëtojnë» këto të dhëna nëpër botë pa i vjedhur ose lexuar dikush? Përgjigja qëndron në një shkencë të lashtë, por sot ultramoderne – kriptografia.

Kriptografia është shkenca e komunikimit të sigurt, e cila mundëson mbrojtjen e informacionit nga qasja e paautorizuar në botën digjitale. Në thelb të saj qëndron procesi i enkriptimit, përmes të cilit teksti i zakonshëm dhe i lexueshëm shndërrohet në një kod të pakuptueshëm duke përdorur algoritme matematikore. Që mesazhi të bëhet përsëri i kuptueshëm për marrësin, përdoret procesi i kundërt, i quajtur dekriptim. Në varësi të mënyrës së përdorimit të çelësave, kriptografia ndahet në kriptografi simetrike dhe kriptografi asimetrike. Në kriptografinë simetrike përdoret i njëjti çelës sekret si për enkriptimin, ashtu edhe për dekriptimin e të dhënave. Në të kundërt, kriptografia asimetrike përdor një çift prej dy çelësash të ndryshëm: çelësin publik, i cili është i disponueshëm për të gjithë dhe përdoret për enkriptim, dhe çelësin privat, të cilin e posedon vetëm marrësi dhe që përdoret për dekriptim.

TASHMË DI...

- të shpjegosh pse duhet të mbrohen të dhënat personale;
- të dallos përdorimin e sigurt dhe të pasigurt të internetit;
- të shpjegosh se çfarë është privatësia digjitale;
- të japësh shembuj të përdorimit të sigurt të teknologjive digjitale;
- të analizosh rreziqet e mundshme gjatë shkëmbimit të informacionit.

PYETJE PËR REFLEKTIM

1. Si do t'ia dërgoje një mesazh sekret një shoku/shoqeje, pa mundur ta lexojë dikush tjetër?
2. A kishin nevojë njerëzit në të kaluarën të mbronin informacionet e tyre?
3. Cilat informacione sot do të doje të mbeteshin konfidenciale?
4. Si mendon, si i mbrojnë bankat dhe dyqanet online të dhënat e klientëve të tyre?
5. A është e mundur që vetëm personi i autorizuar ta lexojë mesazhin e dërguar?

NË PËRMBAJTJEN QË VIJON DO TË MËSOSH TË:

- të përkufizosh konceptin e kriptografisë me nocionet themelore: enkriptimi, çelësi, mesazhi sekret;
- të dallos përdorimin e kriptografisë në një situatë konkrete të përditshme;
- të shpjegosh se si përdoret kriptografia për mbrojtjen e të dhënave;
- të dallos informacionet e mbrojtura nga ato të pambrojtura;
- të shpjegosh rëndësinë e kriptografisë për privatësinë dhe sigurinë.

Komunikimi dhe shkëmbimi i informacionit kanë qenë gjithmonë një pjesë e rëndësishme e jetës së përditshme të njerëzve. Në periudha të ndryshme historike, ka ekzistuar nevoja për dërgimin e mesazheve që duhej të mbeteshin sekrete dhe të kuptoheshin vetëm nga personat për të cilët ishin të destinuara. Planet ushtarake, marrëveshjet diplomatike dhe informacionet e rëndësishme shtetërore shpeshherë mbroheshin me metoda të ndryshme për fshehjen e përmbajtjes së tyre. Nevoja për mbrojtjen e informacionit çoi në zhvillimin e teknikave të ndryshme për enkriptimin e mesazheve.

Me zhvillimin e teknologjive moderne, kriptografia është bërë një pjesë e rëndësishme e jetës së përditshme. Sot ajo nuk përdoret vetëm në ushtri dhe në sistemet shtetërore, por edhe gjatë përdorimit të shërbimeve bankare, postës elektronike, dyqaneve online dhe aplikacioneve mobile. Falë metodave kriptografike, informacionet e ndjeshme mund të shkëmbehen në mënyrë të sigurt dhe të mbrohen nga qasja e paautorizuar.

1.3.1 ÇFARË PARAQET KRIPTOGRAFIA?

Kriptografia është një fushë shkencore që merret me mbrojtjen e informacionit nga qasja e paautorizuar. Qëllimi themelor i kriptografisë është të mundësojë që një mesazh ose e dhënë e caktuar të lexohet vetëm nga personat për të cilët është destinuar. Për këtë qëllim përdoren procedura të ndryshme, përmes të cilave mesazhi i kuptueshëm shndërrohet në një formë që nuk mund të kuptohet lehtësisht nga persona të tjerë. Personi që e dërgon mesazhin dhe personi që e pranon atë duhet të kenë informacionet e nevojshme për ta lexuar saktë përmbajtjen. Në këtë mënyrë zvogëlohet mundësia që informacionet konfidenciale të keqpërdoren.

Kriptografia sot përdoret në një numër të madh fushash të jetës së përditshme. Gjatë përdorimit të shërbimeve bankare elektronike, dyqaneve online, aplikacioneve mobile dhe shërbimeve të ndryshme digjitale, të dhënat personale dhe informacionet financiare mbrohen përmes metodave kriptografike. Gjithashtu, posta elektronike, certifikatat digjitale dhe shumë shërbime të internetit përdorin teknika të ndryshme për mbrojtjen e informacionit. Falë kriptografisë, përdoruesit mund të shkëmbejnë të dhëna në mënyrë më të sigurt dhe të përdorin shërbime moderne digjitale. Për ta kuptuar atë, duhet të njohim katër koncepte kryesore: enkriptimin, dekriptimin, çelësin kriptografik dhe mesazhin sekret.

1.3.2 ENKRIPTIMI, DEKRIPTIMI DHE ÇELËSI KRIPTOGRAFIK

Enkriptimi paraqet një proces gjatë të cilit një mesazh i kuptueshëm shndërrohet në një formë të pakuptueshme, me qëllim që të mbrohet nga qasja e paautorizuar. Mesazhi që mund të lexohet quhet mesazh burimor, ndërsa pas procesit të enkriptimit fitohet mesazhi i enkriptuar. Ky mesazh duket si një varg shkronjash dhe simbolesh të rastësishme. Edhe nëse dikush e intercepton gjatë transmetimit në internet, për të ai është i pakuptimtë, sepse nuk mund ta lexojë. Për shembull, fjala „Përsëritje!“ mund të shndërrohet në një kod të pakuptueshëm, si: „Xy9#z!“. Kjo realizohet me ndihmën e rregullave të ndërlikuara matematikore, të quajtura algoritme.

Personi që e pranon mesazhin duhet të disponojë informacionet përkatëse në mënyrë që ta kthejë atë përsëri në formën e tij fillestare. Procesi i kthimit të mesazhit në formën e tij të kuptueshme quhet dekriptim.

Për enkriptimin dhe dekriptimin me sukses përdoret çelësi kriptografik. Çelësi përfaqëson një grup rregullash ose të dhënash që mundësojnë që mesazhi të mbrohet siç duhet dhe më pas të lexohet përsëri. Në sistemet e thjeshta, i njëjti çelës mund të përdoret si për enkriptim, ashtu edhe për dekriptim, ndërsa në sistemet më moderne përdoren çelësa të ndryshëm për këto dy procese. Pa pasur çelësin përkatës, mesazhi i enkriptuar është shumë më i vështirë për t'u kuptuar nga personat e paautorizuar.

1.3.3 TEKNOLOGJIA BASHKËKOHORE NË PRAKTIKË

Shifra e Cezarit

Një nga shembujt më të njohur të enkriptimit është shifra e Cezarit. Në këtë mënyrë të mbrojtjes, çdo shkronjë e mesazhit zëvendësohet me një shkronjë tjetër që ndodhet disa vende më tej në alfabet. Nëse zhvendosja është për tri vende, shkronja A zëvendësohet me Ç, B me D, C me Dh dhe kështu me radhë. Personi që e di rregullin e zhvendosjes mund ta lexojë lehtësisht mesazhin, ndërsa për personat e painformuar ai duket si një varg i rastësishëm shkronjash.

Shifra e Cezarit sot nuk ka përdorim praktik për mbrojtjen e informacioneve të rëndësishme, sepse rregulli i saj është i thjeshtë dhe mund të zbulohet lehtësisht. Megjithatë, ajo ka rëndësi të madhe arsimore, sepse në mënyrë të thjeshtë shpjegon idetë themelore të kriptografisë. Me ndihmën e saj mund të kuptohet lehtësisht se si funksionojnë enkriptimi, dekriptimi dhe përdorimi i çelësit kriptografik.

Shembull

Le të supozojmë se zhvendosja është për tri shkronja.

Mesazhi burimor	Mesazhi i enkriptuar
A	Ç
B	D
C	Dh
Ç	E

Nëse mesazhi është „INTERNET“, pas enkriptimit do të fitohet një varg i ri shkronjash, sipas rregullit të zgjedhur për zhvendosjen.

Procesi i enkriptimit (shkronjë për shkronjë):

I (+3 vende më tej) = **L**

N (+3 vende më tej) = **P**

T (+3 vende më tej) = **H**

E (+3 vende më tej) = **Z**

R (+3 vende më tej) = **U**

N (+3 vende më tej) = **P**

E (+3 vende më tej) = **Z**

T (+3 vende më tej) = **H**

Mesazhi sekret (çelësi 3) do të jetë vargu i mëposhtëm i shkronjave: LPHZUPZH



Kriptografia moderne

Me zhvillimin e sistemeve kompjuterike dhe të internetit, kriptografia gjen zbatim shumë më të gjerë sesa në të kaluarën. Sot, një numër i madh shërbimesh digjitale përdorin metoda bashkëkohore kriptografike për mbrojtjen e informacioneve që shkëmbehen ndërmjet përdoruesve dhe sistemeve informatike. Gjatë hyrjes në një llogari përdoruesi, dërgimit të postës elektronike, përdorimit të dyqaneve online ose bankingut elektronik, informacionet e ndjeshme mbrohen me procedura të ndryshme kriptografike. Në këtë mënyrë zvogëlohet mundësia që të dhënat të lexohen ose të ndryshohen nga persona të paautorizuar.

Në kriptografinë moderne përdoren metoda të ndryshme për mbrojtjen e informacionit. Në disa sisteme, i njëjti çelës kriptografik përdoret për enkriptimin dhe dekriptimin e mesazheve, ndërsa në sisteme të tjera përdoren dy çelësa të ndryshëm. Zgjedhja e metodës varet nga niveli i kërkuar i sigurisë dhe nga fusha në të cilën zbatohet. Algoritmet moderne kriptografike janë krijuar për të ofruar një nivel të lartë mbrojtjeje dhe për të mundësuar komunikim të sigurt ndërmjet përdoruesve dhe sistemeve informatike.

Zbatimi i kriptografisë në jetën e përditshme

Kriptografia është pjesë e shumë shërbimeve digjitale që njerëzit i përdorin çdo ditë. Gjatë përdorimit të shërbimeve bankare elektronike, kryerjes së pagesave online ose qasjes në faqe të ndryshme interneti, të dhënat personale dhe financiare mbrohen përmes mekanizmave kriptografikë. Aplikacionet mobile të komunikimit përdorin kriptografinë për të mundësuar që mesazhet të lexohen vetëm nga personat që marrin pjesë në bisedë. Aplikacionet WhatsApp, Viber dhe Signal përdorin të ashtuquajturën enkriptim nga skaji në skaj (end-to-end). Kjo do të thotë se mesazhi enkriptohet në telefonin tënd dhe dekriptohet vetëm kur arrin te personi të cilit i është dërguar. Madje, as kompanitë, si Meta, nuk mund t'i lexojnë bisedat e tua.

Gjithashtu, shumë shërbime elektronike përdorin certifikata digjitale, të cilat mundësojnë verifikimin e identitetit të përdoruesve dhe të faqeve të internetit. Kriptografia nuk është diçka që përdoret vetëm nga shërbimet sekrete. Ajo është e pranishme pothuajse kudo në botën digjitale.

Zbatimi i kriptografisë kontribuon në përmirësimin e sigurisë dhe të besueshmërisë së sistemeve moderne informatike. Falë metodave kriptografike, përdoruesit mund të përdorin shërbime të ndryshme digjitale me një nivel më të lartë sigurie. Në të njëjtën kohë, është e rëndësishme të kuptohet se kriptografia përbën vetëm një pjesë të sigurisë së përgjithshme digjitale. Për të arritur një nivel më të lartë mbrojtjeje, është e nevojshme të zbatohen edhe masa të tjera, si përdorimi i fjalëkalimeve të sigurt, ndarja e kujdesshme e informacionit dhe përditësimi i rregullt i sistemeve informatike.

Gjatë një dite, çdo nxënës mund të përdorë kriptografinë disa herë, pa qenë i vetëdijshëm për këtë. Gjatë hyrjes në ditarin elektronik, dërgimit të një mesazhi përmes një aplikacioni mobile, blerjes së një produkti online ose përdorimit të shërbimeve bankare elektronike, të dhënat mbrohen me metoda moderne kriptografike. Në këtë mënyrë zvogëlohet mundësia që informacionet personale të keqpërdoren.

Kur një përdorues hyn në një dyqan online dhe vendos të dhënat e pagesës, sistemet kriptografike mundësojnë që këto informacione të transmetohen në mënyrë të sigurt ndërmjet përdoruesit dhe serverit. Falë kësaj, të dhënat e ndjeshme nuk dërgohen si tekst i zakonshëm, por në një formë të mbrojtur, e cila është shumë më e vështirë për t'u lexuar nga persona të paautorizuar.

Si të dallojmë informacionet e mbrojtura nga informacionet e pambrojtura?

Si nxënës, është shumë e rëndësishme të dallosh menjëherë nëse informacionet me të cilat merresh janë të mbrojtura dhe të sigurt:

- **Informacione të mbrojtura.** Këto janë të dhëna të enkriptuara, në të cilat kanë qasje vetëm personat e autorizuar. Më së lehti mund t'i dallosh përmes ikonës së drynit në shfletues dhe protokollit HTTPS (shkronja S do të thotë Secure / i sigurt). Futja e fjalëkalimeve ose e të dhënave personale në faqe të tilla është e mbrojtur.
- **Informacione të pambrojtura.** Këto janë informacione publike që transmetohen si tekst i zakonshëm përmes protokollit HTTP (pa ikonën e drynit). Gjithashtu, të pambrojtura konsiderohen të gjitha informacionet që i shpërndan në profile publike në rrjetet sociale, në forume të hapura ose kur përdor një rrjet publik Wi-Fi në një kafene pa fjalëkalim. Një person me njohuri bazë për sigurinë kibernetike mund të përpiket t'i përgjojë ato.

Pa kriptografi, bota moderne digjitale do ta kishte shumë të vështirë të funksiononte. Rëndësia e saj është jashtëzakonisht e madhe për dy arsye:

- **Mbrojtja e privatësisë:** Ajo na mundëson të kemi një hapësirë personale në botën digjitale. Na jep lirinë të komunikojmë me miqtë, familjen ose mjekët, duke ditur se mendimet, fotografitë dhe informacionet tona private mbeten të mbrojtura.
- **Sigurimi i sigurisë digjitale:** Kriptografia mbron institucionet shtetërore, spitalet, shkollat dhe bankat nga sulmet kibernetike. Ajo ndihmon në parandalimin e ndryshimit të paautorizuar të të dhënave, vjedhjes së identitetit dhe ndërprerjes së sistemeve të rëndësishme.



PËRFUNDIM:

Kriptografia është mbrojtësi i padukshëm i jetës sonë digjitale. Kuptimi i parimeve të saj themelore na ndihmon të jemi përdorues më të informuar dhe më të kujdesshëm, të cilët dinë se si ta mbrojnë gjurmën e tyre digjitale.



A E DI?

A e di se, sa herë që në një faqe interneti vëren simbolin e drynit pranë adresës, komunikimi ndërmjet pajisjes tënde dhe faqes është i mbrojtur me teknologji moderne kriptografike?



HULUMTO

Hulumto se në cilat aktivitete të përditshme përdor kriptografi pa qenë i vetëdijshëm për këtë. Përgatit një listë me të paktën pesë shembuj dhe shpjego se çfarë roli ka kriptografia në secilin prej tyre.



PYETJE PËR PËRSËRITJEN E NJOHURIVE

1. Çfarë paraqet kriptografia?
2. Cili është ndryshimi midis enkriptimit dhe dekriptimit?
3. Çfarë roli ka çelësi kriptografik?
4. Pse shifra e Cezarit ka rëndësi edukative edhe sot?
5. Në cilat shërbime digjitale të përditshme zbatohet kriptografia?
6. Pse njerëzit kanë nevojë t'i mbrojnë informacionet e tyre?
7. Si dallohen enkriptimi dhe dekriptimi?
8. Çfarë roli ka çelësi kriptografik në mbrojtjen e informacionit?
9. Pse shifra e Cezarit sot ka kryesisht rëndësi edukative?
10. Në cilat shërbime moderne digjitale zbatohet kriptografia në jetën e përditshme?



DETYRA HULUMTUESE DHE PRAKTIKE

1. Duke përdorur rregullin e shifrës së Cezarit, krijo një mesazh sekret për shokun ose shoqen tënde të klasës. Më pas shkëmbeni mesazhet dhe përpikuni t'i dekriptoni.
2. Hulumto se në cilat aktivitete të përditshme përdor kriptografi pa qenë i/e vetëdijshëm/e për këtë. Shëno të paktën pesë shembuj.
3. Krahaso shifrën e Cezarit me metodat moderne të mbrojtjes së informacionit. Shëno të paktën tri dallime.
4. Mendo pse bankat, dyqanet online dhe aplikacionet mobile përdorin kriptografi. Përgatit një tekst të shkurtër me përfundimet dhe vëzhgimet e tua.
5. Hulumto se çfarë është një certifikatë digjitale dhe shpjego pse është e rëndësishme për komunikimin e sigurt përmes internetit.



MBAJ MEND ÇFARË MËSOVE

Kriptografia është shkenca e mbrojtjes së informacionit.

Enkriptimi e shndërron mesazhin e kuptueshëm në një formë të mbrojtur.

Dekriptimi mundëson që mesazhi të lexohet përsëri.

Çelësi kriptografik ka një rol të rëndësishëm në mbrojtjen e informacionit.

Kriptografia ka zbatim të gjerë në sistemet moderne informatike dhe në shërbimet digjitale që përdorim çdo ditë.



PËR ATA QË DËSHIROJNË TË DINË MË SHUMË

Kriptografia moderne është një nga fushat më të rëndësishme të informatikës dhe luan një rol të rëndësishëm në funksionimin e internetit. Përveç mbrojtjes së mesazheve dhe të dhënave financiare, metodat kriptografike përdoren edhe për nënshkrimin digjital të dokumenteve, krijimin e certifikatave digjitale, mbrojtjen e rrjeteve kompjuterike dhe funksionimin e teknologjisë blockchain. Me zhvillimin e sistemeve moderne informatike, kriptografia vazhdimisht përmirësohet për të siguruar një nivel më të lartë të sigurisë dhe besueshmërisë së informacionit.





KONTROLLO NJOHURITË E TUA

1. Rretho përgjigjen e saktë.

Kriptografia përdoret për:

- a) Përshejtimin e kompjuterëve;
- b) Mbrojtjen e informacionit;
- c) Riparimin e rrjeteve kompjuterike;
- d) Krijimin e aplikacioneve celulare.

2. Përkufizo termin kriptografi, duke përdorur detyrimisht në përkufizim tre konceptet kyçe: enkriptimi, çelësi dhe mesazhi sekret.

3. E saktë apo e pasaktë.

Enkriptimi dhe deenkriptimi përfaqësojnë të njëjtën procedurë.

4. Shpjego konceptet.

Term	Shpjegim
Shifrimi	_____
Deshifrimi	_____
Çelës kriptografik	_____

5. Plotëso.

Një nga mënyrat më të njohura dhe më të thjeshta të shifrimit është _____.

6. Si ndihmon kriptografia në praktikë në mbrojtjen e të dhënave gjatë transmetimit të tyre në internet? Shpjego procesin e shndërrimit të tekstit të thjeshtë në tekst të shifruar dhe anasjelltas.

7. Përmend një situatë konkrete nga jeta tënde e përditshme (për shembull, gjatë përdorimit të aplikacioneve për komunikim ose gjatë blerjeve online), në të cilën, pa e kuptuar, përdorn kriptografi, dhe përshkruaj se çfarë ndodh me të dhënat e tua në atë moment.

8. Bëj një krahasim midis informacionit të mbrojtur dhe atij të pambrojtur në internet. Në bazë të cilave shenja digjitale (simboleve ose protokolleve në shfletuesin e internetit) mund të dallojnë menjëherë nxënësit nëse një faqe interneti i mbron apo nuk i mbron të dhënat e tyre?

9. Vlerëso rëndësinë e kriptografisë për shoqërinë: Çfarë do të ndodhte me privatësinë e qytetarëve dhe sigurinë e institucioneve (si bankat dhe spitalet) nëse kriptografia do të zhdukej papritur nga bota digjitale? Arsyeto pasojat.

10. Imagjino se duhet t'ua shpjegosh nxënësve të moshës më të vogël pse është e rëndësishme të mbrojnë të dhënat e tyre. Krijë tri „rregulla të arta“ për shfletim të sigurt, të bazuara në dallimin midis informacionit të mbrojtur dhe atij të pambrojtur në internet.



DETYRË PROBLEMORE

Imagjino se duhet t'i dërgosh një mesazh konfidencial një shoku/shoqeje përmes internetit. Shpjego pse është e rëndësishme që mesazhi të jetë i shifruar dhe cilat përfitime ofron kriptografia.



1.4

TEKNOLOGJIA BLOCKCHAIN – REGJISTRIME DIGJITALE TË BESUESHME NË BOTËN BASHKËKOHORE

Teknologjia blockchain përfaqëson një zgjidhje moderne informatike që mundëson ruajtjen dhe gjurmimin e sigurt të regjistrimeve digjitale. Falë metodave kriptografike dhe mënyrës së organizimit të të dhënave, çdo ndryshim mbetet i regjistruar dhe mund të verifikohet lehtësisht. Për shkak të këtyre veçorive, teknologjia blockchain gjen zbatim në fusha të ndryshme të shoqërisë bashkëkohore, ndërsa përdorimi i saj vazhdon të zgjerohet.

TASHMË DI...

- të shpjegosh se çfarë përfaqëson kriptografia;
- të dallosh shifrimin nga deshifrimi;
- të shpjegosh rolin e çelësit kriptografik;
- të japësh shembuj të mbrojtjes së informacionit;
- të identifikosh zbatimin e kriptografisë në sistemet moderne informatike.

PYETJE PËR REFLEKTIM

1. Si mund të ruhen informacione të rëndësishme në mënyrë të tillë që çdo ndryshim të mbetet i regjistruar?
2. A është e mundur që një bazë të dhënash të përdoret nga një numër i madh përdoruesish pa një menaxhues qendror?
3. Pse është e rëndësishme të dihet se kush dhe kur ka bërë një ndryshim të caktuar në një regjistrim?
4. A mund të përdoret teknologjia që përdoret për kriptovalutat edhe në fusha të tjera?
5. Si mund të kontribuojnë teknologjitë moderne informatike në rritjen e besueshmërisë së shërbimeve digjitale?

NË PËRMBAJTJET QË VIJOJNË DO TË MËSOSH:

- të shpjegosh se çfarë përfaqëson teknologjia blockchain;
- të shpjegosh strukturën e një blloku dhe lidhjen ndërmjet blloqeve;
- të shpjegosh parimin e decentralizimit;
- të japësh shembuj të zbatimit të teknologjisë blockchain.

Në shoqërinë bashkëkohore krijohen çdo ditë një numër i madh regjistrimesh digjitale që duhet të ruhen në mënyrë të sigurt dhe të mbrohen nga ndryshimet e paautorizuara. Transaksionet bankare, të dhënat për pronësinë, kartelat shëndetësore, kontratat dhe dokumentet e ndryshme elektronike përfaqësojnë

informacione që duhet të jenë të sakta dhe të besueshme. Nëse këto regjistrime mund të ndryshohen ose të fshihen lehtësisht, mund të shfaqen probleme dhe keqkuptime serioze.

1.4.1 ÇFARË PËRFAQËSON TEKNOLOGJIA BLOCKCHAIN?

Në shoqërinë bashkëkohore ekziston nevoja që informacionet e rëndësishme të ruhen në mënyrë të sigurt dhe të jenë të disponueshme për verifikim. Gjatë blerjes së një pasurie të paluajtshme, transferimit të mjeteve financiare, lëshimit të diplomave ose lidhjes së kontratave, është e nevojshme të ekzistojë një evidencë e saktë e aktiviteteve të kryera. Nëse regjistrimet mund të ndryshohen ose të fshihen lehtësisht, mund të shfaqen probleme të mëdha dhe keqkuptime. Për këtë arsye, teknologjitë moderne informatike ofrojnë zgjidhje që mundësojnë ruajtjen dhe gjurmimin e sigurt të regjistrimeve digjitale.

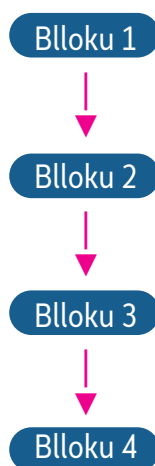
Teknologjia blockchain përfaqëson një mënyrë të veçantë të organizimit dhe ruajtjes së të dhënave. Në vend që informacionet të ruhen në një regjistër të vetëm qendror, ato organizohen në disa blloqe të ndërlidhura. Çdo bllok i ri përmban informacione të reja dhe lidhet me bllokun paraardhës, duke formuar një zinxhir. Falë kësaj mënyre organizimi dhe përdorimit të metodave kriptografike, të dhënat mbrohen, ndërsa çdo ndryshim mbetet i regjistruar. Për këtë arsye, teknologjia blockchain përfaqëson një mënyrë të sigurt për ruajtjen e llojeve të ndryshme të regjistrimeve digjitale.

Bllloqet, transaksionet dhe decentralizimi

Elementi themelor i teknologjisë blockchain është blloku. Blloku përfaqëson një tërësi digjitale në të cilën ruhen informacione të caktuara. Në varësi të zbatimit të sistemit, blloku mund të përmbajë të dhëna për transaksione financiare, ndryshime të pronësisë, kontrata të lidhura ose regjistrime të tjera të rëndësishme. Pasi krijohet një regjistrim i ri, formohet një bllok i ri, i cili lidhet me bllokun paraardhës dhe bëhet pjesë e zinxhirit të blloqeve. Në këtë mënyrë krijohet një histori e njëpasnjëshme e të gjitha ngjarjeve që kanë ndodhur në sistem.

Teknologjia blockchain shpesh lidhet me konceptin e decentralizimit. Në sistemet tradicionale zakonisht ekziston një menaxhues qendror, i cili ruan dhe kontrollon të dhënat. Në sistemet blockchain, regjistrimet mund të jenë të disponueshme për një numër të madh pjesëmarrësish të ndërlidhur, ndërsa çdo ndryshim i ri regjistrohet dhe mund të verifikohet. Kjo mënyrë organizimi kontribuon në rritjen e transparencës dhe besueshmërisë së sistemit, sepse historia e regjistrimeve mbetet e ruajtur dhe mund të gjurmohet lehtësisht.

Paraqitje e lidhjes së blloqeve



Çdo bllok i ri lidhet me bllokun paraardhës dhe bëhet pjesë e zinxhirit të përbashkët të regjistrimeve.



Ku ruhen blloqet?

Një nga karakteristikat më të rëndësishme të teknologjisë blockchain është mënyra e ruajtjes së të dhënave. Në vend që të gjitha regjistrimet të ruhen në një kompjuter ose server qendror, kopje të sistemit blockchain mund të ruhen në disa kompjuterë të ndërlidhur që marrin pjesë në funksionimin e tij. Këta kompjuterë quhen nyje (ang. nodes) dhe secili prej tyre përmban një kopje të zinxhirit të blloqeve. Kur në sistem shtohet një bllok i ri, informacioni regjistrohet dhe përditësohet te të gjithë pjesëmarrësit e rrjetit. Falë kësaj mënyre organizimi, të dhënat nuk varen nga një vend i vetëm ruajtjeje, ndërsa siguria dhe disponueshmëria e tyre përmirësohen ndjeshëm.

Ruajtja e blloqeve në disa kompjuterë të ndërlidhur kontribuon në rritjen e besueshmërisë dhe qëndrueshmërisë së sistemit. Nëse një kompjuter ndalon së funksionuari ose shfaqet një problem teknik, kompjuterët e tjerë vazhdojnë t'i kenë të ruajtura të dhënat. Në të njëjtën kohë, çdo ndryshim i ri duhet të regjistrohet në sistem, gjë që mundëson verifikimin më të lehtë të saktësisë së regjistrimeve. Në këtë mënyrë, teknologjia blockchain siguron ruajtje të besueshme të informacionit dhe krijon besim ndërmjet pjesëmarrësve që përdorin rrjetin.

1.4.2 TEKNOLOGJIA BLOCKCHAIN NË BOTËN BASHKËKOHORE

Një nga zbatimet më të njohura të teknologjisë blockchain janë kriptovalutat. Kriptovalutat përfaqësojnë mjete digjitale të këmbimit që përdorin blockchain për regjistrimin e sigurt të transaksioneve. Çdo transaksion regjistrohet në një bllok dhe bëhet pjesë e zinxhirit të regjistrimeve. Falë kësaj, mund të gjurmohet historia e transaksioneve dhe të zvogëlohet mundësia e ndryshimeve të paautorizuara. Bitcoin përfaqëson një nga kriptovalutat më të njohura, por sot ekzistojnë edhe shumë sisteme të tjera që përdorin këtë teknologji. Teknologjia blockchain nuk përdoret vetëm për kriptovaluta. Ajo gjen zbatim në shëndetësi, logjistikë, arsim, administratë publike dhe në menaxhimin e regjistrimeve të ndryshme digjitale.

Përparësitë dhe mundësitë e teknologjisë Blockchain

Teknologjia blockchain ofron disa përparësi në ruajtjen dhe përpunimin e regjistrimeve digjitale. Falë ndërlidhjes së blloqeve, çdo ndryshim mbetet i regjistruar dhe mund të verifikohet. Kjo kontribuon në rritjen e transparencës dhe besueshmërisë në sistem. Gjithashtu, përdorimi i metodave kriptografike siguron mbrojtje më të mirë të informacionit dhe zvogëlon mundësinë e ndryshimeve të paautorizuara të të dhënave.

Përveç përparësive të shumta, përdorimi i teknologjisë blockchain sjell edhe sfida të caktuara. Për përdorimin e saj të suksesshëm nevojiten infrastrukturë e përshtatshme teknike dhe organizim i mirë i sistemeve që e zbatojnë atë. Gjithashtu, fushat e ndryshme të zbatimit kanë nevoja dhe kërkesa të ndryshme, prandaj zgjedhja e kësaj teknologjie varet nga qëllimi konkret i përdorimit. Me zhvillimin e informatikës dhe transformimit digjital, pritet që teknologjia blockchain të gjejë zbatim në një numër gjithnjë e më të madh të sistemeve dhe shërbimeve moderne informatike.



1.4.3 TEKNOLOGJIA BASHKËKOHORE NË PRAKTIKË

Teknologjia Blockchain dhe evidencat kadastrale

Në shumë shtete, shërbimet kadastrale mbajnë evidencë për pronësinë e tokës dhe pasurive të paluajtshme. Gjatë blerjes, shitjes, trashëgimit ose dhurimit të një pasurie, është e nevojshme që çdo ndryshim të regjistrohet me saktësi. Nëse regjistrimet nuk përditësohen ose mund të ndryshohen lehtësisht, mund të lindin mosmarrëveshje lidhur me pronësinë dhe të krijohen probleme të mëdha juridike. Për këtë arsye, shtete të ndryshme po shqyrtojnë zgjidhje moderne informatike që do të mundësojnë siguri dhe transparencë më të madhe të të dhënave kadastrale.

Në Afrikë ekzistojnë iniciativa për përdorimin e teknologjisë blockchain në menaxhimin e evidencave kadastrale dhe regjistrave të tokës. Një nga shtetet ku po zhvillohen projekte të tilla është Gana, ku zgjidhjet moderne informatike po shqyrtohen si një mundësi për përmirësimin e evidencës së pronësisë. Me ndihmën e teknologjisë blockchain mund të ruhet historia e të gjitha ndryshimeve të një pasurie, në mënyrë që çdo shitje, trashëgimi ose formë tjetër e transferimit të pronësisë të mbetet e regjistruar në mënyrë të përhershme. Kjo qasje kontribuon në rritjen e besueshmërisë së të dhënave dhe në zvogëlimin e mundësive për keqpërdorime.



A E DI?

A e di se teknologjia blockchain sot nuk përdoret vetëm për kriptovaluta? Përveç sektorit financiar, ajo përdoret ose po testohet edhe në shëndetësi, arsim, transport, logjistikë dhe në lëshimin e dokumenteve digjitale.



HULUMTO

Hulumto tri fusha në të cilat zbatohet teknologjia blockchain. Për secilën fushë, përmend:

- Çfarë evidentohet;
- Çfarë përfitimesh ofron blockchain;
- Në ç'mënyrë teknologjia kontribuon në sigurinë më të madhe të të dhënave.



PYETJE PËR PËRSËRITJE

1. Çfarë përfaqëson teknologjia blockchain?
2. Çfarë përfaqëson një bllok në sistemin blockchain?
3. Çfarë përfaqëson një transaksion?
4. Çfarë do të thotë decentralizimi?
5. Pse kriptovalutat përfaqësojnë një nga zbatimet më të njohura të teknologjisë blockchain?
6. Jep të paktën tri fusha në të cilat mund të zbatohet teknologjia blockchain.



DETYRA HULUMTUESE DHE PRAKTIKE

1. Imagjino se në shkollën tënde duhet të ruhen diploma elektronike. Shpjego se në ç'mënyrë teknologjia blockchain do të mund të ndihmonte në evidentimin dhe ruajtjen e tyre.
2. Hulumto një kriptovalutë sipas zgjedhjes sate dhe përgatit një raport të shkurtër për përdorimin dhe karakteristikat e saj themelore.
3. Përgatit një tabelë në të cilën do të krahasosh një bazë klasike të të dhënave me një sistem blockchain, sipas mënyrës së ruajtjes së informacionit.
4. Mendo se në cilat fusha në Republikën e Maqedonisë së Veriut do të mund të zbatohet teknologjia blockchain. Jep të paktën tre shembuj dhe arsyeto zgjedhjen tënde.
5. Hulumto se në ç'mënyrë teknologjia blockchain mund të kontribuojë në përmirësimin e evidencave kadastrale dhe në menaxhimin e pronësisë së pasurive të paluajtshme.



PËRFUNDIM:

Teknologjia blockchain përfaqëson një mënyrë të sigurt për organizimin dhe ruajtjen e regjistrimeve digjitale. Elementet themelore të një sistemi blockchain janë blloqet dhe lidhja e tyre në një zinxhir. Transaksioni përfaqëson një regjistrim që futet në sistem. Decentralizimi mundëson që regjistrimet të jenë të disponueshme dhe të verifikueshme nga më shumë pjesëmarrës. Kriptovalutat përfaqësojnë një nga zbatimet më të njohura të teknologjisë blockchain. Ajo gjen zbatim edhe në shumë fusha të tjera të shoqërisë bashkëkohore.



PËR ATA QË DËSHIROJNË TË DINË MË SHUMË

Teknologjia blockchain zhvillohet vazhdimisht dhe po gjen zbatim në fusha të reja. Përveç sektorit financiar, ajo përdoret për gjurmimin e produkteve në zinxhirët e furnizimit, për lëshimin e certifikatave digjitale, për menaxhimin e të dhënave mjekësore dhe për mbrojtjen e pronësisë intelektuale. Në shumë shtete dhe organizata po hulumtohen mënyra të reja të zbatimit të saj, me qëllim përmirësimin e sigurisë, transparencës dhe besueshmërisë së sistemeve të ndryshme informatike.



KONTROLLO NJOHURITË E TUA

1. Rretho përgjigjen e saktë.

Teknologjia blockchain më së shpeshti përdoret për:

- a) përpunimin e fotografive;
- b) ruajtjen dhe gjurmimin e sigurt të regjistrimeve digjitale;
- c) zhvillimin e sistemeve operative;
- d) riparimin e rrjeteve kompjuterike.

2. E saktë apo e pasaktë.

Kriptovalutat përfaqësojnë zbatimin e vetëm të teknologjisë blockchain.

3. Shpjego termat.

Termi	Shpjegimi
Blok	_____
Transaksioni	_____
Decentralizimi	_____

4. Plotëso.

Zbatimi më i njohur i teknologjisë blockchain janë _____.

5. Si zbatohet parimi i decentralizimit gjatë kryerjes së një transaksioni digjital në një rrjet blockchain, në krahasim me një pagesë tradicionale bankare?

6. Analizo rolin e decentralizimit në sigurinë e të dhënave. Pse është pothuajse e pamundur të fshihet ose të ndryshohet një transaksion i regjistruar më parë, pasi blloku të jetë lidhur në zinxhir?

7. Vlerëso përparësitë dhe mangësitë e kriptovalutave si zbatimi më i përhapur i teknologjisë blockchain. A mendon se ato mund t'i zëvendësojnë plotësisht paratë tradicionale në të ardhmen? Arsyeto qëndrimin tënd.

8. Përveç kriptovalutave, propozo një ide se si teknologjia blockchain, me karakteristikat e saj të pandryshueshmërisë dhe decentralizimit, mund të përdoret për zgjidhjen e një problemi real shoqëror, siç janë votimi në zgjedhje ose mbrojtja e të drejtave të autorit.

**DETYRË E BAZUAR NË SITUATË**

Imagjino se duhet të krijohet dhe të mbahet një evidencë për pronësinë e shtëpive në një vendbanim. Shpjego se në ç'mënyrë teknologjia blockchain do të mund të kontribuonte në sigurinë dhe besueshmërinë më të madhe të këtyre të dhënave.



1.5

ROBOTIKA DHE AUTOMATIZIMI

Teknologjitë moderne të informacionit mundësojnë krijimin e makinerive që mund të kryejnë në mënyrë të pavarur detyra të ndryshme. Falë zhvillimit të sistemeve kompjuterike, sensorëve dhe sistemeve të kontrollit automatik, robotët sot përdoren në një numër të madh fushash të jetës së përditshme dhe në industri. Ata mund të kryejnë detyra të sakta, të përsëritshme ose të rrezikshme, duke u ardhur në ndihmë njerëzve dhe duke kontribuar në përmirësimin e proceseve të punës.

TASHMË DI...

- të japësh shembuj të teknologjive moderne;
- të shpjegosh se si sistemet informatike përpunojnë informacionin;
- të dallosh përdorimin e teknologjive digjitale në jetën e përditshme;
- të japësh shembuj të automatizimit të proceseve të ndryshme;
- të shpjegosh se si teknologjitë moderne kontribuojnë në zhvillimin e shoqërisë.

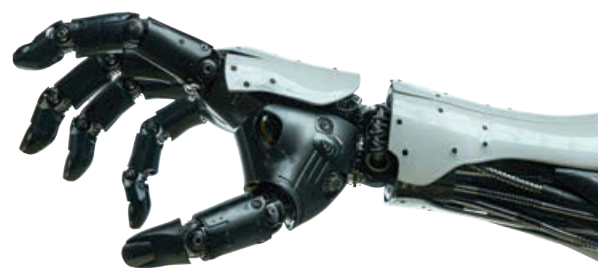
PYETJE PËR REFLEKTIM

1. A mund të kryejnë robotët detyra që më parë i kryenin njerëzit?
2. Në cilat profesione ke parë përdorimin e robotëve?
3. Si mund t'u ndihmojë një robot njerëzve në jetën e përditshme?
4. A kanë robotët gjithmonë pamje njerëzore?
5. Çfarë detyrash do t'ia besojë një roboti?

NË PËRMBAJTJET QË VIJOJNË DO TË MËSOSH:

- të shpjegosh konceptin e robotikës.
- të japësh shembuj të përdorimit të robotëve.
- të kuptosh rolin e automatizimit.
- të lidhësh përdorimin e robotëve me teknologjitë moderne.

Robotika përfaqëson një fushë bashkëkohore shkencore dhe teknike që merret me projektimin, prodhimin dhe zbatimin e sistemeve robotike. Zhvillimi i saj është i lidhur me informatikën, elektronikën, makinerinë dhe automatikën. Sot robotët mund të hasen në fabrika, spitale, në bujqësi, transport, në familje dhe në shumë fusha të tjera. Falë zbatimit të tyre, shumë detyra kryhen më shpejt, më saktë dhe më sigurt.



1.5.1 KONCEPTI I ROBOTIKËS

Robotika përfaqëson një fushë shkencore dhe teknike që merret me krijimin dhe përdorimin e robotëve. Roboti është një makinë që mund të kryejë detyra të përcaktuara paraprakisht, duke përdorur programe dhe informacionin që merr nga mjedisi përreth. Për funksionimin e tyre, robotët përdorin sensorë të ndryshëm, sisteme kontrolli dhe pjesë mekanike, të cilat u mundësojnë të lëvizin, të njohin objekte dhe të kryejnë veprime të caktuara. Në varësi të qëllimit për të cilin përdoren, robotët mund të kenë pamje dhe mundësi të ndryshme.

Robotët nuk është e domosdoshme të kenë pamje njerëzore. Shumica e sistemeve robotike që përdoren sot kanë një formë të përshtatur me detyrat që kryejnë. Në industri, më së shpeshti përdoren krahë robotikë për montimin e produkteve; në familje përdoren robotë për pastrim, ndërsa në bujqësi përdoren makina të automatizuara për monitorimin dhe përpunimin e të mbjellave. Llojet e ndryshme të robotëve krijohen me qëllim që t'u ndihmojnë njerëzve dhe të mundësojnë kryerjen e detyrave më shpejt dhe me saktësi më të madhe.

1.5.2 AUTOMATIZIMI DHE PËRDORIMI I ROBOTËVE

Automatizimi përfaqëson një proces gjatë të cilit detyra të caktuara kryhen në mënyrë automatike, me përfshirje minimale ose pa përfshirje të drejtpërdrejtë të njeriut. Sistemet robotike janë një pjesë e rëndësishme e automatizimit, sepse mund të kryejnë një numër të madh veprimesh të përsëritshme me shpejtësi dhe saktësi të lartë. Në prodhim, robotët montojnë produkte, kryejnë saldim, lyerje dhe paketim, ndërsa në magazina ndihmojnë në klasifikimin dhe transportin e mallrave.

Përdorimi i robotëve nuk kufizohet vetëm në industri. Në mjekësi përdoren sisteme robotike që u ndihmojnë mjekëve gjatë operacioneve të ndërlikuara. Në bujqësi, robotët mund të monitorojnë gjendjen e të mbjellave dhe të kontribuojnë në ujitje më të saktë. Në transport po zhvillohen automjete autonome dhe dronë, ndërsa në familje përdoren robotë për pastrimin dhe mirëmbajtjen e hapësirave. Falë zhvillimit të informatikës, robotët po bëhen pjesë e një numri gjithnjë e më të madh aktiviteteve në shoqërinë moderne.

1.5.3 ROBOTËT MJEKËSORË, DRONËT DHE SISTEMET MODERNE ROBOTIKE

Zhvillimi i robotikës mundëson krijimin e robotëve të specializuar që u ndihmojnë njerëzve në fusha të ndryshme të jetës dhe të punës. Në mjekësi përdoren sisteme robotike që u mundësojnë mjekëve të kryejnë ndërhyrje kirurgjikale të ndërlikuara dhe të sakta. Ato ndihmojnë gjithashtu në rehabilitimin e pacientëve, në transportin e pajisjeve mjekësore dhe në shpërndarjen e barnave në spitale. Falë saktësisë së tyre të lartë, robotët mjekësorë kontribuojnë në përmirësimin e shërbimeve shëndetësore dhe në zvogëlimin e mundësisë për gabime gjatë procedurave të caktuara.

Një grup i veçantë i sistemeve moderne robotike përbëhet nga dronët. Ata janë mjete fluturuese pa pilot, të cilat mund të kontrollohen nga distanca ose të kryejnë në mënyrë automatike detyra të përcaktuara paraprakisht. Dronët përdoren për fotografim dhe filmim, monitorimin e sipërfaqeve bujqësore, vëzhgimin e trafikut, kërkim-shpëtimin, si dhe për dërgimin e produkteve të ndryshme. Në disa vende, dronët përdoren edhe për transportin e materialeve mjekësore në zona të vështira për t'u arritur. Zhvillimi i sistemeve të tilla tregon se robotika ka një zbatim të gjerë në shoqërinë moderne.

1.5.4 ROBOTËT NË INDUSTRIJËN AUTOMOBILISTIKE

Në fabrikat moderne të prodhimit të automjeteve, një numër i madh procesesh pune kryhen me ndihmën e robotëve industrialë. Krahët robotikë kryejnë saldimin, lyerjen, montimin dhe zhvendosjen e pjesëve të rënda me shpejtësi dhe saktësi të lartë. Ata mund të punojnë pa ndërprerje dhe të kryejnë të njëjtën detyrë me cilësi të njëjtë, duke kontribuar kështu në rritjen e prodhimit.

Edhe pse robotët kryejnë një pjesë të madhe të punës, ata nuk i zëvendësojnë plotësisht njerëzit. Për programimin, mirëmbajtjen dhe kontrollin e tyre nevojiten profesionistë me njohuri dhe aftësi të ndryshme. Në këtë mënyrë, robotika krijon mundësi të reja për zhvillimin e profesioneve moderne që lidhen me informatikën, elektronikën dhe inxhinierinë.

1.5.5 PËRPARËSITË DHE KUFIZIMET E ROBOTIKËS

Robotika sjell një sërë përfitimesh për shoqërinë moderne. Robotët mund të kryejnë detyra që kërkojnë saktësi të lartë, shpejtësi ose qëndrueshmëri. Në industri, ata ndihmojnë në prodhimin e produkteve të ndryshme; në mjekësi kontribuojnë në përmirësimin e shërbimeve shëndetësore, ndërsa në bujqësi mundësojnë monitorimin dhe përpunimin më efikas të të mbjellave. Robotët mund të punojnë në mjedise të rrezikshme, si miniera, zona të prekura nga fatkeqësitë natyrore ose gjatë misioneve hapësinore, duke zvogëluar kështu rrezikun për njerëzit. Përdorimi i tyre kontribuon në rritjen e produktivitetit dhe përmirësimin e cilësisë së proceseve të ndryshme të punës.

Pavarësisht përparësive të shumta, sistemet robotike kanë edhe kufizime të caktuara. Ato funksionojnë sipas programeve dhe rregullave të krijuara nga njerëzit dhe nuk mund të zëvendësojnë plotësisht shumë aftësi njerëzore, si krijimtaria, empatia dhe gjykimi moral. Për funksionimin e tyre nevojiten mirëmbajtje e rregullt, mbështetje teknike dhe profesionistë që do t'i programojnë dhe kontrollojnë. Zhvillimi i robotikës ngre gjithashtu çështje të reja lidhur me sigurinë, përgjegjësinë dhe ndikimin e automatizimit në profesione të ndryshme. Prandaj, sistemet robotike duhet të zhvillohen dhe të përdoren në mënyrë të tillë që të kontribuojnë në përmirësimin e jetës dhe të punës së njerëzve.

1.5.6 ROBOTIKA DHE E ARDHMJJA

Zhvillimi i robotikës vazhdon me ritëm të shpejtë dhe pritet që në të ardhmen sistemet robotike të jenë edhe më të pranishme në jetën e përditshme. Në qytetet e zgjuara po zhvillohen automjete autonome, robotë për mirëmbajtjen e hapësirave publike dhe sisteme të automatizuara për menaxhimin e trafikut. Në mjekësi po krijohen robotë që do t'u ndihmojnë mjekëve në diagnostikimin dhe trajtimin e pacientëve, ndërsa në bujqësi po zhvillohen sisteme që do të monitorojnë gjendjen e të mbjellave dhe do të kryejnë automatikisht veprime të ndryshme.

Zhvillimi i robotikës është i lidhur ngushtë me zhvillimin e informatikës dhe të inteligjencës artificiale. Robotët modernë përdorin sensorë të ndryshëm për të mbledhur informacion nga mjedisi përreth dhe sisteme kompjuterike për përpunimin e këtij informacioni. Falë kësaj, ata mund të njohin objekte, të shmangin pengesat dhe të marrin vendime të thjeshta gjatë kryerjes së detyrave të tyre. Në të ardhmen pritet që robotët dhe inteligjenca artificiale të bashkëpunojnë edhe më shumë dhe të krijojnë mundësi të reja për zhvillimin e shoqërisë moderne.



1.5.7 ROBOTËT NË MISIIONET E SHPËTIMIT

Pas tërmeteve, zjarreve dhe fatkeqësive të tjera natyrore, ekipet e shpëtimit shpesh përballen me kushte të rrezikshme pune. Në situata të tilla përdoren robotë dhe dronë të specializuar, të cilët mund të hyjnë në vende ku lëvizja e njerëzve është e vështirë ose e rrezikshme. Ata janë të pajisur me kamera dhe sensorë të ndryshëm, të cilët mundësojnë kërkimin në terren dhe gjetjen e personave që kanë nevojë për ndihmë.

Përdorimi i sistemeve robotike në misionet e shpëtimit kontribuon në mbledhjen më të shpejtë të informacionit dhe në organizimin më të mirë të aktiviteteve të shpëtimit. Njëkohësisht, zvogëlohet rreziku për njerëzit që marrin pjesë në këto operacione. Ky shembull tregon se robotika nuk shërben vetëm për automatizimin e prodhimit, por ka një rol të rëndësishëm edhe në mbrojtjen e jetës së njerëzve.



A E DI?

A e di se në fund të oqeanëve dhe në sipërfaqen e Marsit punojnë robotë të specializuar, të cilët mbledhin të dhëna dhe kryejnë kërkime në vende ku njeriu e ka të vështirë të arrijë?



HULUMTO

Zgjidh një fushë në të cilën përdoren robotë (mjekësi, industri, bujqësi, transport, hapësirë ose misione shpëtimi). Përgatit një raport të shkurtër në të cilin do të shpjegosh:

- çfarë lloj robotësh përdoren;
- cilat detyra kryejnë;
- çfarë përfitimesh ofrojnë;
- me çfarë sfidash përballen.



PYETJE PËR KONTROLLIMIN E NJOHURIVE:

1. Çfarë përfaqëson robotika?
2. Çfarë përfaqëson automatizimi?
3. A kanë të gjithë robotët pamje njerëzore? Arsyeja përgjigjen.
4. Në cilat fusha të jetës së përditshme dhe të punës përdoren robotët?
5. Si kontribuojnë robotët në përmirësimin e prodhimit dhe të shërbimeve?
6. Cilat janë përparësitë më të rëndësishme të përdorimit të sistemeve robotike?
7. Cilat kufizime dhe sfida ekzistojnë gjatë përdorimit të tyre?



DETYRA KËRKIMORE DHE PRAKTIKE

1. Krijë një tabelë në të cilën do të krahasosh katër lloje të ndryshme robotësh (industrialë, mjekësorë, robotë shtëpiakë dhe dronë). Për secilin prej tyre shëno:

- fushën e përdorimit;
- detyrat që kryen;
- përfitimet nga përdorimi i tij.

2. Hulumto se si përdoren robotët në bujqësinë moderne. Përgatit një raport të shkurtër dhe përmend të paktën tri veprimtari që mund të kryejnë.

3. Imagjino sikur shkolla jote blen një robot që do t'u ndihmojë nxënësve dhe mësuesve. Përshkruaj:

- cilat detyra do të kryente;
- cilat teknologji do të përdorte;
- si do ta përmirësonte punën e shkollës.

4. Përgatit një prezantim të shkurtër për përdorimin e robotëve në mjekësi ose në misionet e shpëtimit. Përfshi shembuj dhe shpjego se çfarë përfitimesh ofrojnë.

5. Mendo se cilat profesione mund të përdorin robotë në të ardhmen. Përmend të paktën pesë shembuj dhe shpjego se në ç'mënyrë do t'u ndihmonin njerëzve.



PËRFUNDIM:

Robotika përfaqëson një fushë shkencore dhe teknike që merret me krijimin dhe përdorimin e robotëve. Robotët nuk është e domosdoshme të kenë pamje njerëzore dhe mund të përshtaten për kryerjen e detyrave të ndryshme. Automatizimi mundëson që veprimtari të caktuara të kryhen në mënyrë automatike. Robotët përdoren në industri, mjekësi, bujqësi, transport, në familje dhe në shumë fusha të tjera. Sistemet robotike mund të kryejnë detyra të rrezikshme, të vështira dhe të përsëritshme. Zhvillimi i robotikës është i lidhur ngushtë me informatikën dhe teknologjitë moderne.



PËR ATA QË DËSHIROJNË TË DINË MË SHUMË

Vitet e fundit po zhvillohet një gjeneratë e re robotësh që mund të bashkëpunojnë me njerëzit gjatë kryerjes së detyrave të ndryshme. Këto sisteme përdorin sensorë, kamera dhe zgjidhje moderne informatike për ta njohur dhe kuptuar më mirë mjedisin përreth. Përveç robotëve industrialë, po zhvillohen robotë për ndihmën e të moshuarve, automjete autonome, robotë për eksplorimin e oqeanëve dhe të hapësirës, si dhe robotë që marrin pjesë në eksperimente shkencore. Pritet që robotika edhe në të ardhmen të ketë një rol të rëndësishëm në zhvillimin e shoqërisë moderne.



KONTROLLO NJOHURITË E TUA

1. Rretho përgjigjen e saktë.

Robotika përfaqëson një fushë që merret me:

- a) prodhimin e energjisë elektrike;
- b) krijimin dhe përdorimin e sistemeve robotike;
- c) studimin e planetëve;
- ç) riparimin e kompjuterëve.

2. E saktë ose e pasaktë.

Të gjithë robotët kanë pamje njerëzore.

3. Plotëso.

Procesi gjatë të cilit detyra të caktuara kryhen automatikisht quhet _____.

4. Shpjego konceptet.

Term	Shpjegim
Robot	_____
Robotikë	_____
Automatizim	_____



DETYRË PRAKTIKE

Imagjino sikur në një magazinë duhet të zhvendosen çdo ditë ngarkesa të rënda. Shpjego se si do të mund të ndihmonin sistemet robotike në këtë punë dhe çfarë përfitimesh do të ofronin.



1.6

INTELIGJENCA ARTIFICIALE DHE MËSIMI MAKINERIK

Njerëzit çdo ditë përdorin shërbime të ndryshme digjitale që mund të njohin të folurit, të analizojnë fotografi, të sugjerojnë përmbajtje dhe të ndihmojnë në marrjen e vendimeve. Telefonat celularë mund të njohin fytyrën e përdoruesit, platformat e internetit sugjerojnë filma dhe muzikë në përputhje me interesat e përdoruesve, ndërsa sistemet e navigimit zgjedhin rrugën më të përshtatshme drejt një destinacioni të caktuar. Këto mundësi janë rezultat i zhvillimit të teknologjive moderne të informacionit dhe i zbatimit të inteligjencës artificiale.

TASHMË DI...

- të përmendësh shembuj të teknologjive moderne;
- të shpjegosh se si robotët përdorin informacionin nga mjedisi;
- të dallosh zbatime të automatizimit;
- të përmendësh shembuj të shërbimeve digjitale;
- të shpjegosh se si sistemet informatike përpunojnë të dhënat.

PYETJE PËR REFLEKTIM

1. Si e njeh telefoni celular fytyrën e pronarit të tij?
2. Si sugjerojnë dyqanet në internet produkte që mund të të interesojnë?
3. Si i zgjedhin aplikacionet e navigimit rrugëtimin më të mirë?
4. A mund të mësojnë sistemet kompjuterike nga të dhënat që përpunojnë?
5. Në cilat fusha të jetës së përditshme keni hasur në zbatime të inteligjencës artificiale?

NË PËRMBAJTJET QË VIJOJNË DO TË MËSOSH:

- të shpjegosh çfarë përfaqëson inteligjenca artificiale;
- të shpjegosh çfarë përfaqëson mësimi makinerik;
- të përmendësh shembuj të zbatimit të inteligjencës artificiale;
- të analizosh se si funksionon IA-ja;
- të dallosh përparësitë dhe rreziqet në një shembull të dhënë.

Inteligjenca artificiale përfaqëson një fushë të informatikës që merret me krijimin e sistemeve kompjuterike të afta për të përpunuar informacione dhe për të zgjidhur detyra që zakonisht kërkojnë inteligjencë njerëzore. Për punën e tyre, këto sisteme përdorin sasi të mëdha të dhënash dhe algoritme të ndryshme për analizimin dhe përpunimin e informacioneve. Falë kësaj, ato mund të njohin modele, të parashikojnë rezultate të mundshme dhe t'u ndihmojnë njerëzve gjatë kryerjes së aktiviteteve të ndryshme.

1.6.1 ÇFARË PËRFAQËSON INTELIGJENCA ARTIFICIALE?

Inteligjenca artificiale është një fushë e informatikës që merret me zhvillimin e sistemeve kompjuterike të afta për të kryer detyra që zakonisht kërkojnë të menduarit njerëzor. Sisteme të tilla mund të njohin imazhe, të përpunojnë të folurën, të analizojnë të dhëna dhe të ndihmojnë në marrjen e vendimeve të caktuara. Për të funksionuar, ato përdorin informacione që marrin nga burime të ndryshme dhe zbatojnë algoritme të veçanta për përpunimin e tyre. Qëllimi kryesor i inteligjencës artificiale është krijimi i sistemeve që mund të zgjidhin në mënyrë më efikase detyra të ndryshme praktike.

Përdorimi i inteligjencës artificiale sot është i pranishëm në shumë fusha të jetës së përditshme. Gjatë përdorimit të motorëve të kërkimit në internet, rrjeteve sociale, platformave për muzikë dhe filma, si dhe telefonave celularë, përdoruesit çdo ditë shfrytëzojnë shërbime që përdorin algoritme të inteligjencës artificiale. Në mjekësi, këto sisteme ndihmojnë në analizimin e imazheve mjekësore; në transport kontribuojnë në zhvillimin e automjeteve autonome; ndërsa në arsim mundësojnë krijimin e sistemeve interaktive për të nxënë. Falë këtyre përdorimeve, inteligjenca artificiale po bëhet një pjesë e rëndësishme e shoqërisë moderne.

1.6.2 ÇFARË PËRFAQËSON MËSIMI MAKINERIK?

Mësimi makinerik është një fushë e inteligjencës artificiale që u mundëson sistemeve kompjuterike të mësojnë nga të dhënat që përpunojnë. Në vend që për çdo situatë të programohen rregulla të veçanta, sistemi analizon një numër të madh shembujsh dhe gradualisht zbulon rregullsi dhe modele të caktuara. Sa më shumë të dhëna të përshtatshme të përpunojë sistemi, aq më mirë mund t'i kryejë detyrat për të cilat është krijuar.

Mësimi makinerik ka zbatim të gjerë në sistemet informatike moderne. Ai përdoret për njohjen e të folurit, përkthimin automatik të teksteve, njohjen e fytyrave dhe objekteve në fotografi, zbulimin e mesazheve elektronike të padëshiruara dhe rekomandimin e përmbajtjeve të ndryshme në internet. Për shembull, kur një përdorues shikon video në internet, sistemi analizon se cilat përmbajtje zgjedh më shpesh dhe i propozon video të reja me tematikë të ngjashme. Në këtë mënyrë, mësimi makinerik i ndihmon sistemet kompjuterike t'i kuptojnë më mirë nevojat e përdoruesve dhe të ofrojnë shërbime më të dobishme.

1.6.3 PËRDORIMI I INTELIGJENCËS ARTIFICIALE

Inteligjenca artificiale sot përdoret në një numër të madh fushash të shoqërisë moderne. Në shëndetësi, ajo ndihmon në analizimin e të dhënave dhe imazheve mjekësore; në transport kontribuon në zhvillimin e sistemeve për menaxhimin e trafikut dhe të automjeteve autonome; ndërsa në arsim përdoret për krijimin e përmbajtjeve interaktive dhe sistemeve për të nxënë. Në bujqësi përdoret për monitorimin e gjendjes së të mbjellave, ndërsa në industri ndihmon në automatizimin e proceseve të prodhimit dhe në kontrollin e cilësisë së produkteve.

Një numër i madh shërbimesh digjitale që njerëzit përdorin çdo ditë zbatojnë metoda të ndryshme të inteligjencës artificiale. Motorët e kërkimit në internet ndihmojnë në gjetjen e informacionit, sistemet e navigimit sugjerojnë rrugë të përshtatshme, ndërsa platforma të ndryshme rekomandojnë muzikë, filma ose përmbajtje të tjera në bazë të interesave të përdoruesve. Gjatë përpunimit të fotografive dhe videove, sistemet moderne mund të njohin fytyra dhe objekte, ndërsa gjatë përdorimit të telefonave celularë mund të ndihmojnë në njohjen e të folurit dhe shndërrimin e tij në tekst. Këta shembuj tregojnë se inteligjenca artificiale tashmë është bërë pjesë e jetës së përditshme.

1.6.4 SI FUNKSIONON INTELIGJENCA ARTIFICIALE?

Sa herë që telefoni yt zhblokohet përmes njohjes së zërit ose kur YouTube të sugjeron videon tënde të radhës të preferuar, ti po përdor fuqinë e inteligjencës artificiale (IA). Edhe pse në pamje të parë mund të duket si magji, pas këtyre proceseve qëndrojnë hapa të saktë matematikorë, analiza e të dhënave dhe algoritme të mësimit makinerik. Për të kuptuar se si funksionon në të vërtetë inteligjenca artificiale, do të analizojmë dy shembuj realë që i përdorim çdo ditë.

SHEMBULLI 1: SISTEMET E REKOMANDIMEVE (Si TikTok, Netflix ose Spotify se çfarë të pëlqen?)

A je pyetur ndonjëherë se si arrin një aplikacion ta mbajë vëmendjen tënde për orë të tëra? Ai përdor një sistem rekomandimesh të bazuar në inteligjencën artificiale (IA). Ky sistem nuk bën rekomandime rastësisht, por funksionon përmes katër fazave kryesore:

1. **Mbledhja e të dhënave (gjurma jote digjitale):** Gjatë përdorimit të aplikacionit, algoritmi analizon me kujdes veprimet e tua. Ai mbledh të dhëna të tilla si: në cilat publikime ndalesh më gjatë, cilat video i kalon pas sekondave të para, çfarë përmbajtjesh të pëlqejnë, çfarë shpërndan ose komenton.
2. **Njohja e modeleve (zbulimi i modeleve):** IA-ja i analizon këto të dhëna dhe të grupon me mijëra përdorues të tjerë që kanë zakone të ngjashme. Për shembull, nëse ti dhe përdoruesi Marko shikoni të njëjtat tri video për gatim, IA-ja ju klasifikon si „binjakë digjitalë“, me interesa të ngjashme.
3. **Filtrimi dhe parashikimi:** Nëse Marko shikon një video të katërt, për shembull një recetë të re për pica, dhe e shikon deri në fund, algoritmi llogarit matematikisht se ekziston një mundësi prej 95% që edhe ty të të pëlqejë ajo video.
4. **Rezultati:** Videoja për picën shfaqet në faqen tënde kryesore. Nëse e shikon, sistemi merr konfirmimin se llogaritja e tij ishte e saktë dhe algoritmi bëhet edhe më i zgjuar për rekomandimin e ardhshëm.

SHEMBULLI 2: Njohja e të folurit (Si funksionojnë Siri, Google Assistant ose diktimi i mesazheve?)

Kur do të thuash „Hey Siri“ ose kur përdor opsionin për shndërrimin e të folurit në tekst (Voice-to-Text), inteligjenca artificiale duhet të ndërmjetësojë mes botës njerëzore (biologjike) dhe asaj kompjuterike (digjitale). Ky proces kompleks zhvillohet në mënyrën e mëposhtme:

1. **Analiza akustike:** Zëri yt është një valë zanore. Mikrofonit i pajisjes e regjistron këtë valë dhe e shndërron atë në një sinjal digjital (një varg numrash). Inteligjenca artificiale e pastron këtë sinjal nga zhurmat e ambientit (për shembull, zhurma e automjeteve në rrugë).
2. **Ndarja në fonema:** IA-ja e ndan të folurin e regjistruar në njësitë më të vogla të mundshme zanore, të quajtura fonema. Për shembull, nëse thoni fjalën „kompjuter“, algoritmi e njeh atë si një bashkim të tingujve k-o-m-p-j-u-t-e-r.
3. **Modelimi gjuhësor:** Ky është një nga pjesët kryesore të IA-së. Algoritmi i krahason këto tinguj me një bazë të madhe të dhënash (miliona orë të regjistruara të të folurit njerëzor), për të përcaktuar se cila fjalë përputhet më së shumti me tingullin e dëgjuar. Ai gjithashtu analizon kontekstin e fjalisë. Nëse thoni „Dua një mollë“, IA-ja e kupton se bëhet fjalë për frutin dhe jo për kompaninë teknologjike Apple, falë fjalës së mëparshme „dua“.
4. **Ekzekutimi i komandës:** Pasi IA-ja ta kuptojë tekstin, ajo aktivizon funksionin përkatës në telefon – e shkruan fjalën në ekran, e lëshon këngën që i ke kërkuar ose i përgjigjet pyetjes me një zë të gjeneruar.

Nga këta dy shembuj kuptojmë se IA-ja nuk „mendon“ si njeriu, me emocione dhe vetëdije. Ajo funksionon si një përpunues i fuqishëm statistikor. Parimi është gjithmonë i njëjtë: Të dhënat hyrëse (pëlqimet ose zëri

yt)→**përpunimi matematikor dhe krahasimi me njohuritë e mëparshme** → rezultati dalës (një rekomandim i ri ose tekst i shkruar).

Sa më shumë të dhëna të ketë në dispozicion IA-ja, aq më saktë do të funksionojë në shembujt e përdorimit të përditshëm.

1.6.5 PËRPARËSITË DHE RREZIQET E INTELIGJENCËS ARTIFICIALE

Inteligjenca artificiale (IA) është, pa dyshim, një nga risitë teknologjike më të rëndësishme në historinë e njerëzimit. Ajo nuk është më vetëm një teori e fantashkencës, por një mjet që po transformon në mënyrë aktive mjekësinë, arsimin, biznesin dhe jetën tonë të përditshme. Megjithatë, si çdo teknologji e fuqishme, IA-ja mbart një potencial të madh për përparim, por edhe sfida serioze. Për të qenë të përgatitur për të ardhmen, duhet t'i analizojmë në mënyrë të hollësishme të dyja anët e „medaljes“: përparësitë dhe rreziqet e mundshme nga përdorimi i saj.

PËRPARËSITË E PËRDORIMIT TË IA-së

Zbatimi i IA-së në shoqëri sjell përfitime revolucionare, të cilat e bëjnë jetën më të lehtë, më të sigurt dhe më efikase.

- **Automatizimi dhe efikasiteti i jashtëzakonshëm:** IA-ja mund të përpunojë miliarda të dhëna brenda pak sekondash dhe të kryejë detyra të përsëritshme (rutinore/të mërzitshme) pa pushim. Kjo u mundëson njerëzve të çlirohen nga punët administrative dhe të përqendrohen në të menduarit krijues dhe në zgjidhjen e problemeve komplekse.
- **Përparimi në mjekësi dhe shpëtimi i jetëve:** Algoritmet e IA-së mund të analizojnë imazhet mjekësore (si radiografite ose imazhet e rezonancës magnetike) me një saktësi më të madhe se syri i njeriut, duke zbuluar sëmundjet në fazat e tyre më të hershme. Gjithashtu, IA-ja e përshpejton ndjeshëm procesin e zbulimit dhe testimit të barnave të reja.
- **Të nxënët e personalizuar:** Në arsim, sistemet inteligjente mund ta përshtatin materialin mësimor sipas ritmit dhe mënyrës së të nxënët të secilit nxënës. Nëse një nxënës „ngec“ në një temë të caktuar, asistenti i IA-së mund të ofrojë shpjegime alternative dhe ushtrime shtesë.
- **Siguri e shtuar në mjedise të rrezikshme:** Robotët e kontrolluar nga IA-ja mund të kryejnë detyra që janë vdekjeprurëse për njerëzit, si çminimi i zonave, eksplorimi i thellësive të oqeanëve, shuarja e zjarreve të mëdha ose puna në centrale bërthamore.

RREZIQET E MUNDSHME NGA PËRDORIMI I IA-së

Pavarësisht të gjitha përparësive, zhvillimi i pakontrolluar i inteligjencës artificiale ngre çështje serioze etike, sociale dhe të sigurisë.

- **Ndikimi në tregun e punës (humbja e vendeve të punës):** Zhvillimi i shpejtë i IA-së shkakton frikën nga papunësia masive. Sistemet e bazuara në IA mund të zëvendësojnë profesione në fusha si: mbështetja e klientëve, përkthimi, drejtimi i automjeteve (automjetet autonome), madje edhe programuesit më të rinj ose shkrimtarët, gjë që u imponon njerëzve nevojën urgjente për rikualifikim.



- Paragjykimet dhe diskriminimi (algorithmic bias): IA-ja mëson nga të dhënat e krijuara nga njerëzit. Nëse këto të dhëna përmbajnë paragjykime historike ose diskriminim (në baza racore, gjinore ose kombëtare), IA-ja do t'i mësojë këto paragjykime dhe do të vazhdojë të marrë vendime të padrejta – për shembull, gjatë përzgjedhjes së kandidatëve për punë ose gjatë miratimit të kredive bankare.
- Humbja e privatësisë: Për të funksionuar me saktësi, IA-së i nevojiten sasi të mëdha të të dhënave personale. Një rrezik serioz është mbikëqyrja e vazhdueshme digjitale, ku çdo hap, bisedë, kërkim dhe emocion i yni gjurmohet, analizohet dhe mund të keqpërdoret për qëllime komerciale ose politike.
- Përhapja e dezinformatave dhe „Deepfakes“: Mjetet e avancuara të IA-së mundësojnë krijimin e lehtë të lajmeve të rreme, teksteve bindëse, madje edhe të videove dhe regjistrimeve audio të rreme, në të cilat figura publike ose njerëz të zakonshëm duken sikur thonë gjëra që nuk i kanë thënë kurrë. Kjo e rrezikon besimin në informacionet dhe mund të shkaktojë paqëndrueshmëri politike.
- Dembelia kognitive të njerëzit: Mbështetja e tepruar të IA-ja për shkrimin e detyrave të shtëpisë, zgjidhjen e problemeve dhe marrjen e vendimeve mund të dobësojë, në planin afatgjatë, të menduarit kritik, aftësitë logjike dhe krijuese të brezave të rinj.

Pavarësisht mundësive të shumta, përdorimi i inteligjencës artificiale shoqërohet edhe me rreziqe të caktuara. Funksionimi i saj varet nga cilësia dhe sasia e të dhënave që përpunon, si dhe nga rregullat sipas të cilave është krijuar. Sistemet moderne mund të bëjnë gabime ose të japin rezultate të pasakta, prandaj nevojiten verifikimi dhe mbikëqyrja nga njerëzit. Përdorimi i përgjegjshëm i inteligjencës artificiale nënkupton përdorimin e saj si një teknologji ndihmëse që kontribuon në përmirësimin e aktiviteteve dhe shërbimeve të ndryshme.

Inteligjenca artificiale duhet të përdoret në mënyrë krijuese, etike dhe të përgjegjshme. Çelësi për zbatimin e saj të suksesshëm qëndron në miratimin e ligjeve dhe rregulloreve të rrepta etike, të cilat do të mbrojnë të drejtat e njeriut dhe privatësinë, duke nxitur njëkohësisht përparimin teknologjik. Qëllimi i njerëzimit nuk duhet të jetë zëvendësimi i njeriut me IA-në, por krijimi i një sinergie, në të cilën IA-ja do ta ndihmojë njeriun të arrijë më shumë.

1.6.6 INTELIGJENCA ARTIFICIALE DHE SHOQËRIA BASHKËKOHORE

Inteligjenca artificiale ka një rëndësi gjithnjë e më të madhe në zhvillimin e shoqërisë bashkëkohore. Zbatimi i saj kontribuon në përmirësimin e shërbimeve të ndryshme dhe në lehtësimin e aktiviteteve të përditshme të njerëzve. Në shëndetësi, ajo ndihmon në analizimin e të dhënave mjekësore; në arsim mundëson krijimin e mjeteve moderne digjitale për të nxënëit, ndërsa në transport kontribuon në menaxhimin më të mirë të sistemeve të trafikut. Në industri përdoret për automatizimin e prodhimit dhe kontrollin e cilësisë, ndërsa në bujqësi ndihmon në monitorimin e gjendjes së kulturave bujqësore dhe në përdorimin më efikas të burimeve.

Zhvillimi i inteligjencës artificiale hap mundësi të reja për kërkime shkencore, zhvillim ekonomik dhe përmirësimin e cilësisë së jetës. Në të njëjtën kohë, është e nevojshme që teknologjitë moderne të përdoren në mënyrë të përgjegjshme dhe të kujdesshme. Njerëzit kanë një rol të rëndësishëm në zhvillimin, kontrollin dhe zbatimin e këtyre sistemeve, pasi ata përcaktojnë rregullat sipas të cilave funksionojnë ato. Prandaj, krahas njohurive teknike, është e rëndësishme të zhvillohen edhe të menduarit kritik, shkrim-leximi digjital dhe përdorimi i përgjegjshëm i teknologjive informatike.

Inteligjenca artificiale dhe e ardhmja

Me zhvillimin e informatikës pritet që inteligjenca artificiale të gjejë zbatim në një numër gjithnjë e më të madh fushash. Sistemet moderne do të ndihmojnë në kërkimet shkencore, zhvillimin e barnave të reja, përmirësimin e arsimit, mbrojtjen e mjedisit jetësor dhe menaxhimin e sistemeve teknike komplekse.

Në shumë profesione, inteligjenca artificiale do të jetë një mjet që do t'u ndihmojë njerëzve të përpunojnë informacionet më shpejt dhe të zgjidhin më lehtë detyra të ndryshme.

Pavarësisht zhvillimit të shpejtë të teknologjisë, njeriu mbetet faktori më i rëndësishëm në marrjen e vendimeve dhe në zbatimin e sistemeve bashkëkohore. Inteligjenca artificiale përfaqëson një mjet që mund të kontribuojë në përmirësimin e aktiviteteve të ndryshme, por nuk mund ta zëvendësojë plotësisht dijen, kreativitetin, përvojën dhe përgjegjësinë njerëzore. Prandaj, zhvillimi i ardhshëm i kësaj fushe do të varet nga bashkëpunimi i suksesshëm ndërmjet njerëzve dhe teknologjive moderne informatike.

Inteligjenca artificiale në arsim

Platformat moderne arsimore përdorin teknologji të ndryshme të inteligjencës artificiale për t'u ndihmuar nxënësve dhe mësuesve. Sisteme të tilla mund të ofrojnë materiale shitesë për të nxënit, të ndihmojnë në gjetjen e informacionit dhe të mundësojnë aktivitete interaktive për përvetësimin e përmbajtjeve mësimore. Në këtë mënyrë, nxënësit mund të kenë qasje më të lehtë në burime të ndryshme të njohurive dhe ta organizojnë më mirë procesin e të nxënit.

Inteligjenca artificiale nuk e zëvendëson mësuesin, por përfaqëson një mjet shtesë që mund të kontribuojë në përmirësimin e procesit mësimor. Mësuesi ka një rol të rëndësishëm në përzgjedhjen e përmbajtjeve mësimore, orientimin e nxënësve dhe verifikimin e saktësisë së informacionit. Përdorimi i përgjegjshëm i teknologjive moderne kontribuon në zhvillimin e kompetencave digjitale dhe të të menduarit kritik te nxënësit.



A E DI SE...?

A e di se çdo ditë përdor sisteme që zbatojnë inteligjencën artificiale? Motorët e kërkimit në internet, aplikacionet e navigimit, sistemet për njohjen e të folurit dhe shumë shërbime të tjera digjitale përdorin metoda të ndryshme për përpunimin e të dhënave dhe ndihmojnë në zgjidhjen e detyrave të ndryshme.



HULUMTO

Hulumto pesë shembuj të zbatimit të inteligjencës artificiale në jetën e përditshme. Për secilin shembull shpjego:

- ku përdoret;
- çfarë detyre kryen;
- si ndihmon njerëzit.



PYETJE PËR PËRSËRITJE

1. Çfarë përfaqëson inteligjenca artificiale?
2. Çfarë përfaqëson mësimi makinerik?
3. Në cilat fusha të shoqërisë bashkëkohore zbatohet inteligjenca artificiale?
4. Në ç'mënyrë ndihmon inteligjenca artificiale në jetën e përditshme?
5. Cilat janë përparësitë më të rëndësishme të zbatimit të inteligjencës artificiale?
6. Cilat kufizime ekzistojnë gjatë përdorimit të sistemeve me inteligjencë artificiale?
7. Pse është e rëndësishme që inteligjenca artificiale të përdoret në mënyrë të përgjegjshme?





DETYRA HULUMTUESE DHE PRAKTIKE

1. Bëj një listë me dhjetë shembuj të zbatimit të inteligjencës artificiale në jetën e përditshme. Për secilin shembull, trego ku përdoret dhe çfarë detyre kryen.
2. Hulumto se si përdoret inteligjenca artificiale në mjekësi, në arsim ose në bujqësi. Përgatit një raport të shkurtër.
3. Imagjino se shkolla jote dëshiron të vendosë një sistem me inteligjencë artificiale. Propozo tri aktivitete në të cilat ai mund të ndihmojë dhe shpjego përfitimet.
4. Bëj një tabelë në të cilën do të krahasosh njeriun dhe një sistem me inteligjencë artificiale sipas:
 - shpejtësinë e përpunimit të të dhënave;
 - të nxënës;
 - kreativitetin;
 - marrjen e vendimeve;
 - nevojën për mbikëqyrje njerëzore.
5. Vlerëso përparësitë e përdorimit të inteligjencës artificiale në mjekësi dhe në arsim përballë rrezikut të mundshëm të humbjes së vendeve të punës në të ardhmen. A mendon se përfitimet e justifikojnë zhvillimin e kësaj teknologjie? Arsyejo qëndrimin tënd.
6. Duke pasur parasysh rreziqet që lidhen me privatësinë e të dhënave dhe shfaqjen e videove dhe regjistrimeve audio të rreme, të ashtuquajtura „Deepfake“, jep një gjykim: Cilat janë sfidat etike nga përdorimi i inteligjencës artificiale që janë më të rrezikshme për brezat e rinj dhe si ndikojnë ato në aftësinë e tyre për të menduar në mënyrë kritike?
7. Mendo dhe shkruaj një tekst të shkurtër me temë:

„Si mund t’ua përmirësojë inteligjenca artificiale jetën e përditshme njerëzve?”



MBAJ MEND ÇFARË MËSOVE

Inteligjenca artificiale është një fushë e informatikës që merret me krijimin e sistemeve kompjuterike inteligjente.

Mësimi makinerik është pjesë e inteligjencës artificiale që mundëson të mësuarit nga të dhënat.

Inteligjenca artificiale zbatohet në shëndetësi, arsim, industri, transport, bujqësi dhe në shumë fusha të tjera.

Shërbimet moderne digjitale përdorin çdo ditë metoda të ndryshme të inteligjencës artificiale.

Inteligjenca artificiale sjell shumë përfitime, por ka edhe kufizime të caktuara.

Përdorimi i përgjegjshëm i inteligjencës artificiale ka një rol të rëndësishëm në zhvillimin e shoqërisë bashkëkohore.



PËR ATA QË DËSHIROJNË TË DINË MË SHUMË

Zhvillimi i inteligjencës artificiale përfaqëson një nga fushat më dinamike të informatikës bashkëkohore. Sot krijohen sisteme që mund të përpunojnë sasi të mëdha të të dhënave, të ndihmojnë në kërkimet shkencore dhe të kontribuojnë në zgjidhjen e problemeve të ndërlikuara. Inteligenca artificiale përdoret në kërkimet e hapësirës, në zhvillimin e barnave të reja, në mbrojtjen e mjedisit dhe në shumë aktivitete të tjera. Me zhvillimin e mëtejshëm të teknologjisë pritet që përdorimi i saj të vazhdojë të zgjerohet, ndërsa njeriu do të ketë një rol të rëndësishëm në kontrollin dhe përdorimin e përgjegjshëm të këtyre sistemeve.



VERIFIKO NJOHURITË E TUA

1. Rretho përgjigjen e saktë.

Inteligjenca artificiale përfaqëson:

- a) programë për përpunimin e tekstit;
- b) fushë e informatikës që krijon sisteme kompjuterike inteligjente;
- c) lojë kompjuterike;
- ç) sistem operativ.

2. E saktë ose e pasaktë.

Mësimi makinerik është pjesë e inteligjencës artificiale.

3. Plotëso.

Fusha që mundëson sistemeve kompjuterike të mësojnë nga të dhënat quhet _____.

4. Shpjego termat.

Term	Shpjegim
Inteligjenca artificiale	_____
Mësimi makinerik	_____
Algoritëm	_____



DETYRË E BAZUAR NË SITUATË

Imagjino se një shkollë dëshiron të përdorë një sistem me inteligjencë artificiale për përmirësimin e mësimdhënies. Përmend dy aktivitete në të cilat sistemi mund të ndihmojë dhe shpjego pse është e nevojshme mbikëqyrja njerëzore gjatë përdorimit të tij.



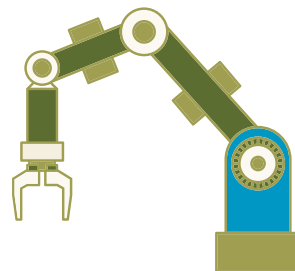
1.7

MODELE TË MËDHA GJUHËSORE DHE INTELIGJENCA ARTIFICIALE GJENERATIVE

Modelet e mëdha gjuhësore, të njohura me shkurtesën në anglisht LLM (Large Language Models), përfaqësojnë sisteme të avancuara të inteligjencës artificiale, të trajnuara me sasi të mëdha të të dhënave tekstuale, me qëllim që të kuptojnë, përpunojnë dhe interpretojnë gjuhën njerëzore. Aftësia e tyre kryesore dhe më mbresëlënëse është gjenerimi i tekstit, ku modeli, bazuar në një pyetje ose në një kontekst të dhënë, mund të shkruajë në mënyrë të pavarur ese, artikuj, poezi ose shpjegime të hollësishme që tingëllojnë natyrshëm, sikur të jenë shkruar nga një njeri. Kjo bazë teknologjike mundësoi krijimin e sistemeve inteligjente të bisedës (chatbot), si ChatGPT, të cilat mundësojnë komunikim interaktiv në kohë reale dhe japin përgjigje të menjëhershme ndaj pyetjeve të ndryshme të përdoruesve. Falë fleksibilitetit të tyre, këto modele përdoren gjerësisht në arsim, ku shërbejnë si tutorë digjitalë të personalizuar, duke i ndihmuar nxënësit të përvetësojnë mësimet të vështira ose të përkthejnë nga një gjuhë në tjetrën. Njëkohësisht, ato kanë sjellë një transformim të madh edhe në botën e teknologjisë, duke mundësuar përdorim të avancuar në programim, ku zhvilluesit i përdorin për shkrimin automatik, rishikimin, gjetjen dhe korrigjimin e gabimeve (debugging), si dhe shpjegimin e kodit kompjuterik.

TASHMË DI ...

- të shpjegosh se çfarë përfaqëson inteligjenca artificiale;
- të shpjegosh se çfarë përfaqëson mësimi makinerik;
- të japësh shembuj të zbatimit të inteligjencës artificiale;
- të analizosh përfitimet dhe kufizimet e inteligjencës artificiale;
- të njohësh zbatimin e teknologjive moderne të informacionit.

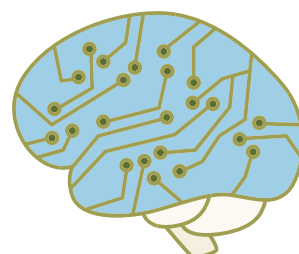


PYETJE PËR TË MENDUAR

1. Si mundet një sistem kompjuterik t'i përgjigjet një pyetjeje të parashtruar në gjuhë natyrore?
2. Si është e mundur që një kompjuter të krijojë tekst, imazh ose muzikë?
3. A jep sistemi kompjuterik gjithmonë informacion të saktë?
4. Si mund të ndihmojë inteligjenca artificiale gjenerative në arsim?
5. Pse është e rëndësishme që informacioni i marrë nga sistemet moderne digjitale të verifikohet në mënyrë kritike?

NË PËRMBAJTJET QË VIJOJNË DO TË MËSOSH:

- të shpjegosh se çfarë përfaqësojnë modelet e mëdha gjuhësore;
- të japësh shembuj të zbatimit të tyre;
- të kuptosh se si përdoren për gjenerimin e tekstit;
- të identifikosh mundësitë e përdorimit të tyre në mësim dhe në punë.



Vitet e fundit, inteligjenca artificiale është zhvilluar ndjeshëm dhe ka fituar mundësi të reja për përpunimin dhe krijimin e llojeve të ndryshme të përmbajtjes. Sistemet moderne kompjuterike janë në gjendje të kuptojnë pyetje të parashtruara në gjuhë natyrore, të krijojnë tekste, të ndihmojnë në përkthime, të hartojnë përmbledhje dhe të gjenerojnë ide të ndryshme. Falë këtyre aftësive, ato gjejnë zbatim në arsim, shkencë, biznes dhe në shumë fusha të tjera.

Modelet e mëdha gjuhësore përfaqësojnë sisteme moderne të inteligjencës artificiale, të cilat trajnohen përmes përpunimit të sasive të mëdha të të dhënave tekstuale. Falë këtij procesi, ato janë në gjendje të identifikojnë modele gjuhësore dhe të krijojnë përmbajtje të reja në përputhje me kërkesat e përdoruesve. Inteligjenca artificiale gjeneruese jo vetëm që përpunon informacionin, por mund të krijojë tekste, imazhe, muzikë dhe lloje të tjera të përmbajtjes digjitale. Për shkak të përdorimit të gjerë të këtyre teknologjive, është e rëndësishme që përdoruesit t'i kuptojnë si mundësitë e tyre, ashtu edhe kufizimet.

1.7.1 ÇFARË JANË MODELET E MËDHA GJUHËSORE?

Modelet e mëdha gjuhësore përfaqësojnë sisteme moderne kompjuterike të afta për të përpunuar dhe krijuar tekste në gjuhë natyrore. Për funksionimin e tyre, ato përdorin sasi të mëdha të të dhënave dhe algoritme të ndryshme të inteligjencës artificiale, me ndihmën e të cilave identifikojnë lidhjet ndërmjet fjalëve dhe fjalive. Falë kësaj, ato mund t'u përgjigjen pyetjeve, të shpjegojnë koncepte, të përkthejnë tekste dhe të ndihmojnë në krijimin e përmbajtjeve të ndryshme.

Modelet e mëdha gjuhësore (LLM) janë sisteme shumë të avancuara të inteligjencës artificiale, të projektuara për të kuptuar dhe gjeneruar gjuhën njerëzore. Fjala „të mëdha“ nuk është e rastësishme. Ajo nënkupton dy gjëra:

- **Një bazë jashtëzakonisht e madhe të dhënash.** Këto modele janë trajnuar me miliarda faqe tekstesh nga interneti – libra, artikuj, Wikipedia, punime shkencore dhe kode kompjuterike.
- **Miliarda parametrash.** Ato kanë një strukturë të ndërlikuar të brendshme (rrjet nervor digjital), e cila u mundëson të lidhin fjalët me njëra-tjetrën dhe të kuptojnë kontekstin e fjalive pothuajse po aq mirë sa njerëzit.

Thënë në mënyrën më të thjeshtë, LLM-ja është si një lexues digjital jashtëzakonisht inteligjent, i cili ka lexuar pothuajse të gjithë librat e botës dhe tani mund të bisedojë me ty për çdo temë.

Modelet e mëdha gjuhësore gjejnë zbatim në arsim, shkencë, biznes dhe në shërbime të ndryshme digjitale. Ato mund të ndihmojnë në kërkimin e informacionit, përgatitjen e teksteve, mësimin e gjuhëve të huaja dhe organizimin e aktiviteteve të ndryshme. Megjithatë, është e rëndësishme të kuptohet se këto sisteme nuk mendojnë në të njëjtën mënyrë si njerëzit, por krijojnë përmbajtje në bazë të të dhënave të përpunuara dhe algoritmeve mbi të cilat funksionojnë.

Sot ekzistojnë disa gjigantë kryesorë të teknologjisë që zhvillojnë mjetet LLM më të fuqishme në botë. Me siguri, tashmë përdor disa prej tyre:

- **ChatGPT (OpenAI):** Mjeti i parë që solli një revolucion në përdorimin publik të inteligjencës artificiale. Është veçanërisht i përshtatshëm për biseda të përditshme, shkrim kreativ dhe shpjegimin e koncepteve komplekse.
- **Gemini (Google):** Model i lidhur drejtpërdrejt me motorin e kërkimit të Google, gjë që i mundëson qasje të shpejtë në informacionet më të reja dhe më të sakta të disponueshme në internet, në kohë reale.
- **Claude (Anthropic):** Mjet i njohur për stilin e tij jashtëzakonisht natyral të të shkruarit, aftësinë për arsyetim logjik dhe përpunimin e dokumenteve shumë të mëdha dhe të gjata.
- **Copilot (Microsoft):** Asistent i integruar në programet që i përdor për shkollë (Word, PowerPoint) dhe që ndihmon gjatë shkrimit dhe dizajnit në to.



1.7.2 SI FUNKSIONON GJENERIMI I TEKSTIT?

Shumë njerëz mendojnë se modelet LLM „mendojnë“ ose se e dinë të vërtetën. Realiteti është ndryshe – ato janë makineri të fuqishme statistikore për parashikim.

Kur i parashtron një pyetje modelit (kjo pyetje quhet instruksion (prompt)), ai nuk e kërkon përgjigjen në bazën e të dhënave për ta kopjuar dhe „ngjitur“ atë si një përgjigje të gatshme. Përkundrazi, ai vepron në këtë mënyrë:

- E analizon pyetjen tënde fjalë për fjalë.
- Në bazë të gjithçkaje që ka lexuar më parë, llogarit se cila fjalë duhet të vijojë në mënyrë logjike.
- Për shembull, nëse fjalia fillon me: „Kryeqyteti i Maqedonisë është...“, modeli parashikon matematikisht se, me një probabilitet prej 99,9%, fjala vijuese është „Shkupi“.

Procesi zhvillohet me një shpejtësi të jashtëzakonshme, fjalë pas fjale, duke krijuar një tekst që duket sikur po e shkruan një njeri i vërtetë në kohë reale.

Modelet LLM mund t'i përdorësh si asistentin tënd personal, i cili është i disponueshëm 24/7.

Në procesin e të nxënës, ato mund të shërbejnë si super-tutorë personalë:

- Shpjegimi i mësimave komplekse. Nëse nuk e kupton një mësim të fizikës ose kimisë, mund t'i thosh modelit: „Ma shpjego Ligjin e Dytë të Njutonit sikur të isha 10 vjeç dhe më jep një shembull nga jeta e përditshme.“
- Mësimi i gjuhëve të huaja. Mund të zhvillosh një bisedë me ChatGPT në anglisht ose në gjermanisht, ndërsa në përgjigjet e tij modeli mund të t'i korrigjojë gabimet gramatikore.
- Përgatitja për teste. Mund t'ia ngarkosh shënimet e tua dhe t'i thuash: „Më përgatit një kuiz me 5 pyetje me alternativa, në mënyrë që të testoj njohuritë e mia.“

Në punë dhe në programim:

- Shkrimi dhe korrigjimi i kodit. Nëse mëson programim në shkollë (për shembull, Python ose C++), LLM-ja mund të të tregojë menjëherë se ku mungon pikëpresja (;) ose pse kodi nuk funksionon.
- Gjenerimi i ideve. Nëse duhet të përgatisësh një projekt ose ese, por nuk e di se si të fillosh, mjeti mund të të ofrojë pesë struktura dhe tema të ndryshme për shkrim.
- Automatizimi i detyrave rutinë. Në biznes, këto modele përdoren për t'iu përgjigjur shpejt e-maileve, për të hartuar raporte dhe për të përkthyer kontrata të gjata.

Rregull i rëndësishëm: Mjetet LLM janë bashkëpunëtorë të shkëlqyer, por zëvendësues të dobët të aftësive tënde njohëse. Ato ndonjëherë mund të „halucinojnë“ (të krijojnë fakte të pasakta me vetëbesim të lartë). Prandaj, gjithmonë verifiko informacionet e rëndësishme dhe përdori këto mjete për të përmirësuar njohuritë e tua, jo për të anashkaluar procesin e të menduarit!

Modelet e mëdha gjuhësore dhe inteligjenca artificiale gjeneruese gjejnë zbatim në fusha të ndryshme të shoqërisë bashkëkohore. Në biznes, ato përdoren për përgatitjen e dokumenteve dhe analizën e informacionit, ndërsa në shkencë mund të ndihmojnë në përpunimin e sasive të mëdha të të dhënave dhe organizimin e kërkimeve. Falë përdorimit të tyre, shumë aktivitete mund të kryhen më shpejt dhe në mënyrë më efikase.

Inteligjenca artificiale gjeneruese mund të krijojë lloje të ndryshme të përmbajtjes digjitale. Përveç teksteve, sistemet moderne mund të ndihmojnë në krijimin e imazheve, muzikës, prezantimeve dhe materialeve të tjera multimediale. Këto teknologji gjejnë zbatim në arsim, dizajn, art, media dhe në shumë fusha të tjera. Megjithatë, përmbajtjet e krijuara duhet të analizohen me kujdes, ndërsa saktësia dhe përshtatshmëria e tyre duhet të verifikohen në përputhje me qëllimin konkret të përdorimit.

1.7.3 PËRDORIMI I PËRGJEGJSHËM I INTELIGJENCËS ARTIFICIALE GJENERUESE

Sistemet moderne të inteligjencës artificiale gjeneruese përfaqësojnë mjete të dobishme që mund t'i ndihmojnë njerëzit në kryerjen e aktiviteteve të ndryshme. Megjithatë, përdorimi i tyre kërkon përgjegjshmëri dhe mendim kritik. Informacionet e krijuara nga këto sisteme nuk duhet të pranohen automatikisht si plotësisht të sakta, por duhet të krahasohen me burime të besueshme dhe të verifikohen me kujdes. Kjo është veçanërisht e rëndësishme gjatë përdorimit të informacionit në arsim, shëndetësi dhe fusha të tjera me rëndësi të veçantë.

Gjatë përdorimit të inteligjencës artificiale gjeneruese, është e nevojshme të respektohen parimet etike dhe rregullat për përdorimin e sigurt të teknologjive digjitale. Duhet t'i kushtohet vëmendje mbrojtjes së të dhënave personale dhe privatësisë, si dhe respektimit të të drejtave të autorit dhe pronësisë intelektuale. Inteligjenca artificiale gjeneruese duhet të përdoret si një mjet ndihmës që kontribuon në të nxënit, kërkimin dhe krijimin e ideve të reja, dhe jo si zëvendësim për njohuritë, kreativitetin dhe përgjegjësinë njerëzore.

1.7.4 TEKNOLOGJIA BASHKËKOHORE NË PRAKTIKË

Inteligjenca artificiale gjeneruese në arsim

Një nxënës/e përgatit një prezantim për ndryshimet klimatike. Me ndihmën e inteligjencës artificiale gjeneruese, mund të marrë ide për strukturën e prezantimit, shpjegime të koncepteve dhe sugjerime për tema shtesë për kërkim. Më pas, ai/ajo i verifikon informacionet në tekste shkollore, enciklopedi dhe burime të tjera të besueshme dhe i plotëson ato me përfundimet e veta.

Në këtë mënyrë, inteligjenca artificiale gjeneruese përfaqëson një mjet që ndihmon në procesin e të nxënit, por nuk e zëvendëson të menduarit dhe kërkimin e pavarur. Përdorimi i përgjegjshëm i sistemeve të tilla kontribuon në zhvillimin e kompetencave digjitale, të mendimit kritik dhe të aftësisë për të analizuar burime të ndryshme informacioni.



A E DI SE?

A e di se inteligjenca artificiale gjeneruese mund të krijojë lloje të ndryshme të përmbajtjes, si tekste, imazhe, muzikë dhe video, por njeriu ka një rol të rëndësishëm në verifikimin dhe përdorimin e përgjegjshëm të këtyre përmbajtjeve?



HULUMTO

Hulumto tri fusha në të cilat zbatohet inteligjenca artificiale gjeneruese. Për secilën fushë, paraqit:

- çfarë lloj përmbajtjesh mund të krijojë;
- si u ndihmon njerëzve;
- pse është e rëndësishme të përdoret në mënyrë të përgjegjshme.





PYETJE PËR PËRSËRITJE

1. Çfarë janë modelet e mëdha gjuhësore?
2. Çfarë përfaqëson inteligjenca artificiale gjeneruese?
3. Çfarë llojesh përmbajtjesh mund të krijojë inteligjenca artificiale gjeneruese?
4. Në cilat fusha zbatohen modelet e mëdha gjuhësore?
5. Pse është e nevojshme të verifikohen në mënyrë kritike informacionet e marra nga inteligjenca artificiale gjeneruese?
6. Pse është e rëndësishme të respektohen të drejtat e autorit dhe privatësia gjatë përdorimit të këtyre teknologjive?
7. Si mund të ndihmojë inteligjenca artificiale gjeneruese në arsim?



DETYRA KËRKIMORE DHE PRAKTIKE

1. Hulumto tri përdorime të ndryshme të inteligjencës artificiale gjeneruese në jetën e përditshme dhe përgatit një raport të shkurtër.
2. Imagjino se duhet të përgatisësh një prezantim për mbrojtjen e mjedisit. Përkruaj se si mund të të ndihmojë inteligjenca artificiale gjeneruese gjatë përgatitjes dhe cilat informacione duhet t'i verifikosh patjetër në mënyrë të pavarur.
3. Përgatit një tabelë në të cilën do të krahasosh një motor kërkimi tradicional në internet me inteligjencën artificiale gjeneruese, sipas këtyre kriterëve:
 - mënyrës së funksionimit;
 - llojit të informacionit;
 - përdorimit;
 - nevojës për verifikimin e informacionit.
4. Mendo dhe paraqit pesë rregulla për përdorimin e sigurt dhe të përgjegjshëm të inteligjencës artificiale gjeneruese.
5. Shkruaj një ese të shkurtër me temën:
„Si mund të ndihmojë inteligjenca artificiale gjeneruese në të nxënit dhe në arsim?“



MBAJ MEND ÇFARË MËSOVE

Modelet e mëdha gjuhësore përfaqësojnë sisteme moderne të inteligjencës artificiale që përpunojnë dhe krijojnë tekst. Inteligjenca artificiale gjeneruese mund të krijojë lloje të ndryshme të përmbajtjes digjitale. Këto teknologji gjejnë zbatim në arsim, shkencë, art, biznes dhe në shumë fusha të tjera. Përmbajtjet e krijuara duhet të analizohen dhe të verifikohen në mënyrë kritike. Gjatë përdorimit të inteligjencës artificiale gjeneruese, duhet të respektohen të drejtat e autorit dhe privatësia. Inteligjenca artificiale gjeneruese përfaqëson një mjet ndihmës që e plotëson, por nuk e zëvendëson, njohurinë dhe përgjegjësinë njerëzore.





PËR ATA QË DËSHIROJNË TË DINË MË SHUMË

Zhvillimi i inteligjencës artificiale gjeneruese po përparon me ritme të shpejta dhe mundëson krijimin e përmbajtjeve digjitale gjithnjë e më të ndërlukuara. Përveç teksteve, sistemet moderne mund të krijojnë imazhe, muzikë, video dhe programe kompjuterike. Në të ardhmen pritet që përdorimi i tyre të zgjerohet edhe më shumë në arsim, shkencë, mjekësi dhe në veprimtari të ndryshme ekonomike. Njëkohësisht, po zhvillohen rregulla dhe rekomandime për përdorimin e tyre të sigurt, etik dhe të përgjegjshëm, me qëllim që teknologjia të kontribuojë në zhvillimin e shoqërisë dhe në përmirësimin e cilësisë së jetës.



KONTROLLO NJOHURITË E TUA

1. Rretho përgjigjen e saktë.

Modelet e mëdha gjuhësore përdoren për:

- a) prodhimin e kompjuterëve;
- b) përpunimin dhe krijimin e përmbajtjeve tekstuale;
- c) prodhimin e energjisë elektrike;
- ç) riparimin e telefonave celularë.

2. E saktë apo e pasaktë.

Inteligjenca artificiale gjeneruese mund të krijojë tekste, imazhe dhe përmbajtje të tjera digjitale.

3. Plotëso fjalinë.

Sistemet bashkëkohore që mund të krijojnë përmbajtje të reja digjitale quhen _____.

4. Shpjego termat.

Term	Shpjegim
Model i madh gjuhësor	_____
Inteligjencë artificiale gjeneruese (IAGj)	_____
Përdorim i përgjegjshëm	_____



DETYRË NGA SITUATA

Një nxënës përdor inteligjencën artificiale gjeneruese për përgatitjen e një projekti shkollor. Shpjegoni se cilët hapa duhet të ndërmarrë për të siguruar saktësinë e informacionit dhe përdorimin e përgjegjshëm të teknologjisë.

PROGRAMIMI NË C++

Programimi është procesi i krijimit të programeve kompjuterike që mundësojnë zgjidhjen e problemeve të ndryshme dhe automatizimin e detyrave. Me anë të gjuhëve programuese përcaktohen udhëzime sipas të cilave kompjuteri përpunon të dhëna, kryen llogaritje, merr vendime dhe realizon veprime të ndryshme. Sot, programimi gjen zbatim të gjerë në shumë fusha të shoqërisë bashkëkohore, si: arsimi, shkenca, industria, mjekësia, komunikimet dhe argëtimi, duke përfaqësuar një nga aftësitë themelore të epokës digjitale.

Në këtë tërësi tematike trajtohen parimet themelore të programimit përmes gjuhës programuese C++. Vëmendje e veçantë i kushtohet mënyrave të paraqitjes së të dhënave në kompjuter, hartimit të algoritmeve dhe programeve, përdorimit të funksioneve dhe aplikimit të vargjeve njëdimensionale gjatë zgjidhjes së detyrave programuese. Përmes shembujve dhe aktiviteteve praktike zhvillohet të menduarit algoritmik dhe fitohen njohuri dhe aftësi themelore për analizimin dhe krijimin e programeve kompjuterike.

KONCEPTE TË REJA

- sistemi binar i numërimit;
- sistemi dhjetor i numërimit;
- sistemi oktal i numërimit;
- sistemi heksadecimal i numërimit;
- funksion;
- parametër;
- bibliotekë;
- varg njëdimensional;
- indeks i elementit.

TASHMË DI ...

- të përdorësh kompjuterin dhe aplikacione të ndryshme;
- të futësh dhe të përpunosh të dhëna;
- të zbatosh të menduarit logjik gjatë zgjidhjes së problemeve;
- të dallosh lloje të ndryshme të informacionit;
- të ndjekësh dhe të zbatosh udhëzimet e dhëna;
- të analizosh situata të thjeshta nga jeta e përditshme;
- të përdorësh teknologjitë themelore të informacionit.



TEMA 2





2.1

PARAQITJA DIGJITALE E TË DHËNAVE NË KOMPJUTER

TASHMË DI ...

- të shpjegosh se çfarë përfaqëson një sistem kompjuterik;
- të japësh shembuj të pajisjeve digjitale;
- të dallosh harduerin nga softueri;
- të përdorësh llojet themelore të të dhënave në C++;
- të krijosh programe të thjeshta duke përdorur variabla dhe operatorë.

NË PËRMBAJTJET QË VIJOJNË DO TË MËSOSH:

- të shpjegosh se si kompjuteri i paraqet të dhënat;
- të dallosh bitin nga bajti;
- të shpjegosh se si paraqiten numrat, teksti, imazhet dhe tingulli;
- të dallosh llojet e ndryshme të të dhënave;
- të analizosh mënyrën se si përdoret memoria për ruajtjen e informacionit.

Në botën bashkëkohore digjitale, një numër i madh informacioni përpunohet, ruhet dhe transmetohet vazhdimisht përmes sistemeve të ndryshme kompjuterike. Tekstet që shkruhen, fotografitë që realizohen, muzika që dëgjohet dhe videot që shfaqen nuk ruhen në kompjuter në formën e tyre natyrore, por fillimisht shndërrohen në të dhëna digjitale. Për këtë arsye, është e nevojshme të kuptohet mënyra se si kompjuteri i paraqet informacionet dhe si ato ruhen në memorie.

Çdo e dhënë në kompjuter paraqitet përmes kombinimeve të vlerave binare. Edhe pse përdoruesit shohin shkronja, numra, imazhe dhe dëgjojnë tinguj, në prapavijë të gjitha informacionet paraqiten me zero dhe njësha. Kjo mënyrë e paraqitjes mundëson që të dhënat të përpunohen lehtësisht nga komponentët elektronikë të kompjuterit. Kuptimi i paraqitjes digjitale të të dhënave përbën bazën për studimin e mëtejshëm të programimit dhe informatikës.

Sistemet kompjuterike funksionojnë me sinjale elektronike që mund të kenë dy gjendje. Për këtë arsye, në kompjuterë përdoret mënyra binare e paraqitjes së të dhënave. Njësia themelore e informacionit quhet bit. Biti mund të ketë vetëm dy vlera – 0 ose 1. Edhe pse një bit i vetëm mbart një sasi shumë të vogël informacioni, me kombinimin e disa biteve mund të paraqiten të dhëna të ndërlikuara. Tetë bitë formojnë një bajt, i cili përfaqëson njësinë themelore për matjen e madhësisë së të dhënave. Në përdorimin e përditshëm përdoren shpesh edhe njësi më të mëdha matëse, si: kilobajt, megabajt, gigabajt dhe terabajt. Me zhvillimin e teknologjisë, sasia e të dhënave që përpunohen rritet vazhdimisht, prandaj pajisjet bashkëkohore disponojnë memorie në të cilat mund të ruhen sasi të mëdha informacioni. Të dhënat që ruhen në memorie duhet të jenë të organizuara në mënyrë që të mundësojnë leximin dhe përpunimin e tyre të shpejtë. Prandaj, paraqitja digjitale përbën parimin themelor mbi të cilin funksionon e gjithë teknologjia bashkëkohore e informacionit.

Shembulli 1: Shndërrimi i njësive matëse të të dhënave

```

1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      int megabajt;
6
7      cout << "Fut numrin e megabajtëve: ";
8      cin >> megabajt;
9
10     int kilobajt = megabajt * 1024;
11
12     cout << "Në kilobajt ka: " << kilobajt << endl;
13
14     return 0;
15 }
16

```

Përveç numrave, në kompjuter paraqiten edhe të dhënat tekstuale. Çdo shkronjë, shifër ose simbol merr një kod të caktuar binar përmes standardeve të veçanta të kodimit. Një nga standardet më të njohura është kodi ASCII, ndërsa sot përdoret gjerësisht edhe standardi Unicode, i cili mundëson paraqitjen e një numri të madh gjuhësh dhe simbolesh nga mbarë bota. Falë këtyre standardeve, teksti mund të shfaqet në mënyrë të saktë në pajisje dhe sisteme operative të ndryshme. Përveç tekstit, në formë digjitale paraqiten edhe imazhet. Çdo imazh digjital përbëhet nga një numër i madh pikash të vogla, të quajtura pikselë. Çdo piksel përmban informacion për ngjyrën, ndërsa kombinimi i mijëra ose miliona pikselëve krijon imazhin e plotë. Sa më i madh të jetë numri i pikselëve, aq më i lartë është cilësia e imazhit. Në mënyrë të ngjashme paraqitet edhe tingulli, ku valët zanore shndërrohen në vlera digjitale që kompjuteri mund t'i përpunojë. Në këtë mënyrë, lloje të ndryshme informacioni mund të ruhen, kopjohen dhe përpunohen në formë digjitale.

Shembulli 2: Paraqitja e vlerës ASCII të një shenje

```

1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      char shenja;
6
7      cout << "Fut një shenjë: ";
8      cin >> shenja;
9
10     cout << "Vlera ASCII është: " << (int)shenja << endl;
11
12     return 0;
13 }
14

```

Në gjuhët programuese, të dhënat organizohen përmes llojeve të ndryshme të të dhënave. Çdo lloj i të dhënave zë një hapësirë të caktuar në memorie dhe shërben për ruajtjen e një lloji të veçantë informacioni. Për shembull, tipi int përdoret për numra të plotë, double për numra realë, char për karaktere, ndërsa bool për vlera logjike. Zgjedhja e llojit të përshtatshëm të të dhënave është shumë e rëndësishme, sepse ndikon në saktësinë e të dhënave dhe në sasinë e memories që do të përdoret. Nëse për një vlerë të thjeshtë zgjidhet një tip që zë shumë hapësirë memorie, programi mund të shfrytëzojë burime në mënyrë të panevojshme. Nga ana tjetër, nëse zgjidhet një tip me interval të kufizuar vlerash, mund të shfaqen gabime gjatë llogaritjeve.

Për këtë arsye, programuesit i zgjedhin me kujdes llojet e të dhënave në përputhje me kërkesat e programit. Kuptimi i mënyrës se si organizohen të dhënat në memorie përbën një hap të rëndësishëm drejt krijimit të programeve më efikase.

Shembulli 3: Përdorimi i llojeve të ndryshme të të dhënave

```
*main.cpp X
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      int mosha = 16;
6      double mesatarja = 4.75;
7      char paralelja = 'A';
8      bool iRregullt = true;
9
10     cout << "Mosha: " << mosha << endl;
11     cout << "Mesatarja: " << mesatarja << endl;
12     cout << "Paralelja: " << paralelja << endl;
13     cout << "Nxënës i rregullt: " << iRregullt << endl;
14
15     return 0;
16 }
17
```

Memoria e kompjuterit paraqet hapësirën në të cilën ruhen programet dhe të dhënat. Çdo variabël që përdoret në një program merr vendin e vet në memorie. Kompjuteri i cakton automatikisht një adresë çdo të dhëne, ndërsa programi qaset te këto të dhëna përmes emrave të variablave.

Mënyra se si organizohet memoria ka rëndësi të madhe për shpejtësinë dhe efikasitetin e programeve. Kompjuterët bashkëkohorë disponojnë lloje të ndryshme memorieje, si memoria RAM dhe memoria e përhershme për ruajtjen e të dhënave. Memoria RAM shërben për ruajtjen e përkohshme të informacionit gjatë ekzekutimit të programit, ndërsa memoria e përhershme i ruan të dhënat edhe pas fikjes së kompjuterit.

Sasia e memories që përdor një program varet nga numri dhe lloji i të dhënave që përpunohen. Prandaj, kuptimi i paraqitjes digjitale dhe organizimit të të dhënave përbën një bazë të rëndësishme për studimin e mëtejshëm të algoritmeve, funksioneve dhe strukturave të të dhënave.

PYETJE DHE DETYRA PËR PËRSËRITJE



PYETJE

- | | |
|------------------------------|---|
| 1. Çfarë përfaqëson biti? | 3. Si paraqiten të dhënat tekstuale në kompjuter? |
| 2. Sa bitë përmban një bajt? | 4. Cili është roli i tipit double në C++? |



DETYRA

- | | |
|--|---|
| 1. Shkruaj një program që do të llogarisë sa bajtë përmban një numër i dhënë kilobajtësh. | 4. Shkruaj një program që do të llogarisë sa megabajt përmbajnë një numër i dhënë gigabajtësh. |
| 2. Shkruaj një program që do të shfaqë vlerën ASCII të një shenje të dhënë. | 5. Shkruaj një program që do të përcaktojë nëse shenja e futur është shkronjë e madhe apo shkronjë e vogël. |
| 3. Shkruaj një program që do të ruajë dhe do të shfaqë të dhënat për një nxënës/nxënëse duke përdorur lloje të ndryshme të të dhënave. | |



MBAJ MEND ÇFARË MËSOVE

- Kompjuterët i paraqesin të dhënat në formë binare.
- Njësia themelore e informacionit është biti.
- Tetë bitë formojnë një bajt.
- Teksti, imazhet dhe tingulli paraqiten me vlera binare.
- Llojet e ndryshme të të dhënave zënë hapësirë të ndryshme në memorie.
- Memoria mundëson ruajtjen dhe përpunimin e të dhënave.



PËR ATA QË DËSHIROJNË TË DINË MË SHUMË

Në sistemet moderne kompjuterike përdoren teknika të veçanta për kompresimin e të dhënave. Me kompresim zvogëlohet madhësia e skedarëve pa humbje të konsiderueshme të cilësisë. Falë kësaj, fotografitë, muzika dhe videot mund të transferohen më lehtë përmes internetit dhe të zënë më pak hapësirë në memorie. Formatet si JPG, MP3 dhe MP4 përdorin algoritme të ndryshme kompresimi me qëllim që të zvogëlohet sasia e të dhënave që duhet të ruhen.



2.2

SISTEMET E NUMËRIMIT DHE KONVERTIMET

TASHMË DI ...

- të shpjegosh se si kompjuteri i paraqet të dhënat;
- të dallosh bitin nga bajti;
- të japësh shembuj të llojeve të të dhënave;
- të shpjegosh se si paraqiten teksti, imazhet dhe tingulli;
- të përdorësh llojet themelore të të dhënave në C++.

DETYRA PËR PËRSËRITJE DHE PËRFORCIM

1. Sa bitë përmbajnë 4 bajtë?
2. Cili tip i të dhënave në C++ përdoret për ruajtjen e një karakteri?
3. Çfarë përfaqëson një piksel?
4. Shkruaj një program që do të llogarisë sa kilobajt përmbajnë një numër i dhënë megabajtësh.
5. Shpjego pse të dhënat në kompjuter paraqiten në formë digjitale.

NË PËRMBAJTJET QË VIJOJNË DO TË MËSOSH:

- të dallosh sistemin binar, dhjetor, oktal dhe heksadecimal të numërimit;
- të shpjegosh se çfarë përfaqëson baza e një sistemi të numërimit;
- të kryesh konvertime manuale ndërmjet sistemeve të numërimit;
- të analizosh vlerat pozicionale të shifrave;
- të zbatosh konvertimet në detyra programuese.

Në jetën e përditshme më së shpeshti përdoret sistemi dhjetor i numërimit, i cili përbëhet nga dhjetë shifra – nga 0 deri në 9. Megjithatë, kompjuterët nuk funksionojnë në të njëjtën mënyrë. Meqenëse qarqet elektronike mund të dallojnë në mënyrë të sigurt vetëm dy gjendje, në sistemet kompjuterike përdoret sistemi binar i numërimit. Për këtë arsye, të gjitha informacionet që përpunohen në kompjuter paraqiten përmes kombinimeve të zerove dhe njësheve.

Përveç sistemit binar, në informatikë përdoren shpesh edhe sistemi oktal dhe sistemi heksadecimal i numërimit. Këto sisteme mundësojnë paraqitjen më të lehtë të shënimeve të gjata binare dhe organizimin më të mirë të të dhënave. Kuptimi i sistemeve të numërimit përbën një pjesë të rëndësishme të informatikës, sepse përmes tyre kuptohet mënyra se si kompjuteri i përpunon dhe i ruan informacionet.

Sistemi i numërimit paraqet një mënyrë të paraqitjes së numrave duke përdorur një numër të caktuar simbolesh. Numri i simboleve që përdoren quhet bazë e sistemit të numërimit. Në sistemin dhjetor, baza është 10, sepse përdoren dhjetë shifra. Në sistemin binar, baza është 2, prandaj përdoren vetëm shifrat 0 dhe 1.

Çdo shifër në një numër ka vlerë pozicionale, e cila varet nga pozita e saj dhe nga baza e sistemit. Në sistemin dhjetor, pozitën paraqesin fuqitë e numrit 10, ndërsa në sistemin binar pozitën paraqesin fuqitë e numrit 2. Nëse numri binar ka 4 shifra, përdoren fuqitë e numrit 2 nga 0 deri në 3. Nëse numri binar ka 15 shifra, përdoren fuqitë e numrit 2 nga 0 deri në 14. Mblidhen vetëm vlerat e pozicioneve ku në numrin binar gjendet shifra 1. Për shembull, numri binar 1011 llogaritet në këtë mënyrë:

Shifër binare	1	0	1	1
Fuqi e numrit 2	2^3	2^2	2^1	2^0
Vlerë	8	0	2	1

Me mbledhjen e vlerave përftohet:

$$8 + 0 + 2 + 1 = 11_{10}$$

Në këtë mënyrë kryhet konvertimi nga sistemi binar në sistemin dhjetor. Sistemi binar përbën bazën e funksionimit të kompjuterëve, sepse komponentët elektronikë mund të paraqesin lehtësisht dy gjendje – të ndezur dhe të fikur. Për këtë arsye, të gjitha të dhënat dhe instruksionet në kompjuter paraqiten në formë binare.

Shembull 1: Shndërrimi i një numri binar në numër dhjetor

```
main.cpp x
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      int numerBinar;
7      int numerDhjetor = 0;
8      int baza = 1;
9
10     int mbetja;
11     cout << "Fut numrin binar: ";
12
13     cin >> numerBinar;
14     while (numerBinar > 0) {
15         mbetja = numerBinar % 10;
16         numerDhjetor = numerDhjetor + mbetja * baza;
17         baza = baza * 2;
18         numerBinar = numerBinar / 10;
19     }
20
21     cout << "Numri dhjetor: " << numerDhjetor << endl;
22     return 0;
23 }
24
```

```
Fut numrin binar: 1011
Numri dhjetor: 11
Process returned 0 (0x0)
Press any key to continue.
```

Konvertimi nga sistemi dhjetor në sistemin binar bëhet duke e pjesëtuar në mënyrë të njëpasnjëshme numrin me 2. Në çdo pjesëtim shënohet mbetja, ndërsa numri binar fitohet kur mbetjet lexohen në rend të kundërt. Për shembull, nëse duhet të shndërrohet numri 13_{10} në sistemin binar, procedura kryhet në këtë mënyrë:

Pjesëtim	Herësi	Mbetje
13 : 2	6	1
6 : 2	3	0
3 : 2	1	1
1 : 2	0	1

Me leximin e mbetjeve nga poshtë-lart përfitohet:

$$13_{10} = 1101_2$$

Kjo procedurë përfaqëson një nga konvertimet më të rëndësishme në informatikë. Falë këtyre konvertimeve, programuesit e kuptojnë më mirë mënyrën se si kompjuteri i përpunon numrat në memorie. Në gjuhët programuese shpesh përdoren algoritme që i kryejnë automatikisht këto procedura.

Shembull 2: Shndërrimi i një numri dhjetor në numër binar

```

main.cpp x
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      int numerDhjetor;
7      int numerBinar[32];
8
9      int i = 0;
10     cout << "Fut numrin dhjetor: ";
11
12     cin >> numerDhjetor;
13     while (numerDhjetor > 0) {
14         numerBinar[i] = numerDhjetor % 2;
15         numerDhjetor = numerDhjetor / 2;
16         i++;
17     }
18
19     cout << "Numri binar: ";
20     for (int j = i - 1; j >= 0; j--) {
21         cout << numerBinar[j];
22     }
23
24     cout << endl;
25     return 0;
26 }

```

Fut numrin dhjetor: 13
 Numri binar: 1101
 Process returned 0 (0x0)
 Press any key to continue.

Logs & others

Përveç sistemit binar, në informatikë përdoren edhe sistemi oktal dhe sistemi heksadecimal i numërimit. Sistemi oktal ka bazë 8 dhe përdor shifrat nga 0 deri në 7, ndërsa sistemi heksadecimal ka bazë 16 dhe përdor shifrat nga 0 deri në 9 dhe shkronjat A, B, C, D, E dhe F. Këto sisteme mundësojnë që numrat binarë të gjatë të paraqiten në mënyrë më të shkurtër dhe më të qartë. Gjatë konvertimit nga sistemi binar në sistemin oktal, shifrat binare grupohen nga tri shifra, ndërsa gjatë konvertimit në sistemin heksadecimal grupohen nga katër shifra. Grupimi bëhet nga e djathta në të majtë. Nëse në anën e majtë mbeten një ose dy shifra, ato plotësohen me zero nga e majta.

Shembull për konvertim në sistemin oktal:

10110110_2

Grupimi në grupe me nga 3 shifra:

$010\ 110\ 110$

$010_2 = 2_8\ 110_2 = 6_8\ 110_2 = 6_8$

Do të thotë:

$10110110_2 = 266_8$

Shembull për konvertim në sistemin heksadecimal:

10110110_2

Grupimi në grupe me nga 4 shifra:

$1011\ 0110$

$1011_2 = B_{16}\ 0110_2 = 6_{16}$

Domethënë:

$10110110_2 = B6_{16}$

Sistemi heksadecimal ka zbatim të gjerë në programim, në adresat e memories dhe në paraqitjen e ngjyrave në dizajnin e faqeve të internetit. Për shembull, ngjyra #FF0000 paraqet ngjyrën e kuqe në sistemin RGB.

Shembull 3: Paraqitja e shënimit oktal dhe heksadecimal

```
main.cpp x
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6
7      int numri;
8      cout << "Fut numrin dhjetor: ";
9
10     cin >> numri;
11
12     cout << "Paraqitja oktale: " << oct << numri << endl;
13
14     cout << "Paraqitja heksadecimale: " << hex << numri;
15     return 0;
16 }
```

Sistemet e ndryshme të numërimit përbëjnë një pjesë të rëndësishme të informatikës dhe arkitekturës kompjuterike. Edhe pse kompjuterët modernë i kryejnë automatikisht konvertimet, programuesit duhet të kuptojnë se si funksionojnë ato.

Kjo njohuri mundëson kuptimin më të lehtë të organizimit të memories, mënyrës së funksionimit të procesorit dhe përpunimit të të dhënave. Përveç kësaj, sistemet e numërimit përdoren edhe në rrjetet kompjuterike, kriptografi, elektronikë digjitale dhe zhvillimin e softuerit. Për këtë arsye, studimi i tyre përbën bazë për mësimin e mëtejshëm të programimit dhe të teknologjive moderne informatike.

PYETJE DHE DETYRA PËR PËRSËRITJE



PYETJE

1. Çfarë përfaqëson baza e një sistemi të numërimit?
2. Cilat shifra përdoren në sistemin binar?
3. Si bëhet konvertimi nga sistemi binar në sistemin dhjetor?
4. Sa është baza e sistemit heksadecimal?
5. Pse përdoret shënimi heksadecimal në informatikë?



DETYRA

1. Shndërro me dorë numrin 10101_2 në sistemin dhjetor.
2. Shndërro me dorë numrin 25_{10} në sistemin binar.
3. Shndërro me dorë numrin 11101010_2 në sistemin oktal dhe në sistemin heksadecimal.
4. Shkruaj një program që shndërron një numër binar në numër dhjetor.
5. Shkruaj një program që paraqet numrin e futur në sistemin oktal dhe heksadecimal.



MBAJ MEND ÇFARË MËSOVE

- Sistemi i numërimit paraqet një mënyrë për të shkruar numrat.
- Sistemi binar ka bazën 2.
- Sistemi oktal ka bazën 8.
- Sistemi heksadecimal ka bazën 16.
- Konvertimi nga sistemi binar në sistemin dhjetor bëhet përmes vlerave pozicionale.
- Konvertimi nga sistemi dhjetor në sistemin binar bëhet me pjesëtime të njëpasnjëshme me 2.



PËR ATA QË DËSHIROJNË TË DINË MË SHUMË

Në procesorët e kompjuterëve dhe në sistemet e memories shpesh përdoren vlera heksadecimale, sepse ato mundësojnë paraqitje më të shkurtër dhe më të qartë të informacionit binar. Në gjuhët programuese, adresat e memories shpesh paraqiten në formë heksadecimale. Gjithashtu, sistemi heksadecimal përdoret edhe për paraqitjen e ngjyrave në teknologjitë e uebit, ku çdo ngjyrë paraqitet me një kombinim prej gjashtë shenjash heksadecimale.



2.3

PARAQITJA E NUMRAVE TË PLOTË DHE GAMA E TË DHËNAVE

TASHMË DI ...

- të dallosh sistemin binar, dhjetor, oktal dhe heksadecimal të numërimit;
- të kryesh konvertime ndërmjet sistemeve të numërimit;
- të shpjegosh se çfarë përfaqëson baza e një sistemi të numërimit;
- të analizosh vlerat pozicionale;
- të përdorësh operatorët themelorë në C++.

DETYRA PËR PËRSËRITJE

1. Sa është baza e sistemit oktal të numërimit?
2. Shndërro me dorë numrin 1111_2 në sistemin dhjetor.
3. Shndërro me dorë numrin 45_{10} në sistemin binar.
4. Cili është shënimi heksadecimal i numrit binar 10101111_2 ?
5. Shkruaj një program që paraqet shënimin oktal të një numri të futur.

NË PËRMBAJTJEN QË VIJON DO TË MËSOSH:

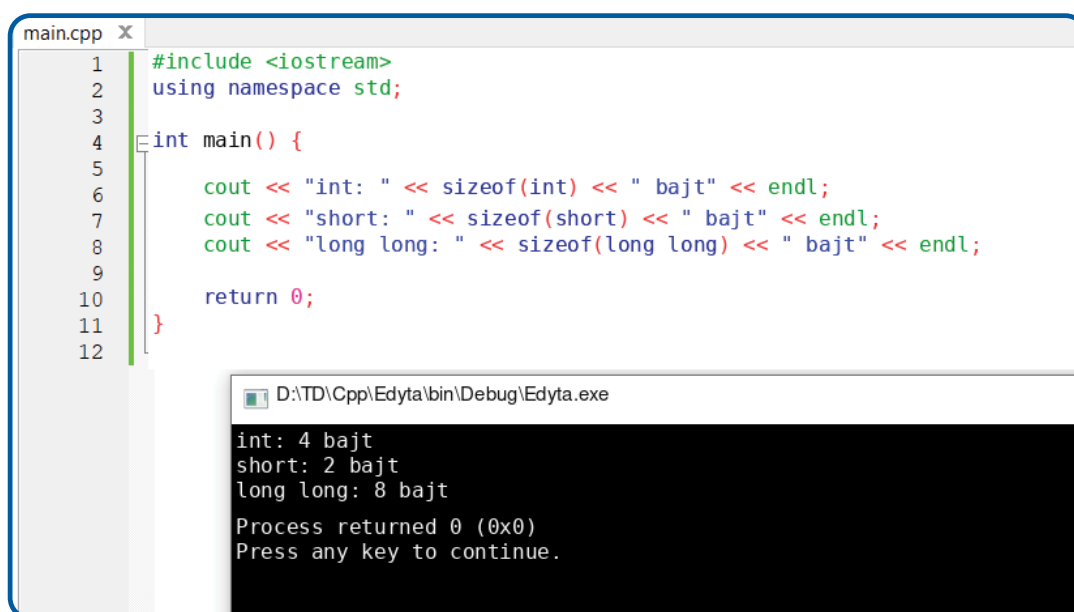
- të shpjegosh se si paraqiten numrat e plotë në kompjuter;
- të dallosh numrat pozitivë dhe negativë në memorie;
- të shpjegosh se çfarë përfaqëson diapazoni i të dhënave;
- të shpjegosh dukurinë e tejmbushjes (overflow);
- të përdorësh llojet e ndryshme të numrave të plotë në C++.

Në komunikimin e përditshëm numrat përdoren më së shpeshti pa menduar për mënyrën se si ruhen në kompjuter. Megjithatë, sistemet kompjuterike kanë memorie të kufizuar, prandaj çdo numër duhet të paraqitet në një format të caktuar dhe me një numër të kufizuar bitësh. Mënyra se si paraqiten numrat në memorie ka ndikim të madh në saktësinë e llogaritjeve, në shpejtësinë e përpunimit dhe në mundësinë e ruajtjes së informacioneve.

Në gjuhët programuese ekzistojnë lloje të ndryshme të të dhënave për paraqitjen e numrave të plotë. Çdo lloj zë hapësirë të ndryshme në memorie dhe mundëson paraqitjen e një diapazoni të ndryshëm vlerash. Nëse programi përpiqet të ruajë një numër që është më i madh se diapazoni i lejuar, mund të shfaqet një gabim i quajtur overflow. Për këtë arsye, programuesit duhet të zgjedhin me kujdes se cilat lloje të të dhënave do të përdorin në programe.

Kompjuterët i paraqesin të gjithë numrat në formë binare. Meqenëse memoria është e kufizuar, për çdo numër rezervohet një numër i caktuar bitësh. Për shembull, nëse për një numër përdoren 8 bitë, atëherë mund të paraqiten një numër i kufizuar vlerash të ndryshme. Te numrat pozitivë, çdo bit merr pjesë në formimin e numrit përmes fuqive të numrit 2. Për shembull, numri 13 në formë binare paraqitet si 00001101₂. Në memorie, çdo bit ka vendin dhe rëndësinë e tij të përcaktuar saktësisht. Sa më shumë bitë të përdoren, aq më i madh është diapazoni i numrave që mund të paraqitet. Në gjuhët programuese më së shpeshti përdoren lloje si: short, int, long long dhe të tjera, ku secili lloj përdor sasi të ndryshme të memories. Prandaj, zgjedhja e llojit të të dhënave ka ndikim të drejtpërdrejtë në sasinë e memories që përdor programi. Në sistemet moderne kompjuterike, zgjedhja e duhur e llojeve të të dhënave përbën një pjesë të rëndësishme të optimizimit të programeve.

Shembull 1: Paraqitja e madhësisë së llojeve të të dhënave



```

main.cpp x
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5
6      cout << "int: " << sizeof(int) << " bajt" << endl;
7      cout << "short: " << sizeof(short) << " bajt" << endl;
8      cout << "long long: " << sizeof(long long) << " bajt" << endl;
9
10     return 0;
11 }
12

```

```

D:\TD\Cpp\Edyta\bin\Debug\Edyta.exe
int: 4 bajt
short: 2 bajt
long long: 8 bajt
Process returned 0 (0x0)
Press any key to continue.

```

Përveç numrave pozitivë, në kompjuterë duhet të paraqiten edhe numrat negativë. Për këtë qëllim përdoret një mënyrë e veçantë e paraqitjes e quajtur two's complement. Në këtë mënyrë të paraqitjes, biti më i majtë zakonisht tregon shenjën e numrit. Falë këtij sistemi, procesori mund të kryejë më lehtë mbledhjen dhe zbritjen, si me numra pozitivë, ashtu edhe me numra negativë. Për shembull, nëse për paraqitjen e numrave përdoren 8 bitë, atëherë mund të paraqiten vlerat nga -128 deri në 127. Kjo do të thotë se ekziston një kufizim për numrin më të madh dhe numrin më të vogël që mund të ruhet në memorie. Nëse programi përpiqet të ruajë një numër jashtë diapazonit të lejuar, rezultati mund të jetë i pasaktë. Këto kufizime shpesh nuk vërehen në programet e vogla, por në sistemet e mëdha mund të shkaktojnë probleme serioze. Për këtë arsye, kuptimi i diapazonit të të dhënave përbën një pjesë të rëndësishme të programimit.

Shembull 2: Paraqitja e vlerës minimale dhe maksimale

```
main.cpp x
1  #include <iostream>
2  #include <climits>
3  using namespace std;
4
5  int main()
6  {
7      cout << "Vlera minimale e int: " << INT_MIN << endl;
8
9      cout << "Vlera maksimale e int: " << INT_MAX << endl;
10     return 0;
11 }
12
```

```
Vlera minimale e int: -2147483648
Vlera maksimale e int: 2147483647
Process returned 0 (0x0)
```

Diapazoni i të dhënave paraqet intervalin e vlerave që mund të paraqiten me një lloj të caktuar të të dhënave. Sa më i madh të jetë numri i bitëve që përdoren, aq më i madh është edhe diapazoni. Për shembull, lloji short zë më pak memorie se lloji long long, por mund të ruajë numra më të vegjël. Në programe shpesh duhet të vlerësohet me kujdes se çfarë diapazoni do të nevojitet. Nëse përdoret një lloj me diapazon të vogël për llogaritje me numra të mëdhenj, mund të ndodhë overflow. Overflow paraqet një situatë në të cilën rezultati e tejkalon vlerën maksimale që mund të ruhet. Në raste të tilla, programi mund të shfaqë rezultat të papritur ose të pasaktë. Gabime të tilla janë veçanërisht të rrezikshme në sistemet financiare, shkencore dhe të sigurisë. Për këtë arsye, programuesit duhet t'i kuptojnë kufizimet e llojeve të të dhënave dhe t'i zgjedhin ato në mënyrë të duhur, sipas nevojave të programit.

Shembull 3: Overflow te numrat e plotë

```
main.cpp x
1  #include <iostream>
2  #include <climits>
3  using namespace std;
4
5  int main()
6  {
7
8      int numri = INT_MAX;
9
10     cout << "Para rritjes: " << numri << endl;
11
12     numri++;
13
14     cout << "Pas rritjes: " << numri << endl;
15     return 0;
16 }
17
```

```
D:\TD\Cpp\Edyta\bin\Debug\Edyta.exe
Para rritjes: 2147483647
Pas rritjes: -2147483648
Process returned 0 (0x0)
Press any key to continue.
```

Gjuhët moderne të programimit mundësojnë përdorimin e llojeve të ndryshme të të dhënave, me qëllim që programet të jenë më efikase dhe më të sakta. Në disa situata është e nevojshme të ruhen numra

shumë të mëdhenj, ndërsa në situata të tjera është e rëndësishme të kursehet memoria. Për këtë arsye, programuesit zgjedhin lloje të ndryshme të të dhënave sipas problemit konkret. Përveç kësaj, kuptimi i saktë i diapazonit të vlerave të numrave mundëson zbulimin më të lehtë të gabimeve dhe analizimin më të mirë të programeve. Njohja e mënyrës se si paraqiten numrat në memorie përbën bazë për mësimin e mëtejshëm të algoritmeve, strukturave të të dhënave dhe optimizimit të programeve.

PYETJE DHE DETYRA PËR PËRSËRITJE



PYETJE

1. Pse kompjuterët përdorin paraqitjen binare të numrave?
2. Çfarë përfaqëson diapazoni i të dhënave?
3. Çfarë është overflow?
4. Cili lloj i të dhënave mund të ruajë vlera më të mëdha – int apo long long?
5. Pse është e rëndësishme të zgjidhet saktë lloji i të dhënave?



DETYRA

1. Shkruaj një program që do ta shfaqë madhësinë e llojeve char, int dhe double.
2. Shkruaj një program që do t'i shfaqë vlerën minimale dhe maksimale të llojit int.
3. Shkruaj një program që do të demonstrojë dukurinë overflow te lloji short.
4. Shkruaj një program që do të marrë dy numra të mëdhenj dhe do t'i mbledhë duke përdorur llojin long long.
5. Hulumto se çfarë diapazoni vlerash ka lloji unsigned int dhe krahasojë atë me int.



MBAJ MEND ÇFARË MËSOVE

- Kompjuterët i paraqesin numrat në formë binare.
- Çdo lloj i të dhënave përdor një numër të ndryshëm bajtësh.
- Diapazoni përfaqëson vlerën më të vogël dhe më të madhe që mund të ruhet.
- Overflow shfaqet kur vlera e tejkalon diapazonin e lejuar.
- Zgjedhja e duhur e llojit të të dhënave është e rëndësishme për saktësinë dhe efikasitetin e programit.



PËR ATA QË DËSHIROJNË TË DINË MË SHUMË

Në gjuhët moderne të programimit ekzistojnë edhe lloje të të dhënave me diapazon shumë të madh, të cilat përdoren në simulimet shkencore, kriptografi dhe përpunimin e bazave të mëdha të të dhënave. Disa biblioteka programuese mundësojnë punën me numra që përmbajnë qindra ose mijëra shifra. Numra të tillë nuk mund të paraqiten drejtpërdrejt me llojet standarde të të dhënave, prandaj për përpunimin e tyre përdoren algoritme dhe struktura të veçanta.



2.4

OPTIMIZIMI DHE EFIKASITETI I ALGORITMEVE

TASHMË DI ...

- të shpjegosh se çfarë përfaqëson një algoritëm;
- të përdorësh strukturat if dhe switch;
- të përdorësh ciklet for dhe while;
- të analizosh kompleksitetin kohor;
- të zgjidhësh detyra me tekst në C++.

DETYRA PËR PËRSËRITJE

1. Çfarë përfaqëson një algoritëm?
2. Shkruaj një shembull të një cikli for që përsëritet 10 herë.
3. Cili është ndryshimi midis cikleve for dhe while?
4. Çfarë përfaqëson kompleksiteti kohor?
5. Shkruaj një program që do të llogarisë katrorin e një numri të dhënë.

NË PËRMBAJTJET QË VIJOJNË DO TË MËSOSH TË:

- të analizosh efikasitetin e algoritmeve;
- të krahasosh zgjidhje të ndryshme për të njëjtin problem;
- të identifikosh veprimet e panevojshme;
- të optimizosh algoritmet;
- të analizosh ndikimin e madhësisë së të dhënave në shpejtësinë e programit.

Në programim, shumë shpesh ekzistojnë mënyra të ndryshme për zgjidhjen e të njëjtit problem. Edhe pse zgjidhjet e ndryshme mund të japin të njëjtin rezultat, ato nuk funksionojnë gjithmonë me të njëjtën shpejtësi dhe nuk përdorin të njëjtën sasi memorie. Në programet e vogla, këto dallime shpesh nuk vërehen, por gjatë përpunimit të sasive të mëdha të të dhënave ato mund të ndikojnë ndjeshëm në funksionimin e sistemit.

Për këtë arsye, programimi modern nuk fokusohet vetëm në saktësinë e programit, por edhe në efikasitetin e tij. Programuesit analizojnë vazhdimisht se sa operacione kryen algoritmi, sa memorie përdor dhe nëse ekziston një mënyrë më e mirë për zgjidhjen e problemit. Ky proces quhet optimizim i algoritmeve dhe përbën një pjesë të rëndësishme të zhvillimit të softuerëve modernë.

Efikasiteti i një algoritmi zakonisht analizohet përmes numrit të operacioneve që kryhen gjatë zgjidhjes së problemit. Sa më shumë operacione të nevojiten, aq më shumë kohë do të kërkohej për ekzekutimin e programit. Për shembull, nëse duhet të llogaritet shuma e numrave nga 1 deri në n , mund të përdoret një cikël që i mbledh të gjithë numrat një nga një. Kjo qasje është e saktë, por për vlera shumë të mëdha të n do të kryhet një numër i madh operacionesh. Nga ana tjetër, i njëjti problem mund të zgjidhet me një formulë matematikore, e cila jep rezultatin menjëherë. Në këtë rast, algoritmi kryen shumë më pak operacione dhe funksionon më shpejt. Për këtë arsye, programuesit vazhdimisht kërkojnë mënyra për të reduktuar llogaritjet e panevojshme. Një mënyrë e tillë e të menduarit përbën bazën e optimizimit të algoritmeve dhe është një nga pjesët më të rëndësishme të programimit modern.

Shembull 1: Krahësimi i dy algoritmeve

```

main.cpp x
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5
6      int n;
7      cin >> n;
8
9      int shuma1 = 0;
10
11     for (int i = 1; i <= n; i++) {
12         shuma1 += i;
13     }
14
15     int shuma2 = n * (n + 1) / 2;
16
17     cout << "Me cikël: " << shuma1 << endl;
18     cout << "Me formulë: " << shuma2 << endl;
19
20     return 0;
21 }
22

```

```

D:\TD\Cpp\Edyta\bin\Debug\Edyta.exe
500
Me cikël: 125250
Me formulë: 125250

Process returned 0 (0x0)   execution time : 3.100 s
Press any key to continue.

```

Gjatë analizës së një algoritmi, është shumë e rëndësishme të shqyrtohet se si ndryshon shpejtësia e programit kur rritet sasia e të dhënave. Një algoritëm që funksionon shpejt për 10 elemente mund të bëhet shumë i ngadaltë kur përpunon një milion elemente. Për këtë arsye, programuesit nuk analizojnë vetëm shpejtësinë momentale, por edhe mënyrën se si sillet algoritmi me rritjen e madhësisë së të dhënave hyrëse. Në informatikë përdoren shpesh shënime si $O(n)$, $O(n^2)$ dhe $O(\log n)$ për të përshkruar kompleksitetin e algoritmeve. Një algoritëm me kompleksitet $O(n^2)$ zakonisht bëhet shumë më i ngadaltë me rritjen e sasisë së të dhënave sesa një algoritëm me kompleksitet $O(n)$. Për shembull, ciklet e ndërthurura shpesh shkaktojnë rritje të konsiderueshme të numrit të operacioneve. Për këtë arsye, programuesit analizojnë me kujdes nëse mund të shmanget përsëritja e panevojshme ose nëse ekziston një zgjidhje më e mirë. Në sistemet e mëdha kompjuterike, optimizimi i duhur mund të përmirësojë ndjeshëm performancën e programit.

Shembull 2: Kërkim joefikas dhe më efikas

```
*main.cpp X
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      int vargu[5] = {2, 4, 6, 8, 10};
7      int numri;
8      bool uGjet = false;
9      cin >> numri;
10     for (int i = 0; i < 5; i++) {
11
12         if (vargu[i] == numri) {
13             uGjet = true;
14             break;
15         }
16     }
17     if (uGjet) {
18         cout << "Numri u gjet." << endl;
19     } else {
20         cout << "Numri nuk u gjet."
21             << endl;
22     }
23     return 0;
24 }
25
```

Në shembullin e mëparshëm përdoret komanda break për ndërprerjen e ciklit menjëherë pas gjetjes së numrit. Pa break, cikli do të vazhdonte të kontrollonte edhe elemente të panevojshme. Edhe pse dallimi është i vogël te një varg me pesë elemente, te vargjet e mëdha kjo mund të ndikojë ndjeshëm në shpejtësinë e programit. Përmirësime të tilla të vogla shpesh kanë rëndësi të madhe në programimin profesional. Programuesit vazhdimisht analizojnë mënyrat për të zvogëluar numrin e operacioneve dhe për ta përdorur

memorien në mënyrë më efikase. Ky proces nuk nënkupton vetëm një „program më të shpejtë“, por edhe organizim më të mirë të kodit dhe mirëmbajtje më të lehtë të sistemit.

Shembull 3: Ndikimi i cikleve të ndërthurura

The screenshot shows a C++ IDE with a file named `main.cpp`. The code is as follows:

```

1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      int n;
7      cin >> n;
8      for (int i = 1; i <= n; i++) {
9          for (int j = 1;
10             j <= n; j++) {
11              cout << "* ";
12          }
13          cout << endl;
14      }
15      return 0;
16  }
```

The program is executed, and the output window shows the following:

```

5
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *

Process returned 0 (0x0)
Press any key to continue.
```

Në shembullin e mëparshëm përdoren cikle të ndërthurura. Nëse n ka vlerën 5, do të kryhen 25 përsëritje. Nëse n ka vlerën 1000, do të kryhen një milion përsëritje. Kjo tregon se si shpejtësia e algoritmit mund të ndryshojë ndjeshëm kur rritet madhësia e të dhënave hyrëse. Për këtë arsye, programuesit duhet të mendojnë me kujdes kur përdorin cikle të ndërthurura dhe të analizojnë nëse ekziston një zgjidhje më efikase. Në sistemet moderne që përpunojnë sasi të mëdha të të dhënave, optimizimi përbën një nga faktorët më të rëndësishëm për funksionimin e suksesshëm të programeve.

PYETJE DHE DETYRA PËR PËRSËRITJE



PYETJE

1. Çfarë përfaqëson optimizimi i një algoritmi?
2. Pse dy algoritme mund të kenë shpejtësi të ndryshme?
3. Si ndikojnë ciklet e ndërthurura në kompleksitetin?
4. Pse është e rëndësishme të shmangen operacionet e panevojshme?
5. Si ndikon madhësia e të dhënave hyrëse në shpejtësinë e programit?



DETYRA

1. Shkruaj një program që do të llogarisë sa numra nga 1 deri në n janë të pjesëtueshëm me 7.
2. Shkruaj një program që do të lexojë n numra dhe do të gjejë numrin më të madh.
3. Shkruaj një program me cikle të ndërthurura që do të shtypë një drejtkëndësh me simbolin #.
4. Analizo sa kontrole do të kryhen nëse programi kontrollon çdo element të një vargu me 1000 elemente.
5. Shkruaj dy zgjidhje të ndryshme për llogaritjen e shumës nga 1 deri në n dhe krahaso se cila zgjidhje është më efektive.



MBAJ MEND ÇFARË MËSOVE

- Algoritme të ndryshme mund të zgjidhin të njëjtin problem me efikasitet të ndryshëm.
- Optimizimi synon të zvogëlojë numrin e operacioneve dhe përdorimin e memories.
- Ciklet e ndërthurura e rrisin ndjeshëm kompleksitetin.
- Komanda break mund të ndihmojë në zvogëlimin e kontroleve të panevojshme.
- Analiza e algoritmeve është e rëndësishme gjatë punës me sasi të mëdha të të dhënave.



PËR ATA QË DËSHIROJNË TË DINË MË SHUMË

Në shërbimet moderne të internetit dhe sistemet e inteligjencës artificiale, përpunohen miliona të dhëna çdo sekondë. Nëse algoritmet nuk janë mjaftueshëm të optimizuara, serverët do të punonin shumë më ngadalë dhe do të shpenzonin një sasi të madhe burimesh. Për këtë arsye, një pjesë e madhe e punës së programuesve profesionistë konsiston në analizimin dhe përmirësimin e algoritmeve. Në disa raste, optimizimi i një algoritmi mund ta përshpejtojë programin qindra ose edhe mijëra herë.

2.5



KOMBINIMI I KUSHTEVE DHE CIKLEVE

TI TASHMË DI...

- të përdorësh struktura if dhe switch;
- të përdorësh ciklet for dhe while;
- të analizosh efikasitetin e algoritmeve;
- të përdorësh cikle të ndërthurura;
- të krahasosh zgjidhje të ndryshme programore.

DETYRA PËR PËRSËRITJE

1. Çfarë përfaqëson optimizimi i një algoritmi?
2. Pse break mund ta përmirësojë efikasitetin?
3. Çfarë ndodh kur përdoren dy cikle të ndërthurura?
4. Shkruaj një program që do të llogarisë shumën nga 1 deri në n.
5. Shpjego pse të dhënat e mëdha kërkojnë algoritme më efikase.

NË PËRMBAJTJET QË VIJOJNË DO TË MËSOSH:

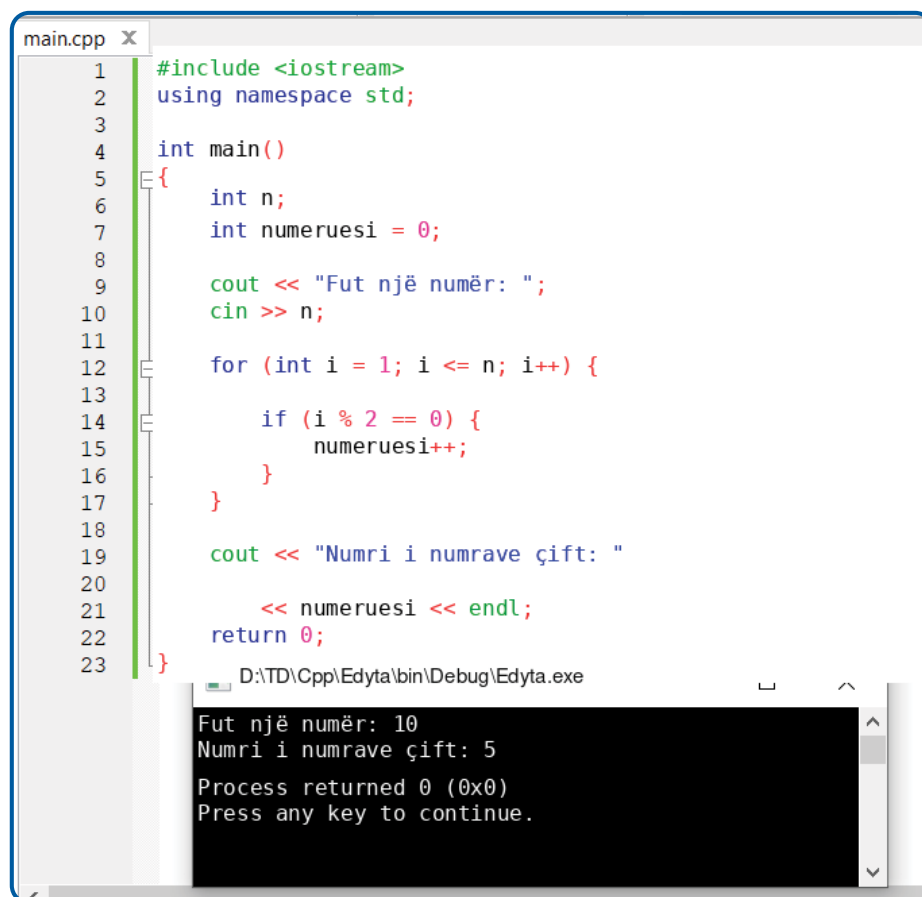
- të kombinosh kushtet dhe ciklet në të njëjtin program;
- të zgjedhësh detyra tekstuale më të ndërlikuara;
- të përdorësh kushte dhe cikle të ndërthurura;
- të analizosh logjikën e përsëritjes dhe të zgjedhjes;
- të ndërtosh zgjidhje programore më të organizuara.

Në programim, shumë probleme nuk mund të zgjidhen vetëm me kushte ose vetëm me cikle. Në programet reale, më së shpeshti është e nevojshme kombinimi i strukturave të ndryshme programore, në mënyrë që të arrihet një zgjidhje e plotë. Për shembull, programi mund të kërkojë futjen e të dhënave disa herë dhe, për çdo të dhënë të futur, të kontrollojë nëse plotësohet një kusht i caktuar. Situata të tilla kërkojnë përdorimin e njëkohshëm të cikleve dhe kushteve.

Kombinimi i strukturave programore përfaqëson një hap të rëndësishëm drejt zhvillimit të algoritmeve më të ndërlikuara. Me këtë qasje, programet bëhen më fleksibile dhe mund të përpunojnë situata të ndryshme. Për këtë arsye, programuesit duhet të analizojnë me kujdes se ku duhet të vendoset kushti, kur duhet të përsëritet procedura dhe si duhet organizuar saktë logjika e programit.

Ciklet mundësojnë përsëritjen e udhëzimeve të caktuara, ndërsa kushtet mundësojnë marrjen e vendimeve. Kur këto struktura kombinohen, programi mund të përsërisë veprime dhe njëkohësisht të kontrollojë kushte të ndryshme. Për shembull, nëse duhet të numërohen vetëm numrat çift nga 1 deri në n, cikli do t'i përshkojë të gjithë numrat, ndërsa kushti do të kontrollojë nëse secili numër është çift. Në këtë mënyrë, kushti ekzekutohet brenda ciklit. Kjo mënyrë pune përfaqëson një nga teknikat më të zakonshme në programim. Në programet bashkëkohore shpesh përpunohen një numër i madh të dhënash, ku për secilën të dhënë kryhen kontrole të ndryshme. Për këtë arsye, kombinimi i saktë i kushteve dhe cikleve përbën bazën për zgjidhjen e problemeve më të ndërlikuara dhe ndërtimin e algoritmeve efikase.

Shembull 1: Numërimi i numrave çift



```
main.cpp X
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      int n;
7      int numeruesi = 0;
8
9      cout << "Fut një numër: ";
10     cin >> n;
11
12     for (int i = 1; i <= n; i++) {
13
14         if (i % 2 == 0) {
15             numeruesi++;
16         }
17     }
18
19     cout << "Numri i numrave çift: "
20
21         << numeruesi << endl;
22     return 0;
23 }
```

D:\TD\Cpp\Edyta\bin\Debug\Edyta.exe

```
Fut një numër: 10
Numri i numrave çift: 5
Process returned 0 (0x0)
Press any key to continue.
```

Në shumë detyra është e nevojshme që kushtet të jenë të ndërthurura brenda cikleve ose ciklet të jenë të ndërthurura brenda kushteve. Ndërtime të tilla mundësojnë zgjidhjen e problemeve shumë më të ndërlikuara. Për shembull, gjatë analizimit të nxënësve në një paralele mund të jetë e nevojshme të futen nota derisa të futet një vlerë e caktuar, dhe më pas për secilën notë të kontrollohet nëse ajo është pozitive apo negative. Këto algoritme kërkojnë organizim të kujdesshëm të logjikës, sepse një kusht i vendosur gabimisht mund të shkaktojë një cikël të pafund ose një rezultat të pasaktë. Për këtë arsye, programuesit shpesh përdorin

një qasje hap pas hapi – fillimisht analizojnë përsëritjen dhe më pas i shtojnë kushtet. Kjo mënyrë pune mundëson testim më të lehtë dhe organizim më të mirë të kodit.

Shembull 2: Shuma e numrave pozitivë

```

main.cpp x
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      int numri;
7      int shuma = 0;
8
9      for (int i = 1; i <= 5; i++) {
10
11         cout << "Fut një numër: ";
12         cin >> numri;
13
14         if (numri > 0) {
15             shuma += numri;
16         }
17     }
18
19     cout << "Shuma e numrave pozitivë: "
20
21         << shuma << endl;
22     return 0;
23 }
D:\TD\Cpp\Edyta\bin\Debug\Edyta.exe
Fut një numër: 2
Fut një numër: -3
Fut një numër: 4
Fut një numër: -5
Fut një numër: 1
Shuma e numrave pozitivë: 7
Process returned 0 (0x0)
Press any key to continue.

```

Kombinimi i kushteve dhe cikleve përdoret shpesh edhe gjatë përpunimit të detyrave tekstuale. Në detyra të tilla është e nevojshme të analizohen situata të ndryshme dhe të zgjidhet procedura e duhur. Për shembull, programi mund të kontrollojë nëse numri është i thjeshtë, nëse është i pjesëtueshëm me një numër tjetër ose nëse plotëson disa kushte njëkohësisht.

Në situata të tilla, kushtet mund të jenë shumë të ndërlikuara dhe të përmbajnë disa operatorë logjikë. Përveç kësaj, ciklet shpesh përdoren për përsëritjen e kontrolleve mbi të dhëna të shumta. Për këtë arsye, programe të tilla përbëjnë një bazë të mirë për zhvillimin e të menduarit algoritmik dhe të analizës logjike. Sa më shumë të kombinohen struktura të ndryshme, aq më të fuqishme bëhen programet dhe aq më të afta janë për zgjidhjen e problemeve reale.

```

main.cpp x
1  #include <iostream>
2  using namespace std;
3  int main()
4  {
5      int numri;
6      bool iThjeshte = true;
7      cout << "Fut një numër: ";
8      cin >> numri;
9      if (numri <= 1) {
10         iThjeshte = false;
11     }
12     for (int i = 2;
13
14         i < numri; i++) {
15         if (numri % i == 0) {
16             iThjeshte = false;
17         }
18     }
19     if (iThjeshte) {
20
21         cout << "Numri është i thjeshtë."
22             << endl;
23     } else {
24
25         cout << "Numri nuk është i thjeshtë."
26             << endl;
27     }
28
29     return 0;
30 }
31

```

Fut një numër: 9
 Numri nuk është i thjeshtë.
 Process returned 0 (0x0)
 Press any key to continue.

Në shembullin e mëparshëm, cikli kontrollon në mënyrë të njëpasnjëshme pjesëtuesit e mundshëm të numrit, ndërsa kushti verifikon nëse ekziston pjesëtueshmëri pa mbetje. Kombinimi i këtyre strukturave mundëson analizimin e saktë të numrit dhe marrjen e rezultatit të duhur. Kjo qasje përdoret në një numër të madh algoritmesh ku kërkohet analizimi i të dhënave, kontrollimi i kushteve dhe përsëritja e veprimeve. Për këtë arsye, kombinimi i kushteve dhe cikleve përbën një nga hapat më të rëndësishëm në zhvillimin e zgjidhjeve programuese më të avancuara.

PYETJE DHE DETYRA PËR PËRSËRITJE



PYETJE

1. Pse programet kombinojnë kushtet dhe ciklet?
2. Çfarë përfaqëson një kusht i ndërthurur?
3. Çfarë roli ka kushti brenda një cikli?
4. Pse është e rëndësishme të organizohet siç duhet logjika e programit?
5. Si ndihmojnë ciklet në përpunimin e më shumë të dhënave?



DETYRA

1. Shkruaj një program që do të numërojë numrat çift nga 1 deri në n.
2. Shkruaj një program që do të pranojë 10 numra dhe do të llogarisë shumën vetëm të numrave negativë.
3. Shkruaj një program që do të kontrollojë nëse numri i dhënë është numër i përsosur.
4. Shkruaj një program që do t'i shfaqë të gjithë numrat e pjesëtueshëm me 3 dhe 5 në një interval të caktuar.
5. Shkruaj një program që do të kontrollojë sa nga numrat e dhënë janë pozitivë dhe sa janë negativë.



MBAJ MEND ÇFARË MËSOVE

- Ciklet dhe kushtet më së shpeshti përdoren së bashku.
- Kushtet mundësojnë kontrollimin e situatave të ndryshme.
- Ciklet mundësojnë përsëritjen e veprimeve.
- Kombinimi i strukturave mundëson zgjidhjen e problemeve më të ndërlikuara.
- Organizimi i duhur i logjikës është shumë i rëndësishëm në programim.



PËR ATA QË DËSHIROJNË TË DINË MË SHUMË

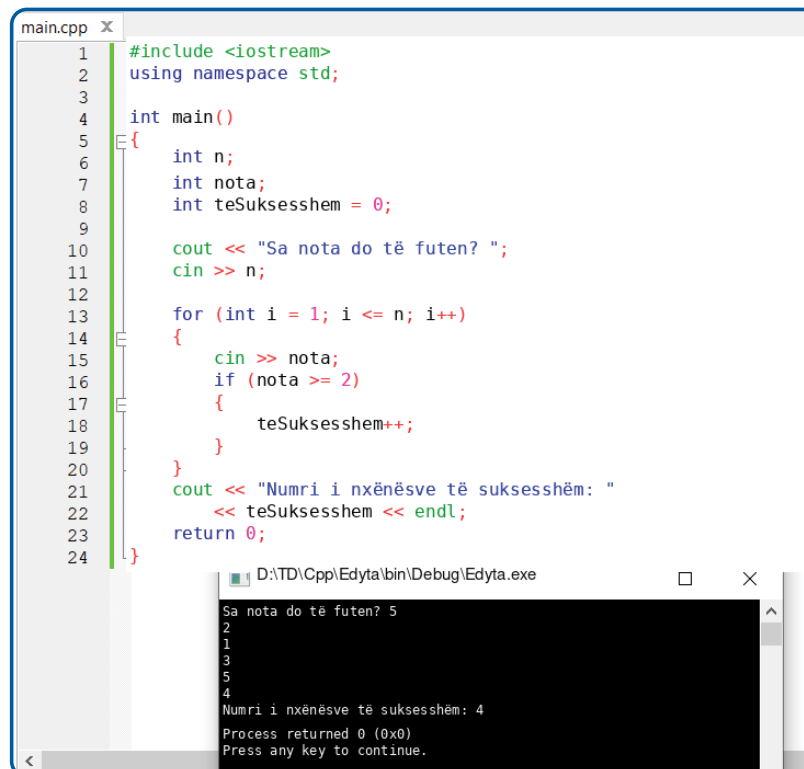
Në sistemet moderne kompjuterike, një numër i madh algoritmesh bazohen në kombinimin e kushteve dhe përsëritjeve. Motorët e kërkimit në internet, sistemet e rekomandimeve dhe inteligjenca artificiale përpunojnë vazhdimisht sasi të mëdha të të dhënave duke përdorur kushte dhe cikle të ndryshme. Në disa programe kryhen miliarda kontrolle në sekondë. Për këtë arsye, organizimi i saktë i kushteve dhe cikleve ka ndikim të madh në shpejtësinë dhe efikasitetin e sistemeve moderne.



Gjatë zgjidhjes së problemeve komplekse, është shumë e rëndësishme që detyra fillimisht të lexohet me kujdes dhe të veçohen informacionet kryesore. Duhet të përcaktohen të dhënat hyrëse, kushtet që duhet të kontrollohen dhe rezultati që duhet të arrihet. Më pas, problemi ndahet gradualisht në hapa më të vegjël, të cilët mund të realizohen më lehtë në program. Kjo qasje mundëson zbulimin më të lehtë të gabimeve dhe krijimin e zgjidhjeve më të organizuara.

Gjatë analizimit të një detyre komplekse tekstuale, hapi i parë është kuptimi i saktë i problemit. Në shumë raste, detyra përmban disa kushte që duhet të plotësohen njëkohësisht. Nëse programuesi nuk i analizon me kujdes të gjitha kërkesat, programi mund të japë rezultate të pasakta. Për këtë arsye, shpesh përdoret një qasje graduale, në të cilën problemi ndahet në disa nëndetyra. Për shembull, nëse duhet të analizohen rezultatet e testimit të nxënësve, fillimisht mund të numërohen nxënësit që kanë kaluar, më pas të llogaritet mesatarja dhe, në fund, të paraqiten të dhënat statistikore. Me një ndarje të tillë, algoritmi bëhet më i qartë dhe më i lehtë për t'u realizuar. Në programimin modern, kjo mënyrë e të menduarit përbën bazën për zgjidhjen e problemeve komplekse dhe zhvillimin e sistemeve më të mëdha softuerike.

Shembull 1: Analiza e notave



The screenshot shows a C++ program in a code editor window titled 'main.cpp'. The code is as follows:

```

1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      int n;
7      int nota;
8      int teSuksesshem = 0;
9
10     cout << "Sa nota do të futen? ";
11     cin >> n;
12
13     for (int i = 1; i <= n; i++)
14     {
15         cin >> nota;
16         if (nota >= 2)
17         {
18             teSuksesshem++;
19         }
20     }
21     cout << "Numri i nxënësve të suksesshëm: "
22          << teSuksesshem << endl;
23     return 0;
24 }
```

Below the code editor, the execution output is shown in a console window titled 'D:\TD\Cpp\Edyta\bin\Debug\Edyta.exe'. The output is:

```

Sa nota do të futen? 5
2
1
3
5
4
Numri i nxënësve të suksesshëm: 4
Process returned 0 (0x0)
Press any key to continue.
```

Detyrat komplekse shpesh kërkojnë kombinimin e disa teknikave programuese në të njëjtin program. Në një detyrë mund të përdoren njëkohësisht cikle, kushte, numërues dhe llogaritje të ndryshme. Për këtë arsye, është shumë e rëndësishme që programi të jetë i organizuar mirë dhe i ndarë në mënyrë logjike. Programuesit shpesh përdorin variabla shtesë për të ndjekur gjendje të ndryshme gjatë ekzekutimit të programit. Për shembull, gjatë analizimit të numrave mund të ndiqen njëkohësisht vlera më e madhe, vlera më e vogël dhe numri i numrave pozitivë. Detyra të tilla kërkojnë planifikim të kujdesshëm të algoritmit përpara fillimit të programimit. Nëse logjika nuk organizohet siç duhet, programi mund të bëhet i vështirë për t'u kuptuar dhe mirëmbajtur. Prandaj, ndërtimi gradual i zgjidhjes përbën një pjesë shumë të rëndësishme të programimit.



```

1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      int n;
7      int numri;
8      cout << "Sa numra do të futen? ";
9      cin >> n;
10     cin >> numri;
11     int maksimumi = numri;
12     int minimumi = numri;
13     for (int i = 2; i <= n; i++)
14     {
15         cin >> numri;
16         if (numri > maksimumi)
17         {
18             maksimumi = numri;
19         }
20         if (numri < minimumi)
21         {
22             minimumi = numri;
23         }
24     }
25     cout << "Maksimumi: "
26          << maksimumi << endl;
27     cout << "Minimumi: " << minimumi << endl;
28     return 0;
29 }

```

D:\TD\Cpp\Edyta\bin\Debug\Edyta.exe

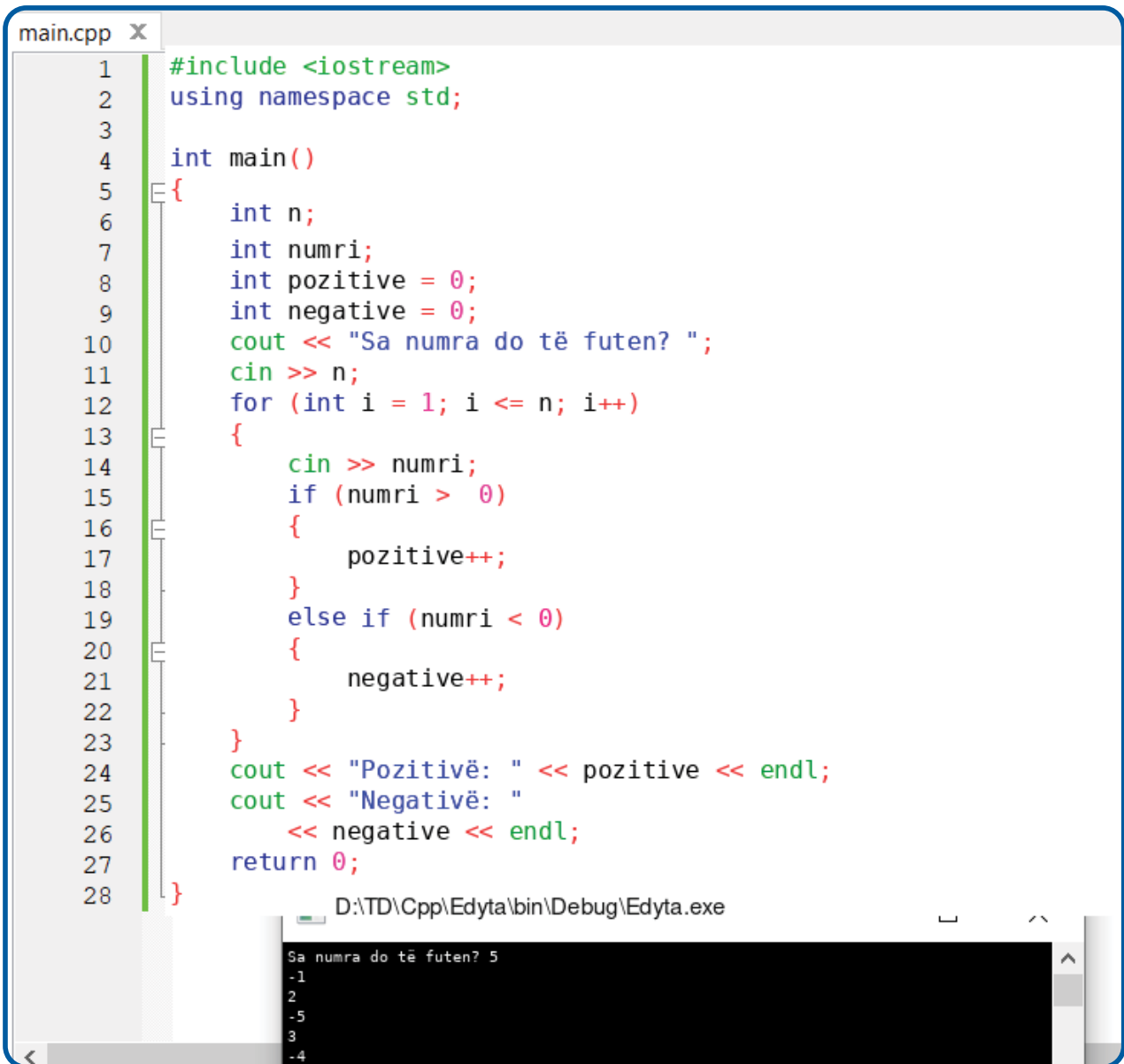
```

Sa numra do të futen? 5
24
-2
45
10
102
Maksimumi: 102

```

Gjatë zgjidhjes së detyrave më të ndërlikuara, është shumë e rëndësishme që algoritmi të testohet me të dhëna të ndryshme hyrëse. Disa programe funksionojnë saktë për vlera të vogla, por japin gabime me të dhëna më të mëdha ose të papritura. Për këtë arsye, programuesit testojnë vazhdimisht situata të ndryshme për të kontrolluar nëse programi është i qëndrueshëm dhe i saktë. Veçanërisht e rëndësishme është të analizohen rastet kufitare, siç janë futja e vlerës zero, numrave negativë ose vlerave shumë të mëdha. Me anë të një testimi të tillë, gabimet logjike dhe problemet në algoritëm zbulohen më lehtë. Në programimin profesional, testimi përbën një pjesë të domosdoshme të zhvillimit të çdo programi.

Shembull 3: Analiza e numrave pozitivë dhe negativë



```

1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      int n;
7      int numri;
8      int pozitiv = 0;
9      int negativ = 0;
10     cout << "Sa numra do të futen? ";
11     cin >> n;
12     for (int i = 1; i <= n; i++)
13     {
14         cin >> numri;
15         if (numri > 0)
16         {
17             pozitiv++;
18         }
19         else if (numri < 0)
20         {
21             negativ++;
22         }
23     }
24     cout << "Pozitivë: " << pozitiv << endl;
25     cout << "Negativë: "
26         << negativ << endl;
27     return 0;
28 }

```

D:\TD\Cpp\Edyta\bin\Debug\Edyta.exe

```

Sa numra do të futen? 5
-1
2
-5
3
-4

```

Detyrat komplekse tekstuale përbëjnë një pjesë të rëndësishme të zhvillimit të të menduarit algoritmik. Gjatë zgjidhjes së tyre, nuk mjafton vetëm njohja e sintaksës së gjuhës programuese, por kërkohet analizë e kujdesshme e problemit dhe organizim i saktë i hapave. Sa më komplekse të bëhen detyrat, aq më shumë zhvillohen aftësitë për të menduar në mënyrë logjike, për të analizuar dhe për të planifikuar zgjidhjet. Për këtë arsye, detyra të tilla përbëjnë një hap të rëndësishëm drejt zhvillimit të aftësive më të avancuara programuese.

PYETJE DHE DETYRA PËR PËRSËRITJE



PYETJE

1. Çfarë përfaqëson një detyrë komplekse tekstuale?
2. Pse është e rëndësishme që detyra të ndahet në hapa më të vegjël?
3. Pse programet duhet të testohen me të dhëna të ndryshme?
4. Çfarë roli kanë numëruesit në programe?
5. Pse është e rëndësishëm organizimi i algoritmit?



DETYRA

1. Shkruaj një program që do të lexojë n numra dhe do të llogarisë mesataren e tyre.
2. Shkruaj një program që do të numërojë sa nga numrat e futur janë të pjesëtueshëm me 3.
3. Shkruaj një program që do të gjejë numrin e dytë më të madh nga vlerat e futura.
4. Shkruaj një program që do të kontrollojë sa nga notat e futura janë të shkëlqyera.
5. Shkruaj një program që do të analizojë sa nga numrat e futur janë çift dhe sa janë tek.



MBAJ MEND ÇFARË MËSOVE

- Detyrat komplekse kërkojnë analizë të kujdesshme.
- Problemi zakonisht ndahet në disa hapa më të vegjël.
- Në detyrat e ndërlikuara kombinohen disa teknika programuese.
- Testimi është pjesë e rëndësishme e zhvillimit të programeve.
- Organizimi i mirë i algoritmit mundëson zgjidhje më të qarta dhe më efikase.



PËR ATA QË DËSHIROJNË TË DINË MË SHUMË

Në sistemet moderne softuerike, programuesit rrallëherë zgjidhin vetëm detyra të thjeshta. Një pjesë e madhe e aplikacioneve moderne përpunojnë sasi të mëdha të të dhënave dhe, njëkohësisht, kontrollojnë një numër të madh kushtesh. Sistemet bankare, dyqanet online dhe rrjetet sociale përdorin algoritme komplekse që vazhdimisht analizojnë informacione dhe marrin vendime. Për këtë arsye, aftësia për të zgjidhur detyra komplekse tekstuale përbën një nga aftësitë më të rëndësishme në programimin modern.

2.7



PROGRAMIMI MODULAR DHE KONCEPTI „PËRÇAJ DHE SUNDO“

TASHMË DI ...

- të përdorësh strukturat kushtore në C++;
- të përdorësh ciklet për përsëritjen e operacioneve;
- të kombinosh kushtet dhe ciklet në të njëjtin program;
- të analizosh efikasitetin e algoritmeve;
- të zgjidhësh detyra më të ndërlikuara programuese.

DETYRA PËR PËRSËRITJE

1. Çfarë përfaqëson optimizimi i një algoritmi?
2. Pse është e rëndësishme të zvogëlohet numri i operacioneve të panevojshme në program?
3. Shkruaj një program që do të numërojë sa numra nga 1 deri në n janë të pjesëtueshëm me 4.
4. Çfarë përfaqëson një cikël i ndërthurur?
5. Shpjego pse programet më të mëdha janë më të vështira për t'u mirëmbajtur sesa programet më të vogla.

NË PËRMBAJTJET QË VIJOJNË DO TË MËSOSH:

- të shpjegosh se çfarë përfaqëson programimi modular;
- të zbatosh parimin „përçaj dhe sundo“;
- të analizosh probleme komplekse dhe t'i ndash në detyra më të vogla;
- të organizosh zgjidhjet programuese në tërësi logjike;
- të përgatitësh programe duke përdorur funksione.

Me rritjen e kompleksitetit të programeve, shfaqet nevoja për një organizim më të mirë të kodit dhe për një qasje më efektive ndaj zgjidhjes së problemeve. Programet e vogla mund të shkruhen relativisht lehtë si një tërësi e vetme, por në zgjidhjet më të mëdha programuese kjo qasje krijon një sërë vështirësish. Nëse të gjitha instruksionet vendosen në një vend të vetëm, programi bëhet i vështirë për t'u lexuar, kuptuar, testuar

dhe mirëmbajtur. Për këtë arsye, në programimin modern përdoren teknika që mundësojnë organizimin më të rregullt të problemeve komplekse.

Një nga parimet më të rëndësishme që përdoren në zgjidhjen e problemeve komplekse është parimi „përçaj dhe sundo“. Në vend që i gjithë problemi të trajtohet si një detyrë e vetme dhe e madhe, ai ndahet në disa detyra më të vogla, të cilat mund të analizohen dhe të zgjidhen më lehtë. Pasi secila detyrë më e vogël të jetë zgjidhur me sukses, zgjidhjet e përfutuara bashkohen për të krijuar një zgjidhje të plotë. Kjo qasje përbën bazën e programimit modular dhe përdoret pothuajse në të gjitha gjuhët moderne të programimit dhe sistemet softuerike.

Në jetën e përditshme, shumë probleme zgjidhen natyrshëm duke i ndarë në disa veprimtari më të vogla. Për shembull, gjatë organizimit të një ekskursioni shkollor, të gjitha detyrat nuk kryhen njëkohësisht. Fillimisht planifikohet itinerari, më pas organizohet transporti, rezervohet akomodimi dhe, në fund, përgatiten të gjitha dokumentet e nevojshme. Secila prej këtyre veprimtarive përbën një detyrë të veçantë, e cila mund të analizohet dhe të realizohet në mënyrë të pavarur.

E njëjta ide zbatohet edhe në programim. Në vend që programuesi të përpiqet ta zgjidhë të gjithë problemin menjëherë, ai e ndan atë në disa tërësi logjike. Në këtë mënyrë, zgjidhja bëhet më e qartë dhe më e lehtë për t'u zhvilluar. Për më tepër, gabimet zbulohen më lehtë, sepse secila pjesë e programit mund të analizohet në mënyrë të veçantë.

Kjo mënyrë pune përbën thelbin e programimit modular. Modulet zakonisht kryejnë detyra të përcaktuara qartë dhe bashkëpunojnë me njëri-tjetrin për të arritur rezultatin përfundimtar. Edhe pse në këtë mësim ende nuk do të përdoren funksionet, do të fillohet me zhvillimin e mënyrës së të menduarit që më vonë do të mundësojë kuptimin më të lehtë të funksioneve dhe zbatimin e tyre në programe.

Shembull 1: Analiza e problemit

Detyra: Të llogaritet sukcesi i një nxënësi/nxënëseje në bazë të pesë notave.

Problemi mund të ndahet në këto detyra më të vogla:

Futja e notave.

Llogaritja e shumës së notave.

Llogaritja e mesatares.

Përcaktimi i suksesit.

Paraqitja e rezultatit.

Në këtë mënyrë, në vend që i gjithë problemi të analizohet menjëherë, ai ndahet në disa hapa të thjeshtë dhe të menaxhueshëm.

Pasi problemi të ndahet në detyra më të vogla, hapi i ardhshëm është organizimi i tyre. Është shumë e rëndësishme që secila detyrë të ketë një rol të përcaktuar qartë. Për shembull, pjesa e programit që është përgjegjëse për futjen e të dhënave nuk duhet të kryejë njëkohësisht edhe llogaritje të ndërlikuara. Po ashtu, pjesa që llogarit rezultatet nuk duhet të jetë përgjegjëse edhe për paraqitjen e të dhënave në ekran.

Një ndarje e tillë mundëson që programi të jetë më i organizuar dhe më i lehtë për t'u kuptuar. Në sistemet e mëdha softuerike, një program mund të përmbajë dhjetëra ose qindra tërësi logjike që komunikojnë me njëra-tjetrën. Nëse organizimi nuk është i mirë, kodi bëhet kaotik dhe vështirë mund të zgjerohet.

Për këtë arsye, që në fazat e hershme të mësimin të programimit duhet të zhvillohet shprehia për ndarjen e problemit dhe organizimin e zgjidhjes në tërësi logjike të qarta. Kjo mënyrë të menduari më vonë e lehtëson ndjeshëm përdorimin e funksioneve, bibliotekave dhe teknikave të tjera programuese.

Shembull 2: Pseudokod sipas parimit „përçaj dhe sundo“

Detyra: Të llogaritet çmimi përfundimtar i një produkti me mundësi zbritjeje.

Problemi mund të ndahet në këto detyra më të vogla:

Futja e çmimit të produktit.

Futja e sasisë.

Llogaritja e shumës totale.

Kontrollimi nëse aplikohet zbritja.

Llogaritja e çmimit përfundimtar.

Paraqitja e rezultatit.

Pseudokodi:

Fillimi

Fut çmimin

Fut sasinë

Llogarit shumën = çmimi * sasia

Nëse shuma > 5000 atëherë

 Llogarit zbritjen = shuma * 0.10

 Llogarit çmimin përfundimtar = shuma - zbritja

Përndryshe

 Çmimi përfundimtar = shuma

Fund nëse

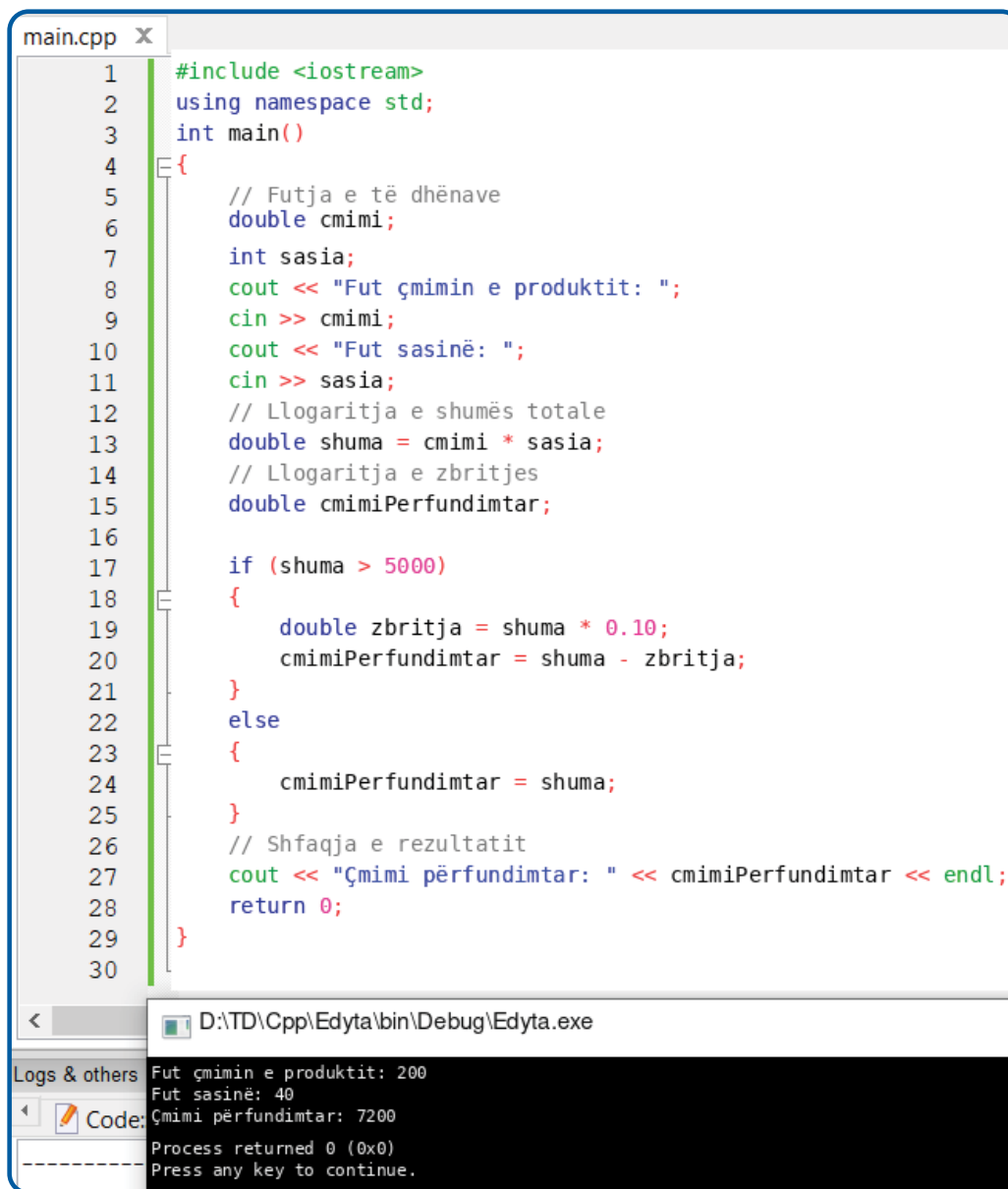
Shfaq çmimin përfundimtar

Fundi

Në këtë shembull mund të vërehet se problemi fillimisht është ndarë në disa detyra më të vogla. Secila detyrë ka një rol të përcaktuar qartë në zgjidhje. Falë këtij organizimi, algoritmi bëhet më i lehtë për t'u kuptuar dhe është i gatshëm për realizimin e mëtejshëm në një gjuhë programuese. Pikërisht kjo mënyrë e të menduarit përfaqëson thelbin e parimit „përçaj dhe sundo“ dhe përbën bazën e programimit modular.

Shembull 3: Program i organizuar në tërësi logjike

Në shembullin vijues do të paraqitet se si një program mund të organizohet në disa tërësi logjike, edhe pse ende nuk përdoren funksione. Programi llogarit çmimin përfundimtar të një produkti me mundësi zbritjeje.



```
main.cpp x
1  #include <iostream>
2  using namespace std;
3  int main()
4  {
5      // Futja e të dhënave
6      double cmimi;
7      int sasia;
8      cout << "Fut çmimin e produktit: ";
9      cin >> cmimi;
10     cout << "Fut sasinë: ";
11     cin >> sasia;
12     // Llogaritja e shumës totale
13     double shuma = cmimi * sasia;
14     // Llogaritja e zbritjes
15     double cmimiPerfundimtar;
16
17     if (shuma > 5000)
18     {
19         double zbritja = shuma * 0.10;
20         cmimiPerfundimtar = shuma - zbritja;
21     }
22     else
23     {
24         cmimiPerfundimtar = shuma;
25     }
26     // Shfaqja e rezultatit
27     cout << "Çmimi përfundimtar: " << cmimiPerfundimtar << endl;
28     return 0;
29 }
30
```

D:\TD\Cpp\Edyta\bin\Debug\Edyta.exe

Logs & others Fut çmimin e produktit: 200
Fut sasinë: 40
Çmimi përfundimtar: 7200
Process returned 0 (0x0)
Press any key to continue.

Në program mund të dallohen qartë tri tërësi logjike. Fillimisht bëhet futja e të dhënave, më pas ato përpunohen përmes llogaritjes së shumës dhe zbritjes, dhe në fund shfaqet rezultati. Edhe pse programi është shkruar brenda funksionit main(), tërësitë logjike tashmë mund të dallohen lehtësisht. Në mësimet e ardhshme, këto tërësi do të veçohen në funksione të veçanta. Në këtë mënyrë, programet do të bëhen më të organizuara, më të lehta për t'u kuptuar dhe më të thjeshta për t'u mirëmbajtur. Ky përbën hapin e parë praktik drejt programimit të vërtetë modular.

PYETJE DHE DETYRA PËR PËRSËRITJE



PYETJE

1. Çfarë përfaqëson programimi modular?
2. Çfarë përfaqëson parimi „përçaj dhe sundo“?
3. Pse problemet komplekse ndahen në detyra më të vogla?
4. Si ndihmon qasja modulare në zbulimin e gabimeve?
5. Pse është e rëndësishme që programi të jetë i organizuar në tërësi logjike?



DETYRA

1. Analizo një program për llogaritjen e pagës dhe ndaje atë në tërësi logjike.
2. Analizo një program për llogaritjen e konsumit të karburantit dhe përcakto se cilat detyra duhet të kryhen sipas radhës.
3. Shkruaj pseudokod për një program që do të llogarisë mesataren e katër testeve.
4. Për një program për evidencën e bibliotekës, përcakto të paktën katër tërësi logjike.
5. Shpjego se si do të ndihmonte parimi „përçaj dhe sundo“ gjatë zhvillimit të një sistemi informativ shkollor.



MBAJ MEND ÇFARË MËSOVE

- Programimi modular përfaqëson organizimin e programit në tërësi më të vogla logjike.
- Parimi „përçaj dhe sundo“ mundëson zgjidhjen e problemeve komplekse përmes ndarjes së tyre në detyra më të vogla.
- Organizimi i mirë i programit mundëson kuptimin, testimin dhe mirëmbajtjen më të lehtë të kodit.
- Tërësitë logjike përbëjnë bazën për përdorimin e funksioneve.
- Mendimi modular përfaqëson një aftësi të rëndësishme për çdo programues.



PËR ATA QË DËSHIROJNË TË DINË MË SHUMË

Në kompanitë e mëdha softuerike, rrallë ndodh që një programues të punojë i vetëm në të gjithë programin. Në vend të kësaj, programi ndahet në një numër të madh modulesh dhe secili ekip zhvillon një pjesë të caktuar të sistemit. Për shembull, në një dyqan online, një ekip mund të punojë në regjistrimin e përdoruesve, një tjetër në sistemin e pagesave, një ekip i tretë në përpunimin e porositë dhe një i katërt në raportet statistikore. Edhe pse secili ekip punon në një modul të ndryshëm, në fund të gjitha modulet funksionojnë së bashku si një sistem i vetëm. Pikërisht për këtë arsye, programimi modular përfaqëson një nga konceptet më të rëndësishme në zhvillimin modern të softuerit dhe një bazë të rëndësishme për krijimin e zgjidhjeve të mëdha dhe komplekse programuese.



2.8

DEFINIMI I FUNKSIONEVE NË C++

TASHMË DI ...

- të shpjegosh se çfarë përfaqëson programimi modular;
- të zbatosh parimin „përçaj e sundo“;
- të analizosh probleme komplekse dhe t'i ndash në detyra më të vogla;
- të organizosh një program në tërësi logjike;
- të përdorësh kushtet dhe ciklet në C++

DETYRA PËR PËRSËRITJE

1. Çfarë përfaqëson programimi modular?
2. Çfarë përfaqëson parimi „përçaj dhe sundo“?
3. Pse problemet e mëdha ndahen në detyra më të vogla?
4. Përmend tri tërësi logjike që mund të ekzistojnë në një program.
5. Shpjego si ndihmon organizimi i mirë i programit në zbulimin e gabimeve.

NË PËRMBAJTJET QË VIJOJNË DO TË MËSOSH:

- të shpjegosh se çfarë përfaqëson një funksion;
- të përkufizosh funksione të thjeshta në C++;
- të thërrasësh funksione nga programi kryesor;
- të dallosh përkufizimin nga thirrja e funksionit;
- të organizosh programe me disa funksione.

Ndërsa programet bëhen më të mëdha, lind nevoja për një organizim më të mirë të kodit. Në mësimin e mëparshëm u shpjegua se problemet komplekse mund të ndahen në disa detyra më të vogla. Kjo mënyrë e të menduarit përbën bazën e programimit modular. Megjithatë, që kjo ndarje të mund të zbatohet praktikisht në një gjuhë programuese, nevojitet një mekanizëm që mundëson vendosjen e secilës detyrë në një tërësi të veçantë programuese.

Në gjuhën programuese C++, këtë rol e kanë funksionet. Funkzioni është një pjesë e programit (nënprogram) dhe përfaqëson një bllok të emërtuar instruksionesh që kryen një detyrë të caktuar. Në vend që i njëjti kod të shkruhet disa herë në program, ai mund të vendoset në një funksion dhe më pas të thirret sa herë që është e nevojshme. Falë kësaj, programet bëhen më të rregullta, më të shkurtra dhe më të lehta për t'u mirëmbajtur. Për këtë arsye, funksionet përfaqësojnë një nga mjetet më të rëndësishme në programimin modern.

Funksioni mund të imagjinohet si një program i vogël brenda një programi më të madh. Çdo funksion ka një emër përmes të cilit mund të thirret dhe përmban instruksione që do të ekzekutohen kur ai të aktivizohet. Çdo program në C++ ka të paktën një funksion – main(). Ai përfaqëson pikën fillestare të ekzekutimit të programit. Përveç main(), programuesi mund të krijojë edhe funksionet e veta për kryerjen e detyrave të ndryshme. Kjo qasje mundëson organizimin e programit në disa pjesë, në vend që i gjithë kodi të vendoset në një funksion të vetëm. Kur çdo funksion ka një detyrë të përcaktuar qartë, programi bëhet shumë më i lehtë për t'u lexuar dhe kuptuar. Gjithashtu, nëse shfaqet ndonjë gabim, vëmendja mund të përqendrohet në funksionin konkret, në vend që të kontrollohet i gjithë programi. Prandaj, përdorimi i funksioneve përbën një hap të rëndësishëm në zhvillimin e shprehive të mira programuese.

Gjatë përcaktimit të një funksioni, para emrit të tij vendoset tipi i funksionit. Tipi përcakton nëse funksioni do të kthejë një rezultat pas ekzekutimit të tij dhe çfarë lloji do të jetë ai rezultat.

Në këtë mësim përdoren tipet:

void

Fjala kyçe void tregon se funksioni nuk kthen asnjë vlerë. Funksione të tilla zakonisht përdoren për të kryer një detyrë të caktuar, si shfaqja e një mesazhi në ekran, shtypja e një titulli ose ekzekutimi i udhëzimeve të caktuara, pa kthyer rezultat në program.

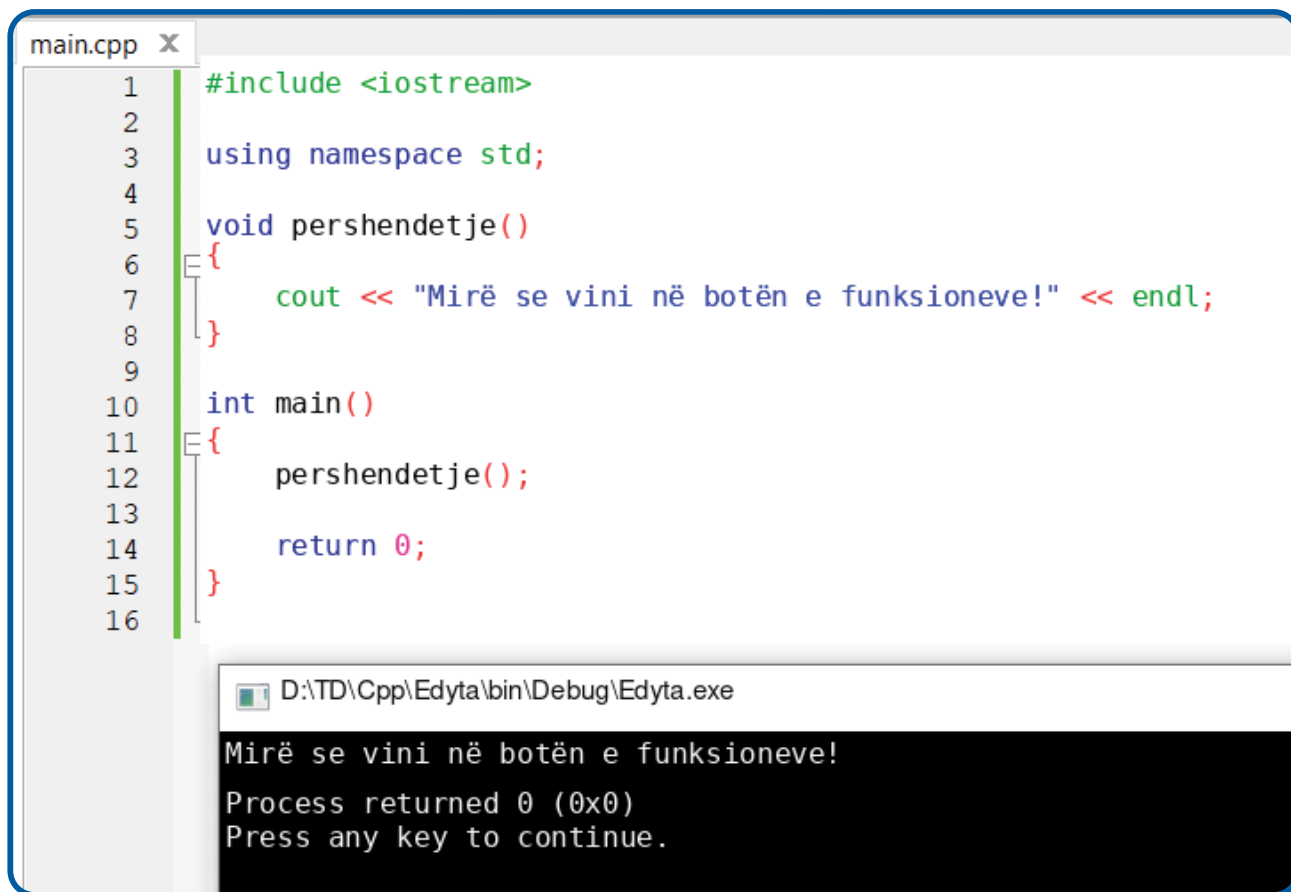
Forma e përgjithshme e një funksioni të tillë është:

```
void emriIFunksionit()
{
    udhezimet
}
```

Në programim përdoren edhe funksione që kthejnë një vlerë, për shembull:

```
int shuma()
{
    return 10;
}
```

Në këtë rast, funksioni kthen një numër të plotë (int), ndërsa funksioni i tipit void nuk kthen asnjë vlerë.



```
main.cpp x
1  #include <iostream>
2
3  using namespace std;
4
5  void pershendetje()
6  {
7      cout << "Mirë se vini në botën e funksioneve!" << endl;
8  }
9
10 int main()
11 {
12     pershendetje();
13
14     return 0;
15 }
16
```

D:\TD\Cpp\Edyta\bin\Debug\Edyta.exe

Mirë se vini në botën e funksioneve!
Process returned 0 (0x0)
Press any key to continue.

Në shembull është përkufizuar funksioni me emrin `pershendetje()`. Kur funksioni të thirret, në ekran do të shfaqet një mesazh. Është shumë e rëndësishme të theksohet se përcaktimi i funksionit nuk nënkupton edhe ekzekutimin e tij automatik. Për të ekzekutuar instruksionet brenda funksionit, ai duhet të thirret nga programi. Thirrja bëhet duke shkruar emrin e funksionit dhe kllapat.

```
pershendetje();
```

Ky mekanizëm mundëson që i njëjti funksion të përdoret disa herë pa pasur nevojë të rishkruhet i njëjti kod. Në programet reale, kjo redukton ndjeshëm gjatësinë e kodit dhe përmirëson lexueshmërinë e tij. Programuesit shpesh krijojnë funksione për detyra që përsëriten shpesh. Në këtë mënyrë shmanget përsëritja e panevojshme e instruksioneve dhe përmirësohet organizimi i programit.

Shembulli 2: Thirrja e shumëfishtë e një funksioni

```

main.cpp x
1  #include <iostream>
2
3  using namespace std;
4
5  void vija()
6  {
7      cout << "=====" << endl;
8  }
9
10 int main()
11 {
12     vija();
13     cout << "Pjesa e parë e programit" << endl;
14
15     vija();
16     cout << "Pjesa e dytë e programit" << endl;
17
18     vija();
19
20     return 0;
21 }
22
D:\TD\Cpp\Edyta\bin\Debug\Edyta.exe
=====
Pjesa e parë e programit
=====
Pjesa e dytë e programit
=====
Process returned 0 (0x0)
Press any key to continue.

```

Në shembullin e mëparshëm, i njëjti funksion është thirrur disa herë. Në vend që vargu i shenjave të shkruhet përsëri në program çdo herë, ai është vendosur në një funksion. Falë kësaj, kodi është më i shkurtër dhe më i rregullt. Kjo është një nga përparësitë më të rëndësishme të funksioneve – mundësia e ripërdorimit të kodit të shkruar më parë. Në programet e mëdha, i njëjti funksion mund të thirret dhjetëra ose qindra herë. Nëse nevojitet ndonjë ndryshim në mënyrën e funksionimit, mjafton të ndryshohet kodi vetëm brenda funksionit, pa pasur nevojë të kërkohet dhe të ndryshohet në të gjithë programin. Për këtë arsye, funksionet e lehtësojnë ndjeshëm zhvillimin dhe mirëmbajtjen e softuerit.

```
main.cpp x
1  #include <iostream>
2
3  using namespace std;
4
5  void titulli()
6  {
7      cout << "Raporti i nxënësit" << endl;
8  }
9
10 void fundi()
11 {
12     cout << "Fundi i raportit" << endl;
13 }
14
15 int main()
16 {
17     titulli();
18
19     cout << "Nxënësi arriti sukses të shkëlqyer." << endl;
20
21     fundi();
22
23     return 0;
24 }
25
```

D:\TD\Cpp\Edyta\bin\Debug\Edyta.exe

Raporti i nxënësit
Nxënësi arriti sukses të shkëlqyer.
Fundi i raportit
Process returned 0 (0x0)
Press any key to continue.

Në këtë shembull përdoren dy funksione të ndryshme. Njëri është përgjegjës për shfaqjen e titullit, ndërsa tjetri për shfaqjen e mesazhit përfundimtar. Në këtë mënyrë, detyrat në program ndahen në disa tërësi logjike.

Edhe pse programi është i vogël, mund të vërehet qartë se si funksionet kontribuojnë në organizimin më të mirë të kodit. Në mësimet e ardhshme, funksionet do të kenë mundësi shtesë përmes përdorimit të parametrave dhe vlerave kthyesë, duke u mundësuar atyre të kryejnë detyra edhe më komplekse.

PYETJE DHE DETYRA PËR PËRSËRITJE



PYETJE

1. Çfarë përfaqëson një funksion?
2. Cili është roli i funksionit `main()`?
3. Çfarë përfaqëson thirrja e një funksioni?
4. Cili është ndryshimi midis përkufizimit dhe thirrjes së një funksioni?
5. Pse funksionet e përmirësojnë organizimin e programeve?



DETYRA

1. Shkruaj një funksion që do të shfaqë mesazhin: „Mirëdita!“.
2. Shkruaj një funksion që do të shfaqë një vijë të përbërë nga 30 karaktere *.
3. Shkruaj një program që do të përdorë dy funksione: një për shfaqjen e titullit dhe një për shfaqjen e mesazhit përfundimtar.
4. Shkruaj një program që do të thërrasë tri herë të njëjtin funksion.
5. Shpjego se si ndihmojnë funksionet në zhvillimin e programeve të mëdha.



MBAJ MEND ATO QË ÇFARË MËSOVE

- Funksioni përfaqëson një bllok të emërtuar udhëzimesh.
- Funksionet mundësojnë ndarjen e programit në njësi më të vogla.
- Përkufizimi përshkruan funksionin, ndërsa thirrja e aktivizon atë.
- I njëjti funksion mund të thirret disa herë.
- Funksionet kontribuojnë në organizimin dhe mirëmbajtjen më të mirë të programeve.



PËR ATA QË DËSHIROJNË TË DINË MË SHUMË

Në gjuhët moderne të programimit ekzistojnë biblioteka që përmbajnë mijëra funksione të gatshme. Në vend që programuesit ta shkruajnë gjithmonë kodin nga fillimi, ata mund t'i përdorin këto funksione për llogaritje matematikore, punë me tekst, përpunim të skedarëve dhe shumë detyra të tjera. Falë kësaj qasjeje, zhvillimi i softuerit bëhet më i shpejtë dhe më efikas. Prandaj, funksionet përfaqësojnë një nga idetë më të rëndësishme në programim dhe një bazë për zhvillimin e sistemeve softuerike komplekse.



Në këtë mënyrë mundësohet krijimi i zgjidhjeve programore më fleksibile dhe ripërdorimi i të njëjtit kod në situata të ndryshme.

Përveç pranimi të të dhënave, shumë shpesh është e nevojshme që funksioni të kthejë një rezultat, i cili mund të përdoret në një pjesë tjetër të programit. Për shembull, funksioni mund të llogarisë sipërfaqen, katrorin e një numri ose mesataren dhe rezultatin e përfituar ta kthejë përsëri në program. Falë kësaj mundësie, funksionet bëhen shumë më të fuqishme dhe më fleksibile. Në këtë mësim do të trajtohet përdorimi i parametrave dhe do të fillojë mësimi i funksioneve që kthejnë vlerë.

Kur funksioni duhet të përpunojë të dhëna të ndryshme, ato të dhëna i dërgohen përmes parametrave. Parametrat përfaqësojnë variabla që përcaktohen në kllapat e funksionit. Gjatë çdo thirrjeje të funksionit, parametrave u caktohen vlera të reja. Në këtë mënyrë, i njëjti funksion mund të përdoret për përpunimin e të dhënave të ndryshme, pa pasur nevojë që kodi të shkruhet përsëri. Kjo qasje e rrit ndjeshëm ripërdorshmërinë e programeve dhe mundëson krijimin e zgjidhjeve më fleksibile. Në vend që të krijohet një funksion i veçantë për çdo vlerë, mund të krijohet një funksion që do të punojë me të dhëna të ndryshme. Prandaj, parametrat janë një nga karakteristikat më të rëndësishme të funksioneve. Ata mundësojnë komunikimin ndërmjet programit kryesor dhe funksionit dhe përfaqësojnë bazën për zhvillimin e zgjidhjeve programore më të ndërlikuara.

PARAMETRAT DHE ARGUMENTET

Gjatë punës me funksionet shpesh përdoren termat **parametër** dhe **argument**.

- Parametri është një variabël e përcaktuar në përkufizimin e funksionit.
- Argumenti është vlera që i dërgohet funksionit gjatë thirrjes së tij.

Shembull 1: Funksion me një parametër

```

main.cpp x
1  #include <iostream>
2
3  using namespace std;
4
5  void shfaqNumrin(int numri)
6  {
7      cout << "Numri i futur: "
8          << numri << endl;
9  }
10
11 int main()
12 {
13
14     shfaqNumrin(15);
15     return 0;
16 }
    
```

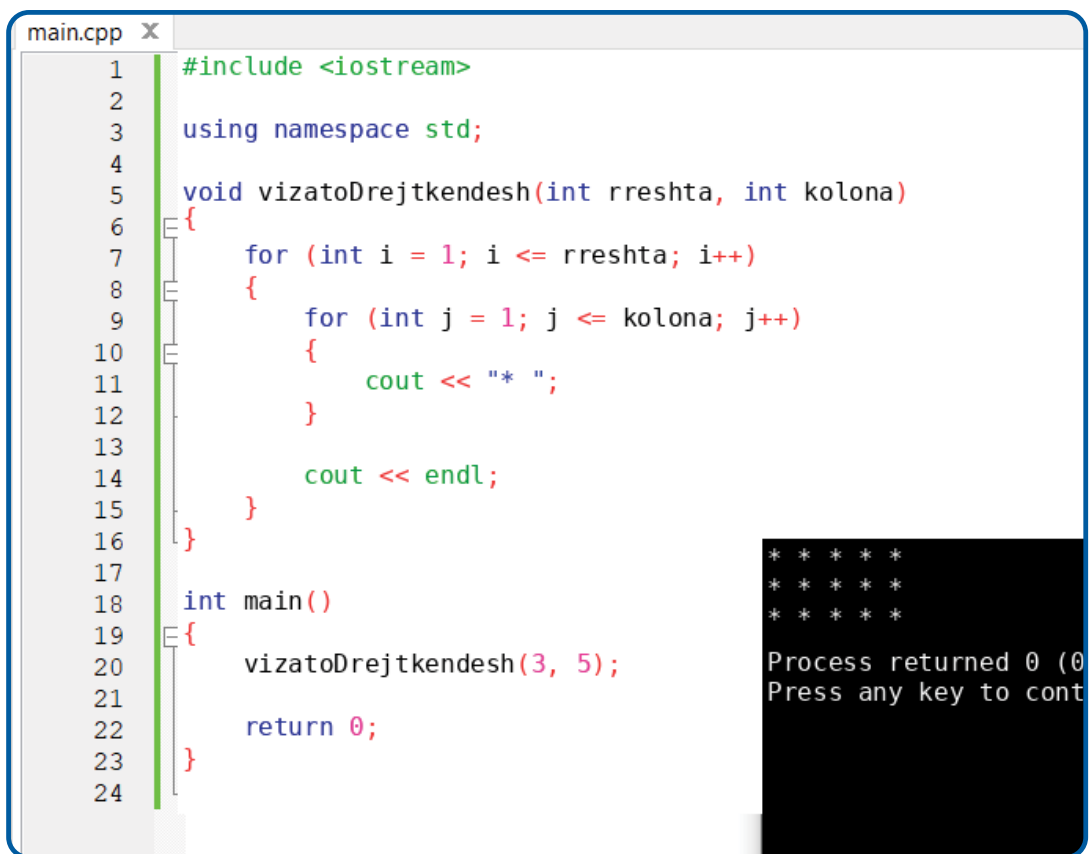
D:\TD\Cpp\Edyta\bin\Debug\Edyta.exe

Numri i futur: 15
Process returned 0 (0x0)
Press any key to continue.

Në shembullin, funksioni pranon një vlerë përmes parametrin broj. Gjatë thirrjes së funksionit, argumenti 15 i përcillet parametrin dhe më pas shfaqet në ekran. Falë parametrave, i njëjti funksion mund të përdoret me vlera të ndryshme pa bërë asnjë ndryshim në kodin e tij. Kjo qasje e rrit ndjeshëm fleksibilitetin e programeve dhe mundëson krijimin e kodit më të shkurtër dhe më të organizuar.

Funksionet mund të pranojnë disa parametra njëkohësisht. Në raste të tilla, çdo parametri i caktohet vlera e vet gjatë thirrjes së funksionit. Kjo mundëson që funksioni të punojë me më shumë të dhëna dhe të kryejë detyra më të ndërlikuara. Në praktikë, shpesh përdoren funksione që pranojnë dy, tri ose më shumë vlera. Në këtë rast, renditja e argumenteve është shumë e rëndësishme, sepse çdo vlerë i caktohet parametrin përkatës sipas pozicionit të saj. Ky mekanizëm përbën bazën e komunikimit ndërmjet pjesëve të ndryshme të programit. Falë tij, funksionet mund të përdoren në një numër të madh situatash të ndryshme, pa pasur nevojë të krijohen funksione të reja për çdo rast të veçantë.

Shembull 2: Funksion me dy parametra



```
main.cpp X
1  #include <iostream>
2
3  using namespace std;
4
5  void vizatoDrejtkenesh(int rreshta, int kolona)
6  {
7      for (int i = 1; i <= rreshta; i++)
8      {
9          for (int j = 1; j <= kolona; j++)
10         {
11             cout << "* ";
12         }
13         cout << endl;
14     }
15 }
16
17
18 int main()
19 {
20     vizatoDrejtkenesh(3, 5);
21
22     return 0;
23 }
24
```

```
* * * * *
* * * * *
* * * * *
Process returned 0 (0
Press any key to cont
```

Në shembullin, funksioni pranon dy parametra. Parametri i parë përcakton numrin e rreshtave, ndërsa i dyti numrin e kolonave. Falë kësaj, i njëjti funksion mund të shfaqë drejtkëndësha me përmasa të ndryshme, pa pasur nevojë të ndryshohet kodi i funksionit.

Këto janë funksione që pranojnë të dhëna përmes parametrave. Megjithatë, në shumë situata nuk mjafton që funksioni vetëm të kryejë një detyrë të caktuar. Shpesh lind nevoja që rezultati i përpunimit të kthehet përsëri në program, në mënyrë që të mund të përdoret më tej. Për këtë qëllim përdoren funksionet që kthejnë vlerë. Ndryshe nga funksionet void, të cilat nuk kthejnë rezultat, këto funksione kanë një tip të përcaktuar të të dhënave që do të kthehet pas ekzekutimit të tyre. Kthimi i vlerës realizohet me ndihmën e

komandës return. Kur ekzekutohet return, puna e funksionit përfundon dhe vlera e përfituar i dërgohet përsëri vendit nga i cili është thirrur funksioni. Falë kësaj mundësie, funksionet mund të përdoren si pjesë e shprehjeve, llogaritjeve dhe konstruksioneve të tjera programore. Kjo mënyrë e funksionimit e zgjeron ndjeshëm përdorimin e tyre dhe mundëson ndërtimin e zgjidhjeve programore më komplekse. Për këtë arsye, funksionet që kthejnë vlerë janë një nga mjetet më të rëndësishme në programim.

Funksionet që kthejnë vlerë

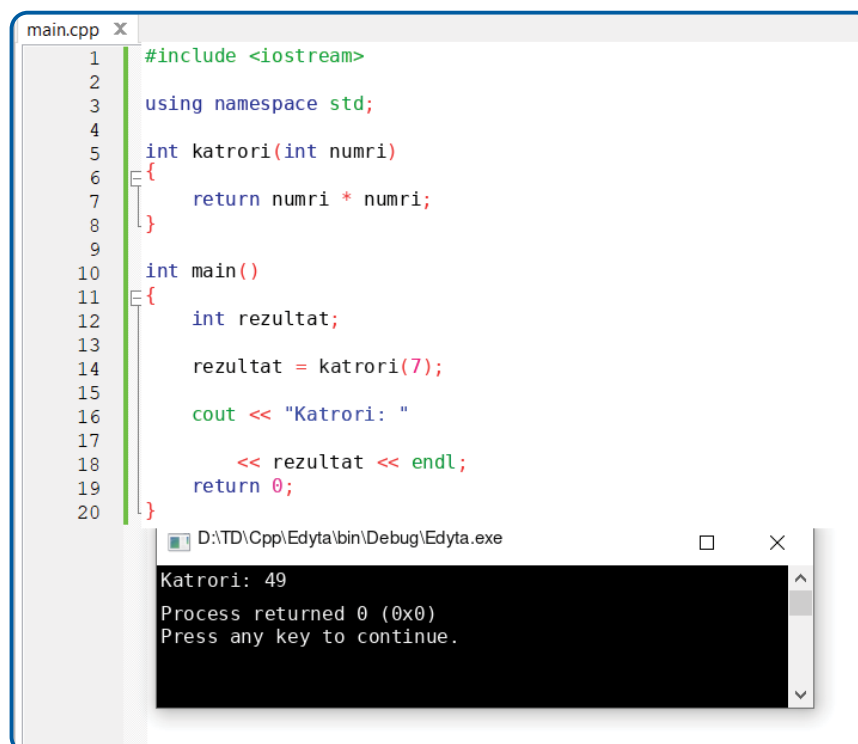
Forma e përgjithshme e një funksioni që kthen vlerë është:

```
tip emriFunksionit(parametri)
{
    udhezimet

    return vlera;
}
```

Në vend të void, para emrit të funksionit shënohet tipi i të dhënave që do të kthehet. Për shembull, nëse funksioni kthen një numër të plotë, përdoret tipi int.

Shembull 3: Funksion që kthen vlerë



```
main.cpp x
1  #include <iostream>
2
3  using namespace std;
4
5  int katrori(int numri)
6  {
7      return numri * numri;
8  }
9
10 int main()
11 {
12     int rezultat;
13
14     rezultat = katrori(7);
15
16     cout << "Katrori: "
17
18         << rezultat << endl;
19     return 0;
20 }
```

D:\TD\Cpp\Edyta\bin\Debug\Edyta.exe

Katrori: 49
Process returned 0 (0x0)
Press any key to continue.

Në shembullin, funksioni katrori() pranon një parametër dhe kthen rezultat me ndihmën e komandës return. Vlera që kthehet më pas ruhet në variablën rezultat. Falë kësaj mundësie, funksionet mund të kryejnë

llogaritje dhe rezultatet e përfituara t'ua përcjellin pjesëve të tjera të programit. Kjo përfaqëson një nga përdorimet më të zakonshme të funksioneve në programimin bashkëkohor.

Funksionet që kthejnë vlerë përdoren shpesh për llogaritje matematikore, analizë të të dhënave dhe zgjidhjen e problemeve të ndryshme programore. Në vend që e njëjta llogaritje të përsëritet disa herë në program, ajo mund të vendoset në një funksion dhe më pas të përdoret sa herë që është e nevojshme. Në këtë mënyrë, kodi bëhet më i shkurtër dhe më i kuptueshëm. Për më tepër, nëse nevojitet ndryshim në mënyrën e llogaritjes, mjafton të ndryshohet vetëm funksioni. Pjesa tjetër e programit vazhdon të funksionojë pa ndryshime. Për këtë arsye, funksionet kontribuojnë në organizim më të mirë, ripërdorim më të madh të kodit dhe mirëmbajtje më të lehtë të programeve. Në mësimet në vijim, këto koncepte do të zgjerohen më tej përmes mësimit të mënyrave të ndryshme të përcjelljes së të dhënave të funksionet dhe punës me variablat brenda fushëveprimit të tyre.

PYETJE DHE DETYRA PËR PËRSËRITJE



PYETJE

1. Çfarë përfaqëson parametri?
2. Çfarë përfaqëson argumenti?
3. Cili është ndryshimi midis një funksioni me tipin void dhe një funksioni që kthen vlerë?
4. Cili është roli i komandës return?
5. Pse funksionet me parametra janë më fleksibile se funksionet pa parametra?



DETYRA

1. Shkruaj një funksion pecatilme() që do të pranojë një emër si parametër dhe do ta shfaqë në ekran.
2. Shkruaj një funksion pogolemBroj() që do të pranojë dy numra të plotë dhe do ta kthejë numrin më të madh.
3. Shkruaj një funksion plostinaPravoagolnik() që do të pranojë gjatësinë dhe gjerësinë dhe do ta kthejë sipërfaqen.
4. Shkruaj një funksion kub() që do ta kthejë kubin e numrit të dhënë.
5. Shkruaj një program që do të përdorë një funksion për llogaritjen e perimetrit të katrorit dhe do ta shfaqë rezultatin.



MBAJ MEND ÇFARË MËSOVE

- Parametrat përfaqësojnë variablat e përcaktuara në përkufizimin e funksionit.
- Argumentet janë vlerat që dërgohen gjatë thirrjes së funksionit.
- Funksionet mund të pranojnë një ose më shumë parametra.
- Komanda return përdoret për kthimin e një vlere nga funksioni.
- Funksionet që kthejnë vlerë mund të përdoren në llogaritje dhe në shprehje.



PËR ATA QË DËSHIROJNË TË DINË MË SHUMË

Në sistemet e mëdha softuerike ekzistojnë funksione që pranojnë dhjetëra parametra dhe kthejnë rezultate të ndërlikuara. Megjithatë, në programimin profesional rekomandohet që funksionet të kenë një qëllim të qartë dhe të kryejnë një detyrë konkrete. Kjo qasje e bën kodin më të lehtë për t'u lexuar, testuar dhe mirëmbajtur. Shumë gjuhë moderne të programimit përmbajnë mijëra funksione të gatshme, të cilat programuesit i përdorin çdo ditë për punë me tekst, skedarë, rrjete, grafikë dhe llogaritje matematikore. Funksionet janë një nga bazat më të rëndësishme të programimit modern dhe njohja e mirë e tyre e lehtëson ndjeshëm zhvillimin e zgjidhjeve të ndërlikuara programore.





e rëndësishme, sepse zgjedhja e duhur e llojit të variablës kontribuon në organizimin më të mirë të kodit dhe në zbulimin më të lehtë të gabimeve në programe.

Çdo variabël që deklarohet në program ka një fushëveprim të caktuar. Kjo do të thotë se ekzistojnë rregulla që përcaktojnë se ku mund të përdoret një variabël dhe ku nuk mund të aksesohet. Kur një variabël deklarohet brenda një funksioni, përdorimi i saj kufizohet vetëm në atë funksion. Variabla të tilla quhen variabla lokale. Variablat lokale janë lloji më i përdorur i variablave në programe, sepse mundësojnë kontroll më të mirë mbi të dhënat dhe zvogëlojnë mundësinë e shfaqjes së gabimeve të padëshiruara. Çdo funksion mund të ketë variablat e veta lokale, të cilat nuk janë të dukshme për funksionet e tjera. Falë kësaj, funksione të ndryshme mund të punojnë në mënyrë të pavarur me të dhënat e tyre, pa ndikuar te njëri-tjetri. Ky parim kontribuon në siguri më të madhe dhe organizim më të mirë të kodit programues.

Shembull 1: Variabël lokale

```

main.cpp x
1  #include <iostream>
2
3  using namespace std;
4
5  void mesazhi()
6  {
7      int numri = 10;
8
9      cout << numri << endl;
10 }
11
12 int main()
13 {
14     mesazhi();
15
16     return 0;
17 }
18
D:\TD\Cpp\Edyta\bin\Debug\Edyta.exe
10
Process returned 0 (0x0)
Press any key to continue.

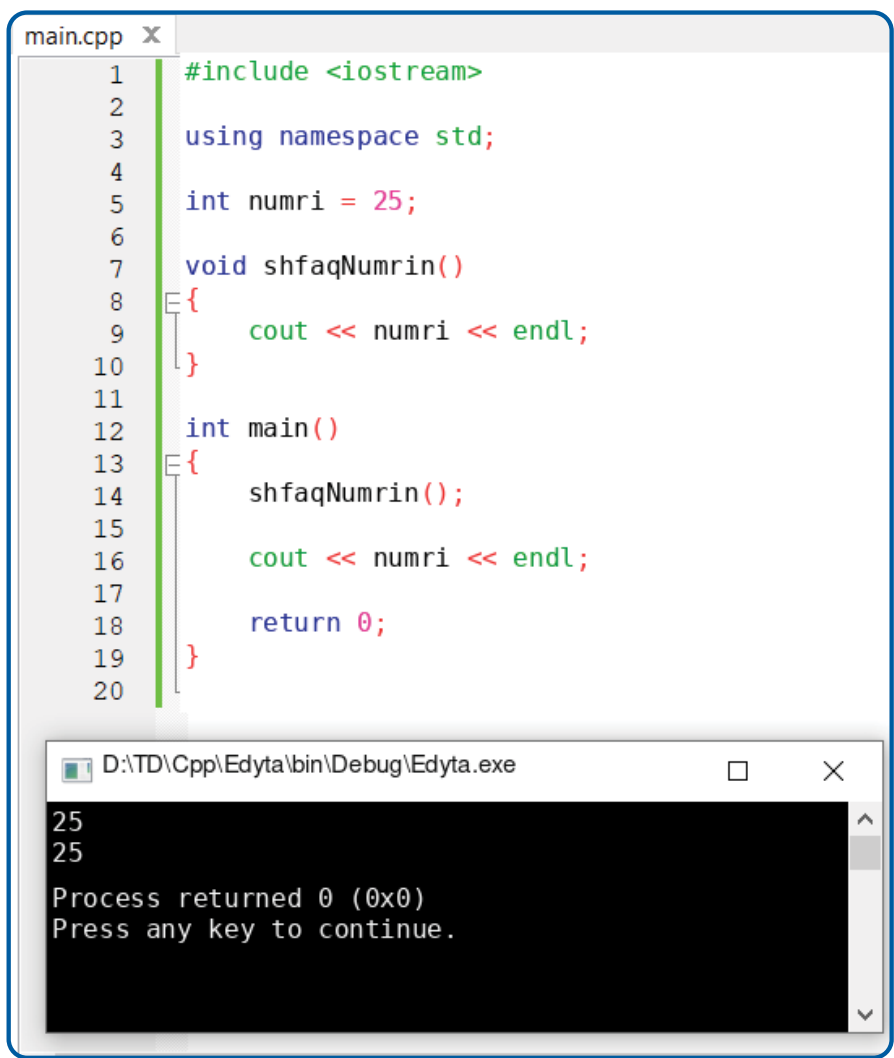
```

Në shembullin, variabla broj është deklaruar në funksionin mesazh(). Për këtë arsye, ajo mund të përdoret vetëm brenda këtij funksioni. Nëse provohet të përdoret në main() ose në një funksion tjetër, programi do të raportojë gabim. Kjo sjellje është rezultat i fushëveprimit të kufizuar të variablave lokale.

Përveç variablave lokale, në programe mund të përdoren edhe variabla globale. Ndryshe nga variablat lokale, variablat globale deklarohen jashtë të gjitha funksioneve. Për këtë arsye, ato mund të aksesohen nga disa funksione në të njëjtin program. Kjo mundëson që të dhëna të caktuara të ndahen ndërmjet pjesëve të ndryshme të programit.

Megjithatë, kjo qasje duhet të përdoret me kujdes. Kur një numër i madh funksionesh përdorin dhe ndryshojnë të njëjtën variabël globale, mund të bëhet më e vështirë të ndiqet rrjedha e të dhënave dhe të zbulohen gabimet e mundshme. Për këtë arsye, në programimin modern, zakonisht u jepet përparësi variablave lokale, ndërsa variablat globale përdoren vetëm kur ekziston nevojë reale për ndarjen e të dhënave ndërmjet funksioneve.

Shembull 2: Variabël globale



The image shows a code editor window titled 'main.cpp' and a separate console window titled 'D:\TD\Cpp\Edyta\bin\Debug\Edyta.exe'. The code in the editor defines a global variable 'numri' and a function 'shfaqNumrin()' that prints its value. The 'main()' function calls 'shfaqNumrin()' and then prints 'numri' directly. The console output shows the number '25' printed twice, followed by the process return message.

```
main.cpp X
1  #include <iostream>
2
3  using namespace std;
4
5  int numri = 25;
6
7  void shfaqNumrin()
8  {
9      cout << numri << endl;
10 }
11
12 int main()
13 {
14     shfaqNumrin();
15
16     cout << numri << endl;
17
18     return 0;
19 }
20
```

D:\TD\Cpp\Edyta\bin\Debug\Edyta.exe

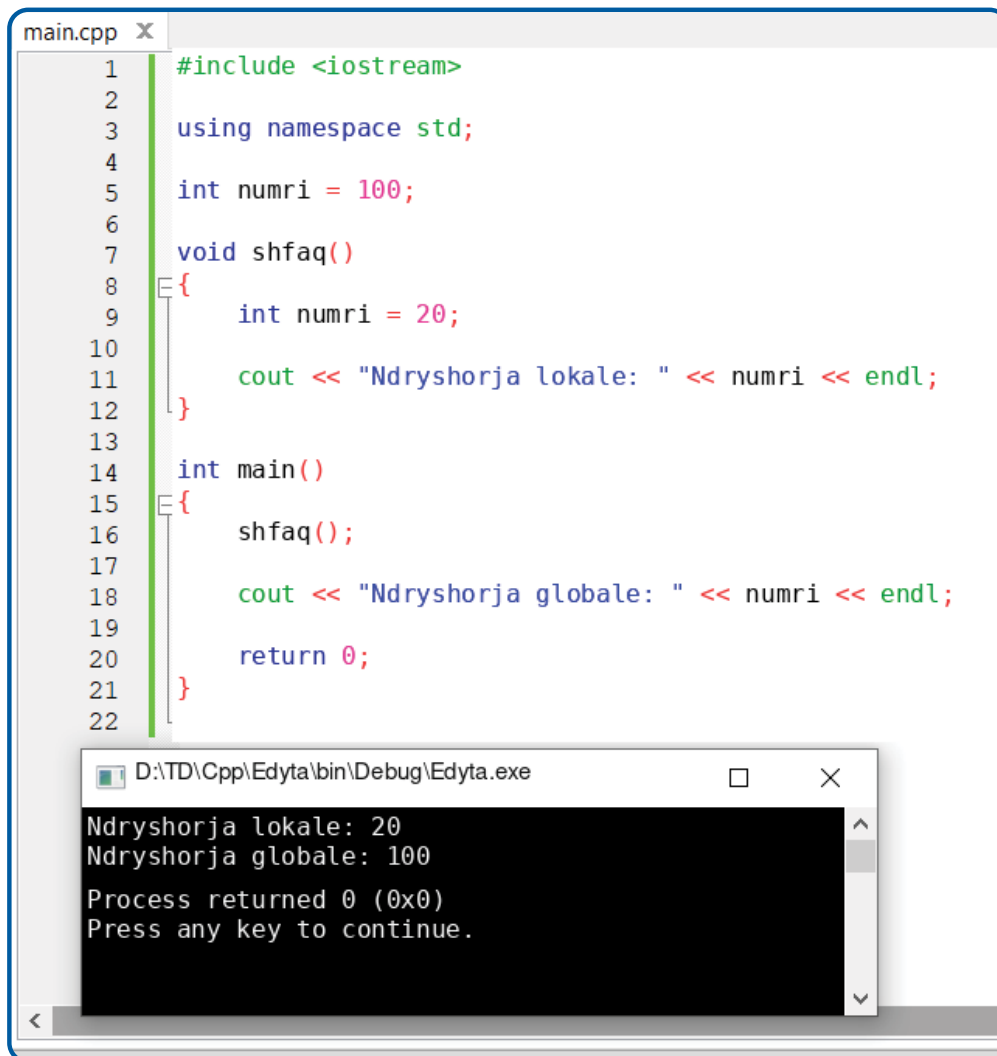
25
25
Process returned 0 (0x0)
Press any key to continue.

Në këtë shembull, variabla broj është deklaruar jashtë të gjitha funksioneve. Për këtë arsye, ajo mund të përdoret si në funksionin shfaqNumrin(), ashtu edhe në funksionin main().

Gjatë punës me disa funksione, mund të paraqitet një situatë në të cilën një variabël lokale dhe një variabël globale kanë të njëjtin emër. Në një rast të tillë, përparësi ka variabla lokale brenda funksionit në të cilin është deklaruar. Kjo do të thotë se në atë funksion nuk do të përdoret variabla globale me të njëjtin emër. Një situatë e tillë mund të shkaktojë konfuzion te programuesit, veçanërisht në programe më të mëdha me një numër të madh variablash.

Për këtë arsye, gjatë emërtimit të variablave duhet të zgjidhen emra të qartë dhe të njëkuptimshëm. Organizimi i mirë i variablave kontribuon në lexueshmëri më të mirë të kodit dhe në zbulimin më të lehtë të gabimeve logjike. Përveç kësaj, përdorimi i variablave lokale mundëson që çdo funksion të jetë i pavarur nga pjesët e tjera të programit. Në këtë mënyrë zvogëlohet mundësia që pa dashje të ndryshohet vlera të cilën e përdor një funksion tjetër. Pikërisht për këtë arsye, variablat lokale zakonisht konsiderohen mënyra e rekomanduar për të punuar me të dhënat në zgjidhjet programuese.

Shembull 3: Variabël lokale dhe globale me të njëjtin emër



```

main.cpp x
1  #include <iostream>
2
3  using namespace std;
4
5  int numri = 100;
6
7  void shfaq()
8  {
9      int numri = 20;
10
11     cout << "Ndryshorja lokale: " << numri << endl;
12 }
13
14 int main()
15 {
16     shfaq();
17
18     cout << "Ndryshorja globale: " << numri << endl;
19
20     return 0;
21 }
22

```

```

D:\TD\Cpp\Edyta\bin\Debug\Edyta.exe
Ndryshorja lokale: 20
Ndryshorja globale: 100
Process returned 0 (0x0)
Press any key to continue.

```

Në shembull kemi dy variabla me të njëjtin emër. Në funksionin shfaq() përdoret variabla lokale me vlerën 20, ndërsa në main() përdoret variabla globale me vlerën 100. Ky shembull tregon se vendi i deklarimit ka ndikim të madh në përcaktimin se cila variabël do të përdoret në program.

Zgjedhja ndërmjet variablave lokale dhe globale varet nga problemi konkret që po zgjidhet. Variablat lokale mundësojnë kontroll më të mirë mbi të dhënat dhe zvogëlojnë mundësinë e ndryshimeve të padëshiruara në program. Nga ana tjetër, variablat globale mundësojnë që të njëjtat të dhëna të përdoren në disa funksione, pa pasur nevojë që këto vlera të kalohen vazhdimisht si parametra.

Megjithatë, përdorimi i tepërt i variablave globale mund ta bëjë programin më të vështirë për t'u kuptuar dhe mirëmbajtur. Për këtë arsye, në praktikë zakonisht rekomandohet që fillimisht të përdoren variablat lokale, ndërsa variablat globale të përdoren vetëm kur ekziston një nevojë e qartë që të dhënat të përdoren në disa pjesë të programit.

Kjo qasje kontribuon në krijimin e zgjidhjeve programuese më të rregullta, më të sigurta dhe më të lehta për mirëmbajtje.

PYETJE DHE DETYRA PËR PËRSËRITJE



PYETJE

1. Çfarë përfaqëson fushëveprimi i një variable?
2. Kur konsiderohet një variabël lokale?
3. Kur konsiderohet një variabël globale?
4. Cili është ndryshimi kryesor midis variablave lokale dhe globale?
5. Pse përdorimi i tepërt i variablave globale mund të shkaktojë probleme?



DETYRA

1. Shkruaj një program që do të përmbajë një variabël lokale brenda një funksioni dhe do ta shfaqë vlerën e saj.
2. Shkruaj një program që do të përdorë një variabël globale në dy funksione të ndryshme.
3. Shkruaj një program që do të përmbajë një variabël lokale dhe një variabël globale me emra të ndryshëm dhe do t'i shfaqë vlerat e tyre.
4. Analizo në cilat funksione mund të përdoret një variabël globale e deklaruar jashtë të gjitha funksioneve.
5. Shpjego në cilat situata është më mirë të përdoret një variabël lokale në vend të një variabli globale.



MBAJ MEND ÇFARË MËSOVE

- Fushëveprimi i variablës përcakton se ku mund të përdoret ajo.
- Variablat lokale deklarohen brenda një funksioni dhe mund të përdoren vetëm brenda tij.
- Variablat globale deklarohen jashtë të gjitha funksioneve.
- Variablat globale mund të përdoren në disa funksione.
- Variablat lokale zakonisht mundësojnë organizim më të mirë dhe siguri më të madhe të programit.



PËR ATA QË DËSHIROJNË TË DINË MË SHUMË

Në sistemet e mëdha programuese mund të ekzistojnë mijëra variabla. Nëse të gjitha do të ishin globale, ndjekja e vlerave të tyre do të ishte jashtëzakonisht e vështirë. Për këtë arsye, praktikat moderne të programimit rekomandojnë kufizimin e përdorimit të variablave globale dhe përdorimin e variablave lokale sa herë që është e mundur. Kjo qasje mundëson organizim më të mirë të kodit programues, testim më të lehtë të funksioneve dhe zvogëlim të numrit të gabimeve logjike gjatë zhvillimit të zgjidhjeve softuerike.





2.11

FUNKSIONET THEMELORE MATEMATIKORE NGA BIBLIOTEKA CMATH (PJESA 1)

TASHMË DI ...

- të përcaktosh dhe thërrasësh funksione;
- të përdorësh parametra në funksione;
- të përdorësh funksione që kthejnë vlerë;
- të dallosh variablat lokale nga ato globale;
- të organizosh programin në disa tërësi logjike.

DETYRA PËR PËRSËRITJE

1. Çfarë përfaqëson një variabël lokale?
2. Çfarë përfaqëson një variabël globale?
3. Në cilën pjesë të programit mund të përdoret një variabël lokale?
4. Cila është përparësia kryesore e variablave lokale?
5. Shkruaj një funksion që pranon një numër dhe kthen katrorin e tij.

NË PËRMBAJTJET QË VIJOJNË DO TË MËSOSH:

- të përdorësh bibliotekën cmath;
- të zbatosh funksione matematikore të gatshme;
- të llogaritësh rrënjën katrore dhe fuqinë e një numri;
- të përdorësh funksione për rrumbullakim;
- të zbatosh funksione trigonometrike;
- të përdorësh funksione matematikore në zgjidhje programuese.

Gjatë hartimit të programeve shpesh lind nevoja për kryerjen e llogaritjeve matematikore. Një pjesë e këtyre llogaritjeve mund të realizohen lehtësisht me operatorët aritmetikorë, por për disa operacione

të caktuara, nevojiten algoritme më të ndërlikuara. Për shembull, llogaritja e rrënjës katrore, fuqizimi ose funksionet trigonometrike do të kërkonte një numër dukshëm më të madh instruksionesh nëse do të realizoheshin manualisht.

Për t'ua lehtësuar punën programuesve, gjuha programuese C++ përmban një numër të madh funksionesh të gatshme, të organizuara në biblioteka të ndryshme. Një nga bibliotekat më të përdorura është `cmath`, e cila përmban funksione për llogaritje matematikore. Me përdorimin e tyre, operacionet komplekse mund të kryhen shpejt dhe thjesht, pa pasur nevojë të shkruhet kod shtesë. Falë kësaj, programet bëhen më të shkurtra, më të kuptueshme dhe më të lehta për mirëmbajtje.

Biblioteka `cmath` përfaqëson një grup funksionesh matematikore të gatshme që mund të përdoren në programe. Për t'i përdorur ato, në fillim të programit duhet të përfshihet biblioteka përmes direktivës:

```
#include <cmath>.
```

Pasi biblioteka të jetë përfshirë, programi fiton qasje në një numër të madh funksionesh që kryejnë operacione të ndryshme matematikore. Në vend që programuesi të zhvillojë vetë një algoritëm për llogaritjen e rrënjës katrore ose të fuqisë së një numri, ai mund të përdorë drejtpërdrejt funksionet që tashmë ndodhen në bibliotekë. Kjo qasje mundëson zvogëlimin e kohës së nevojshme për zhvillimin e programeve dhe kontribuon në besueshmëri më të madhe të rezultateve. Funksionet nga `cmath` janë testuar gjerësisht dhe përdoren në shumë zgjidhje profesionale softuerike. Për këtë arsye, njohja e tyre përbën një pjesë të rëndësishme të aftësive të programimit.

Shembulli 1: Funksioni `sqrt()`

Funksioni `sqrt()` përdoret për llogaritjen e rrënjës katrore të një numri.

The image shows a code editor window titled 'main.cpp' with the following C++ code:

```

1  #include <iostream>
2  #include <cmath>
3
4  using namespace std;
5
6  int main()
7  {
8      double numri = 81;
9
10     double rrenjaKatrore = sqrt(numri);
11
12     cout << "Rrënja katrore: " << rrenjaKatrore << endl;
13
14     return 0;
15 }
16

```

Below the code editor, a console window titled 'D:\TD\Cpp\Edyta\bin\Debug\Edyta.exe' displays the output:

```

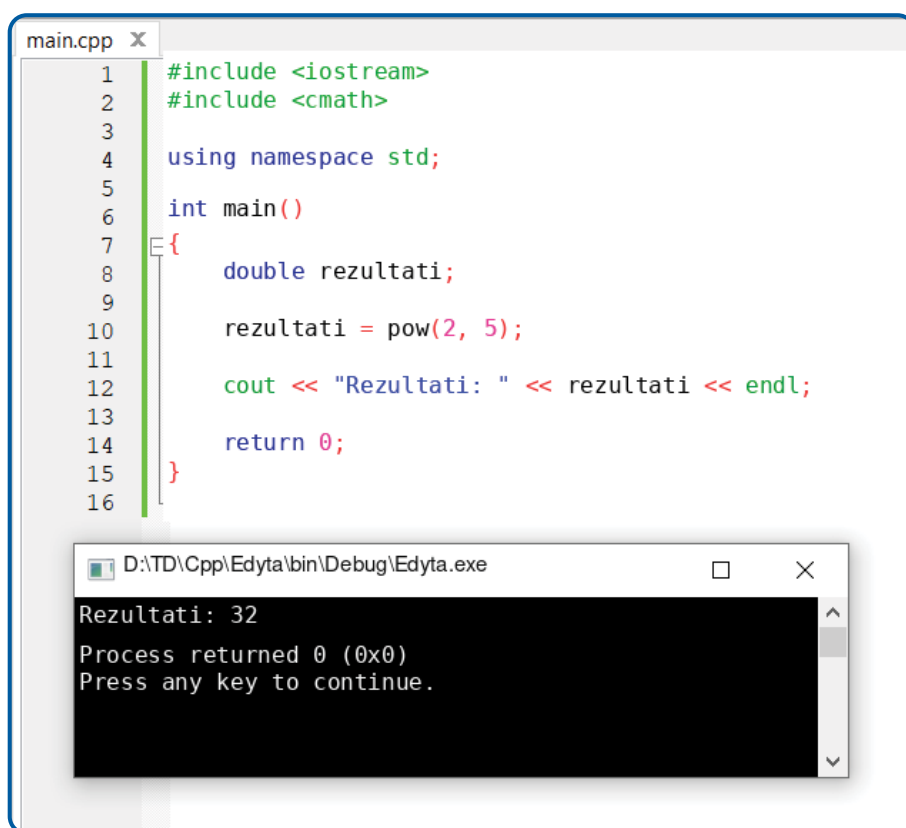
Rrënja katrore: 9
Process returned 0 (0x0)
Press any key to continue.

```

Në shembullin llogaritet rrënja katrore e numrit 81. Rezultati i marrë është 9. Funkzioni `sqrt()` pranon një numër si argument dhe kthen vlerën e rrënjës së tij katrore.

Përveç rrënjës katrore, shpesh përdoret edhe fuqizimi i numrave. Në matematikë, fuqizimi është një veprim gjatë të cilit një numër shumëzohet me vetveten një numër të caktuar herësh. Në bibliotekën `cmath`, ky veprim realizohet përmes funksionit `pow()`. Funkzioni `pow()` pranon dy argumente. Argumenti i parë paraqet bazën, ndërsa argumenti i dytë paraqet eksponentin në të cilin duhet të ngrihet numri. Falë këtij funksioni, mund të realizohen lehtësisht një numër i madh formulash dhe algoritmesh matematikore. Përdorimi i funksioneve të gatshme kontribuon që kodi të jetë më i shkurtër dhe më i kuptueshëm, krahasuar me realizimin manual të të njëjtave operacione. Për këtë arsye, funksioni `pow()` është një nga funksionet më të përdorura të bibliotekës `cmath`.

Shembulli 2: Funkzioni `pow()`



The image shows a code editor window titled 'main.cpp' with the following C++ code:

```
1 #include <iostream>
2 #include <cmath>
3
4 using namespace std;
5
6 int main()
7 {
8     double rezultati;
9
10    rezultati = pow(2, 5);
11
12    cout << "Rezultati: " << rezultati << endl;
13
14    return 0;
15 }
```

Below the code editor is a console window titled 'D:\TD\Cpp\Edyta\bin\Debug\Edyta.exe'. It displays the output of the program:

```
Rezultati: 32
Process returned 0 (0x0)
Press any key to continue.
```

Në shembullin llogaritet vlera e shprehjes 2^5 . Rezultati i marrë është 32. Argumenti i parë paraqet bazën, ndërsa argumenti i dytë – eksponentin.

Përveç funksioneve për llogaritjen e rrënjës katrore dhe fuqizimin, biblioteka `cmath` përmban edhe funksione për punën me vlera absolute dhe për rrumbullakimin e numrave. Vlera absolute e një numri paraqet largësinë e tij nga zeroja në boshtin numerik dhe gjithmonë ka vlerë jo negative. Për llogaritjen e saj përdoret funksioni `fabs()`. Gjatë punës me numra realë shpesh paraqitet nevoja që vlerat të rrumbullakohen. Për këtë qëllim përdoren funksionet `ceil()` dhe `floor()`. Funkzioni `ceil()` e rrumbullakon numrin në numrin e plotë më të vogël që është më i madh ose i barabartë me numrin fillestar, ndërsa funksioni `floor()` e rrumbullakon numrin në numrin e plotë më të madh që është më i vogël ose i barabartë me numrin fillestar. Këto funksione kanë zbatim të gjerë në zgjidhje të ndryshme programuese, veçanërisht kur është e nevojshme që rezultatet e

marra të paraqiten në formën e numrave të plotë. Falë tyre, programet mund të përpunojnë lehtësisht vlera reale dhe t'i përshtatin ato me kërkesat konkrete të problemit që zgjidhet.

Disa funksione të përdorura shpesh nga biblioteka **cmath**

Funksioni	Qëllimi
<code>sqrt(x)</code>	Rrënjë katrore
<code>pow(a,b)</code>	Fuqizim
<code>fabs(x)</code>	Vlerë absolute
<code>ceil(x)</code>	Rrumbullakim lart
<code>floor(x)</code>	Rrumbullakim poshtë
<code>sin(x)</code>	Sinus
<code>cos(x)</code>	Kosinus

Shembulli 3: Përdorimi i disa funksioneve matematikore

```

main.cpp x
1  #include <iostream>
2  #include <cmath>
3
4  using namespace std;
5
6  int main()
7  {
8      double numri = -15.7;
9
10     cout << "Vlera absolute: "
11         << fabs(numri) << endl;
12
13     cout << "Rrumbullakimi lart: "
14         << ceil(numri) << endl;
15
16     cout << "Rrumbullakimi poshtë: "
17         << floor(numri) << endl;
18
19     return 0;
20 }
21
D:\TD\Cpp\Edyta\bin\Debug\Edyta.exe
Vlera absolute: 15.7
Rrumbullakimi lart: -15
Rrumbullakimi poshtë: -16
Process returned 0 (0x0)
Press any key to continue.
  
```

Në shembull përdoren tri funksione të ndryshme nga biblioteka `cmath`. Me funksionin `fabs()` merret vlera absolute e numrit, ndërsa funksionet `ceil()` dhe `floor()` kryejnë lloje të ndryshme rrumbullakimi. Operacione të tilla përdoren shpesh gjatë përpunimit të të dhënave dhe kryerjes së llogaritjeve matematikore në programe.

Në bibliotekën `cmath` gjenden edhe funksione trigonometrike që kanë zbatim të rëndësishëm në matematikë, fizikë, inxhinieri dhe grafikën kompjuterike. Më të përdorurat janë funksionet `sin()` dhe `cos()`, të cilat llogaritin sinusin dhe kosinusin e një këndi të dhënë. Gjatë përdorimit të tyre duhet të kihet parasysh se këndet jepen në radianë. Edhe pse në matematikën shkollore zakonisht përdoren gradët, programet kompjuterike në përgjithësi përdorin radianët si njësi matëse për këndet. Funksionet trigonometrike përdoren në simulime, lojëra, animacione, aplikacione grafike dhe në shumë programe shkencore. Falë bibliotekës `cmath`, këto llogaritje mund të realizohen lehtësisht, pa pasur nevojë për zhvillimin e algoritmeve matematikore të personalizuar.

PYETJE DHE DETYRA PËR PËRSËRITJE



PYETJE

1. Pse përdoret biblioteka `cmath`?
2. Për çfarë shërben funksioni `sqrt()`?
3. Për çfarë shërben funksioni `pow()`?
4. Cili është ndryshimi midis funksioneve `ceil()` dhe `floor()`?
5. Për çfarë përdoren funksionet `sin()` dhe `cos()`?



DETYRA

1. Shkruaj një program që do të llogarisë rrënjën katrore të një numri të futur nga përdoruesi.
2. Shkruaj një program që do të llogarisë vlerën e shprehjes 3^4 duke përdorur funksionin `pow()`.
3. Shkruaj një program që do të shfaqë vlerën absolute të një numri real të futur nga përdoruesi.
4. Shkruaj një program që do të shfaqë rezultatet e funksioneve `ceil()` dhe `floor()` për një numër të futur nga përdoruesi.
5. Shkruaj një program që do të llogarisë sinusin dhe kosinusin e një këndi të dhënë në radianë.



MBAJ MEND ÇFARË MËSOVE

- Biblioteka `cmath` përmban funksione matematikore të gatshme.
- Funksioni `sqrt()` përdoret për llogaritjen e rrënjës katrore.
- Funksioni `pow()` përdoret për fuqizim.
- Funksioni `fabs()` llogarit vlerën absolute të një numri.
- Funksionet `ceil()` dhe `floor()` përdoren për rrumbullakim.
- Funksionet `sin()` dhe `cos()` përdoren për llogaritje trigonometrike.



PËR ATA QË DËSHIROJNË TË DINË MË SHUMË

Biblioteka cmath përmban shumë më tepër funksione nga ato që u trajtuan në këtë mësim. Në të mund të gjenden funksione për llogaritjen e logaritmeve, funksioneve eksponenciale, funksioneve trigonometrike dhe funksioneve trigonometrike të anasjella, si dhe një numër i madh operacionesh të tjera matematikore. Në zgjidhjet profesionale programuese, këto funksione përdoren gjatë zhvillimit të simulimeve shkencore, aplikacioneve inxhinierike, analizave financiare, lojërave kompjuterike dhe sistemeve për përpunimin e të dhënave. Njohja e bibliotekës cmath i mundëson programuesit të zgjidhë dukshëm më shpejt problemet matematikore dhe të krijojë zgjidhje programuese më efikase.





2.12

ZGJIDHJA E DETYRAVE TEKSTUALE ME ZBATIMIN E FUNKSIONEVE

TASHMË DI ...

- të përkufizosh dhe të thërrasësh funksione;
- të përdorësh parametra dhe argumente;
- të përdorësh funksione që kthejnë vlerë;
- të dallosh variablat lokale nga ato globale;
- të zbatosh funksionet nga biblioteka cmath.

DETYRA PËR PËRSËRITJE

1. Pse përdoret biblioteka cmath?
2. Për çfarë shërben funksioni `sqrt()`?
3. Cili është ndryshimi midis `ceil()` dhe `floor()`?
4. Në cilat situata përdoren funksionet që kthejnë vlerë?
5. Cili është roli i komandës `return`?

NË PËRMBAJTJET QË VIJOJNË DO TË MËSOSH:

- të analizosh një detyrë tekstuale;
- të dallosh cilat pjesë të zgjidhjes duhet të realizohen si funksione;
- të kombinosh disa funksione në një program;
- të përdorësh parametra dhe vlera kthyes gjatë zgjidhjes së problemeve;
- të krijosh zgjidhje programuese më të organizuara.

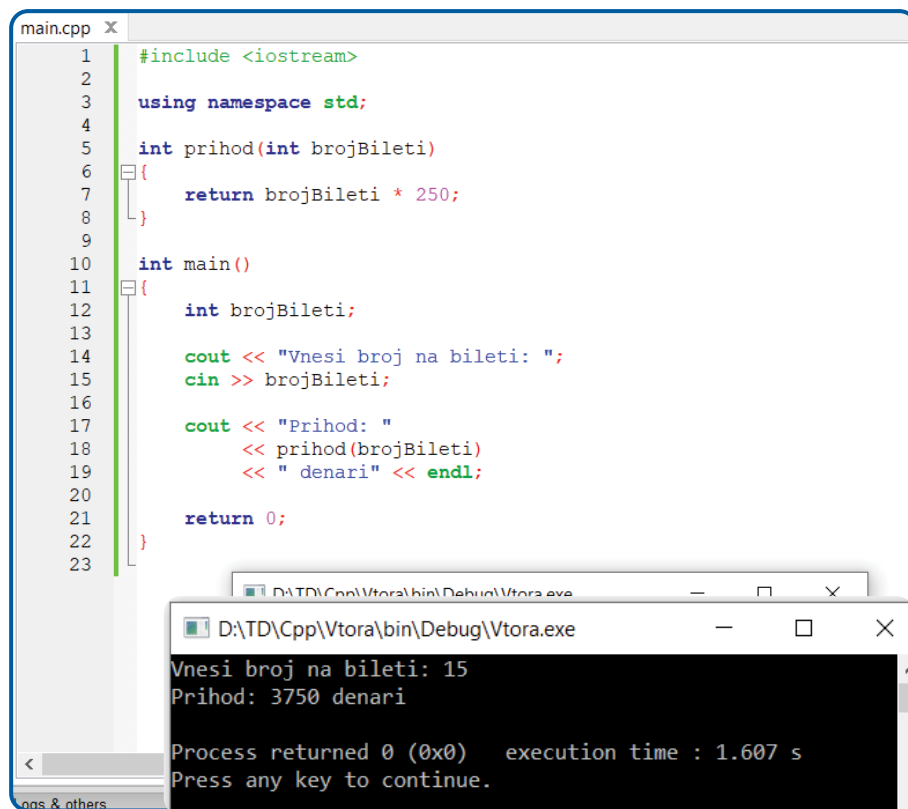
Funksionet përfaqësojnë një nga mjetet më të dobishme për organizimin e kodit programor. Vlera e tyre e vërtetë shihet më së miri gjatë zgjidhjes së problemeve reale, kur një program duhet të kryejë disa detyra të ndryshme. Në vend që të gjitha instruksionet të vendosen brenda funksionit `main()`, programi mund të ndahet në disa pjesë më të vogla, ku secili funksion ka një detyrë të përcaktuar qartë.

Gjatë zgjidhjes së detyrave tekstuale, fillimisht duhet të bëhet analiza e problemit. Më pas duhet të përcaktohet se cilat operacione përsëriten dhe cilat pjesë të zgjidhjes mund të veçohen dhe të realizohen si funksione të veçanta. Kjo qasje mundëson krijimin e programeve më të rregullta, më të kuptueshme dhe më të lehta për mirëmbajtje.

Gjatë zgjidhjes së një problemi programues, fillimisht duhet të përcaktohet se cilat të dhëna do të futen, cilat përllogaritje duhet të kryhen dhe cili rezultat duhet të paraqitet. Pas përfundimit të kësaj analize, zgjidhja mund të ndahet në disa tërësi logjike. Çdo tërësi mund të realizohet si një funksion i veçantë. Në këtë mënyrë, krijohet një program i përbërë nga disa pjesë më të vogla, në vend të një tërësie të vetme, të madhe dhe të ndërlikuar. Kjo qasje ndjek parimin „përçaj dhe sundo“ dhe përbën bazën e programimit modular.

Shembulli 1: Llogaritja e çmimit të biletave

Në kinema shiten bileta me çmim prej 250 denarësh. Të llogariten të ardhurat totale për një numër të caktuar të biletave të shitur.



```

1  #include <iostream>
2
3  using namespace std;
4
5  int prihod(int brojBileti)
6  {
7      return brojBileti * 250;
8  }
9
10 int main()
11 {
12     int brojBileti;
13
14     cout << "Vnesi broj na bileti: ";
15     cin >> brojBileti;
16
17     cout << "Prihod: "
18          << prihod(brojBileti)
19          << " denari" << endl;
20
21     return 0;
22 }
23

```

Output window showing the execution results:

```

D:\TD\Cpp\Vtora\bin\Debug\Vtora.exe
Vnesi broj na bileti: 15
Prihod: 3750 denari

Process returned 0 (0x0)   execution time : 1.607 s
Press any key to continue.

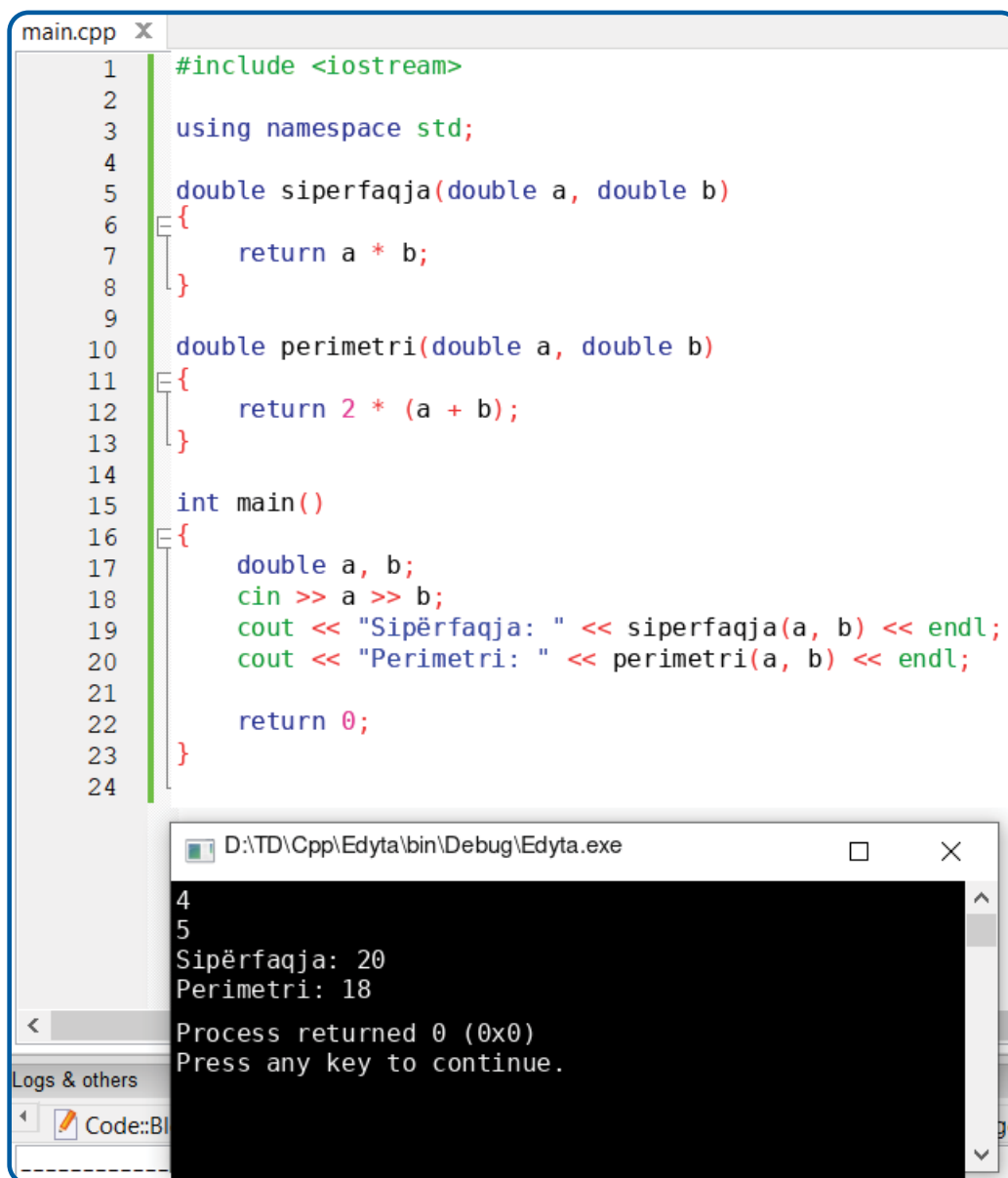
```

Në këtë shembull, llogaritja e të ardhurave është veçuar në një funksion të veçantë. Në këtë mënyrë, zgjidhja është më e rregullt dhe më e lehtë për mirëmbajtje.

Në shumë situata është e nevojshme që një program të përdorë më shumë funksione. Çdo funksion mund të kryejë një detyrë të ndryshme, ndërsa rezultatet e përfituara më pas mund të kombinohen në programin kryesor. Kjo qasje është veçanërisht e dobishme kur problemi përmban më shumë llogaritje që mund të realizohen në mënyrë të pavarur.

Shembulli 2: Llogaritja e sipërfaqes dhe perimetrit

Të futen gjatësia dhe gjerësia e një drejtkëndëshi. Të llogariten sipërfaqja dhe perimetri.



```
main.cpp X
1  #include <iostream>
2
3  using namespace std;
4
5  double siperfaqja(double a, double b)
6  {
7      return a * b;
8  }
9
10 double perimetri(double a, double b)
11 {
12     return 2 * (a + b);
13 }
14
15 int main()
16 {
17     double a, b;
18     cin >> a >> b;
19     cout << "Sipërfaqja: " << siperfaqja(a, b) << endl;
20     cout << "Perimetri: " << perimetri(a, b) << endl;
21
22     return 0;
23 }
24
```

D:\TD\Cpp\Edyta\bin\Debug\Edyta.exe

```
4
5
Sipërfaqja: 20
Perimetri: 18

Process returned 0 (0x0)
Press any key to continue.
```

Logs & others

Code::Bl

Në këtë shembull, çdo funksion ka një detyrë të përcaktuar qartë. Njëri funksion llogarit sipërfaqen, ndërsa tjetri llogarit perimetrin.

Në zgjidhjet praktike programore, shpesh nuk mjafton të përdoret vetëm një funksion. Problemet më të mëdha zakonisht përbëhen nga disa detyra më të vogla, të cilat duhet të zgjidhen në mënyrë të pavarur nga njëra-tjetra. Për këtë arsye, gjatë analizës së problemit fillimisht duhet të përcaktohet se cilat veprime do të kryhen, dhe më pas secili prej tyre të realizohet përmes një funksioni të veçantë. Në këtë mënyrë programi fiton një strukturë të qartë, ndërsa çdo funksion merr një përgjegjësi konkrete. Kur kodi programor është i organizuar në këtë mënyrë, gabimet zbulohen shumë më lehtë dhe shtohen mundësi të reja. Për më tepër, funksionet mund të ripërdoren edhe në programe të tjera, prandaj analiza e problemit dhe ndarja e duhur e detyrave përbëjnë një hap të rëndësishëm në zhvillimin e çdo zgjidhjeje programore.

Shembulli 3: Stacioni kërkimor në Mars

Në një stacion kërkimor në Mars, çdo ditë regjistrohet sasia e energjisë elektrike të prodhuar nga panelet diellore, si dhe sasia e energjisë së konsumuar nga sistemet për mbështetjen e jetës. Të përcaktohet nëse është realizuar tepërcë energjie apo deficit energjetik.

```

1  #include <iostream>
2  using namespace std;
3  double prodhoEnergji(double numriPaneleve, double energjiaPerPanel)
4  {
5      return numriPaneleve * energjiaPerPanel;
6  }
7  double bilanci(double eProdhuar, double eKonsumuar)
8  {
9      return eProdhuar - eKonsumuar;
10 }
11 int main()
12 {
13     double numriPaneleve;
14     double energjiaPerPanel;
15     double eKonsumuar;
16     cout << "Fut numrin e paneleve: ";
17     cin >> numriPaneleve;
18     cout << "Energjia për panel: ";
19     cin >> energjiaPerPanel;
20     cout << "Energjia e konsumuar: ";
21     cin >> eKonsumuar;
22     double eProdhuar = prodhoEnergji(numriPaneleve, energjiaPerPanel);
23     double rezultati = bilanci(eProdhuar, eKonsumuar);
24     cout << "Balanci energjetik: "
25          << rezultati
26          << endl;
27     return 0;
28 }
29
D:\TD\Cpp\Edyta\bin\Debug\Edyta.exe
Fut numrin e paneleve: 25
Energjia për panel: 22
Energjia e konsumuar: 473
Balanci energjetik: 77
Process returned 0 (0x0)
Press any key to continue.

```

Në këtë shembull, problemi është ndarë në dy detyra të pavarura. Në funksionin e parë llogaritet sasia totale e energjisë së prodhuar, ndërsa në funksionin e dytë llogaritet bilanci energjetik. Falë këtij organizimi, çdo funksion ka një qëllim të përcaktuar qartë, ndërsa programi është më i lehtë për t'u kuptuar.

Funksionet përbëjnë një mjet të rëndësishëm për zhvillimin e zgjidhjeve programore cilësore. Me përdorimin e tyre, problemet e ndërlikuara mund të ndahen në më shumë pjesë më të vogla dhe më të lehta për t'u menaxhuar. Në këtë mënyrë mundësohet organizim më i mirë i kodit programor, testim më i lehtë i pjesëve të veçanta të zgjidhjes dhe zbulim më efikas i gabimeve. Gjatë analizës së një detyre tekstuale, gjithmonë duhet të përcaktohet se cilat veprime mund të veçohen dhe të realizohen përmes funksioneve,

si dhe cilat të dhëna duhet të transmetohen përmes parametrave. Kjo qasje përbën bazën për zhvillimin e projekteve më të mëdha programore dhe të sistemeve moderne softuerike. Për këtë arsye, funksionet renditen ndër konceptet më të rëndësishme në gjuhën programuese C++ dhe përbëjnë bazën për studimin e mëtejshëm të strukturave për ruajtjen dhe përpunimin e të dhënave.

PYETJE DHE DETYRA PËR PËRSËRITJE

PYETJE

1. Pse është e dobishme që problemet e ndërlikuara të ndahen në disa funksione?
2. Kur është e arsyeshme të krijohet një funksion i ri?
3. Cili është roli i parametrave gjatë zgjidhjes së detyrave tekstuale?
4. Cila është përparësia e funksioneve që kthejnë një vlerë?
5. Si kontribuojnë funksionet në organizimin më të mirë të kodit programor?

DETYRA

1. Në një kopsht zoologjik mbahet evidencë për konsumin ditor të ushqimit. Shkruaj një program në të cilin do të përdoren dy funksione: njëri të llogarisë sasinë e ushqimit të nevojshëm për luanët, ndërsa tjetri – sasinë e ushqimit të nevojshëm për tigrat.
2. Në një aeroport monitorohet sasia e karburantit në avionë. Shkruaj një program në të cilin një funksion do të llogarisë sasinë totale të karburantit të furnizuar, ndërsa funksioni tjetër do të llogarisë sasinë e karburantit të mbetur pas fluturimit.
3. Në një kopsht botanik mbillen sipërfaqe rrethore me lule. Shkruaj një program në të cilin një funksion do të llogarisë sipërfaqen, ndërsa një funksion tjetër – koston e rregullimit, nëse çmimi për metër katror futet nga tastiera.
4. Në një bazë kërkimore nënujore monitorohet konsumi i oksigjenit. Shkruaj një program në të cilin do të përdoret një funksion që llogarit sasinë e oksigjenit të mbetur pas një numri të caktuar orësh pune.
5. Në një observator astronomik llogaritet koha e nevojshme që një sinjal të arrijë nga sateliti deri në Tokë. Shkruaj një program në të cilin do të përdoret një funksion që llogarit këtë vlerë, bazuar në distancën e futur.

MBAJ MEND ÇFARË KE MËSOVE

- Detyra tekstuale fillimisht duhet të analizohet.
- Problemet e ndërlikuara mund të ndahen në disa detyra më të vogla.
- Çdo detyrë mund të realizohet përmes një funksioni të veçantë.
- Funksionet mundësojnë organizim më të mirë dhe ripërdorim të kodit.
- Parametrat dhe vlerat e kthimit mundësojnë shkëmbimin e të dhënave ndërmjet funksioneve dhe programit.



PËR ATA QË DËSHIROJNË TË DINË MË SHUMË

Lojërat moderne kompjuterike, aplikacionet mobile dhe sistemet operative përmbajnë miliona rreshta kodi programor. Sisteme të tilla nuk do të mund të zhvilloheshin nëse i gjithë kodi do të vendosej në një funksion të vetëm. Për këtë arsye, programuesit i organizojnë programet në një numër të madh funksionesh dhe modulesh, ku secili funksion zgjidh një pjesë të vogël të problemit të përgjithshëm. Në këtë mënyrë mundësohet që disa programues të punojnë njëkohësisht në të njëjtin projekt, ndërsa programet bëhen më të lehta për mirëmbajtje dhe zgjerim. Me këtë përfundon tema për funksionet dhe krijohet baza për studimin e strukturave që mundësojnë përpunimin e një sasive më të madhe të të dhënave, siç janë vargjet njëdimensionale.





Detyra mesatarisht të vështira

1. Në një kështjellë mesjetare, çdo banor paguan një taksë mujore prej 12 monedhash. Shkruaj një program në të cilin do të përdorësh një funksion që pranon numrin e banorëve dhe llogarit shumën totale të taksës mujore.
2. Në një punishte kontrollohen tri dërrasa druri për ndërtimin e një kornize trekëndore. Shkruaj një program në të cilin do të përdorësh një funksion që pranon tri gjatësi dhe kontrollon nëse me to mund të formohet një trekëndësh.
3. Në një garë shkollore jepet një shenjë e veçantë për numrat që pjesëtohen njëkohësisht me 3 dhe me 5. Shkruaj një program në të cilin do të përdorësh një funksion që pranon një numër dhe kontrollon nëse e plotëson këtë kusht.
4. Në një hartë shënohet pozicioni i një objekti me koordinatat x dhe y . Shkruaj një program në të cilin do të përdorësh një funksion që pranon koordinatat dhe llogarit largësinë e objektit nga origjina e sistemit koordinativ.
5. Në një familje llogaritet fatura e energjisë elektrike. Shkruaj një program në të cilin do të ketë një ndryshore globale $\text{cmimiKwh} = 7$ dhe një funksion që pranon sasinë e kilovat-orëve të konsumuar dhe llogarit shumën për pagesë.
6. Në një shtëpi malore, një natë qëndrimi kushton 850 denarë. Shkruaj një program në të cilin do të ketë një ndryshore globale $\text{cmimiNates} = 850$ dhe një funksion që pranon numrin e netëve të qëndrimit dhe llogarit shumën totale.
7. Në një sistem informatik kontrollohet gjatësia e një numri identifikues të futur. Shkruaj një program në të cilin do të përdorësh një funksion që pranon një numër natyror dhe llogarit sa shifra përmban ai.
8. Në një disiplinë sportive krahasohen rezultatet e tre garuesve. Shkruaj një program në të cilin do të përdorësh një funksion që pranon tri rezultate dhe kthen rezultatin më të madh. Në funksion duhet të përdoret një ndryshore lokale najgolem.
9. Në një laborator shkollor llogaritet sinusi i një këndi të dhënë në gradë. Shkruaj një program në të cilin do të përdorësh një funksion që pranon këndin në gradë, e shndërron atë në radianë dhe llogarit sinusin e tij duke përdorur bibliotekën cmath .
10. Në një aplikacion financiar përpunohen vlera reale që duhet të rrumbullakohen. Shkruaj një program në të cilin do të përdorësh një funksion që pranon një numër real dhe paraqet rezultatet e funksioneve $\text{ceil}()$ dhe $\text{floor}()$.

Detyra të vështira

1. Në një muze shiten bileta për të rritur dhe për fëmijë. Shkruaj një program në të cilin do të ketë ndryshore globale $\text{cmimiTeRritur} = 200$ dhe $\text{cmimiFemije} = 120$, si dhe një funksion që pranon numrin e të rriturve dhe numrin e fëmijëve dhe llogarit të ardhurat totale.
2. Në një aplikacion kalendarik duhet të kontrollohet nëse një vit i caktuar është vit i brishtë. Shkruaj një program në të cilin do të përdorësh një funksion që pranon vitin dhe kthen nëse ai vit është i brishtë.
3. Në një sistem sigurie analizohet një numër përmes shumës së shifrave të tij. Shkruaj një program në të cilin do të përdorësh një funksion që pranon një numër natyror dhe llogarit shumën e shifrave të tij.
4. Në një aplikacion matematikor llogariten prodhimet e numrave të njëpasnjëshëm. Shkruaj një program në të cilin do të përdorësh një funksion që pranon një numër natyror dhe llogarit faktorialin e tij.
5. Në një videolojë, lojtari zhbllokoi nivele në varësi të numrit të monedhave të arta të mbledhura. Shkruaj një program në të cilin do të përdorësh një funksion që pranon numrin e monedhave të arta dhe

përcakton nivelin e zhbllokuar: deri në 100 monedha – niveli 1, nga 101 deri në 300 – niveli 2, ndërsa mbi 300 – niveli 3.

6. Në një fabrikë paktohen dy seri produktesh në grupe të barabarta, pa mbetje. Shkruaj një program në të cilin do të përdorësh një funksion që pranon dy numra dhe llogarit pjestuesin më të madh të përbashkët (PMP) të tyre.

7. Në një sistem për matjen e kohës futet kohëzgjatja në sekonda. Shkruaj një program në të cilin do të përdorësh një funksion që pranon numrin e sekondave dhe paraqet orët, minutat dhe sekondat. Në funksion duhet të përdoren ndryshoret lokale casovi, minuti dhe sekundi.

8. Në një stacion meteorologjik maten temperatura dhe lagështia. Shkruaj një program në të cilin do të përdorësh një funksion që pranon dy vlera dhe kthen nëse ekziston mundësia për mjegull, nëse temperatura është nën 5°C dhe lagështia është mbi 85%.

9. Në një aplikacion navigimi llogaritet largësia ndërmjet dy pikave në një hartë. Shkruaj një program në të cilin do të përdorësh një funksion që pranon koordinatat e dy pikave dhe llogarit largësinë ndërmjet tyre.

10. Në një observator astronomik monitorohet lëvizja e një komete. Shkruaj një program në të cilin do të përdorësh një funksion që pranon largësinë e përshkuar dhe kohën dhe llogarit shpejtësinë mesatare. Në funksion duhet të përdoret një ndryshore lokale brzina.



VARGU NJËDIMENSIONAL SI STRUKTURË E TË DHËNAVE

TASHMË DI ...

- të përdorësh ndryshore të llojeve të ndryshme;
- të futësh dhe të shfaqësh të dhëna;
- të përdorësh struktura kushtore;
- të përdorësh cikle;
- të përkufizosh dhe të thërrasësh funksione.

DETYRA PËR PËRSËRITJE

1. Çfarë përfaqëson një funksion?
2. Cili është roli i parametrave?
3. Pse përdoret komanda return?
4. Pse është e dobishme që një program të ndahet në disa funksione?
5. Përmend një funksion nga biblioteka cmath.

NË PËRMBAJTJET QË VIJOJNË DO TË MËSOSH:

- të shpjegosh se çfarë përfaqëson një varg;
- të dallosh situatat në të cilat nevojitet përdorimi i një vargu;
- të dallosh ndryshoren nga vargu;
- të shpjegosh se si ruhen elementet e vargut në memorien e kompjuterit;
- të përdorësh vargje njëdimensionale në zgjidhje të thjeshta programuese.

Gjatë zhvillimit të zgjidhjeve programuese shpesh paraqitet nevoja për përpunimin e një numri të madh të të dhënave të të njëjtit lloj. Për shembull, mund të jetë e nevojshme të ruhen notat e nxënësve të një klase, temperaturat ditore gjatë një muaji ose rezultatet e garuesve në një turne sportiv. Nëse për çdo të dhënë përdoret një ndryshore e veçantë, programi shpejt bëhet i ndërlikuar dhe i vështirë për t'u mirëmbajtur.

Për organizim më efikas të të dhënave, në gjuhën programuese C++ përdoren vargjet (array). Vargu mundëson që disa të dhëna të të njëjtit tip të ruhen nën një emër të përbashkët. Falë kësaj, të dhënat mund të futen, përpunohen dhe analizohen më lehtë. Për këtë arsye, vargjet përfaqësojnë një nga strukturat më të rëndësishme për ruajtjen e të dhënave në programim.

Ndryshorja mundëson ruajtjen e një vlere në memorien e kompjuterit. Kur është e nevojshme të ruhen dhjetë, njëzet ose njëqind vlera të të njëjtit tip, përdorimi i një numri të madh ndryshoresh të veçanta nuk është zgjidhje praktike. Në situata të tilla përdoret vargu. Vargu përfaqëson një bashkësi elementesh të të njëjtit tip të të dhënave, të cilat ruhen njëri pas tjetrit në memorien e kompjuterit. Të gjithë elementet lidhen me një emër të përbashkët, ndërsa secili element ka pozicionin e vet. Falë këtij organizimi, një numër i madh të dhënash mund të përpunohen si një tërësi e vetme. Vargjet përdoren pothuajse në të gjitha zgjidhjet moderne programuese, sepse mundësojnë përpunim efikas të të dhënave dhe përbëjnë bazën për shumë struktura më të avancuara në programim.

tipi `emriVargut[madhesia]`; vargu deklarohet duke përcaktuar tipin e të dhënave, emrin e vargut dhe numrin e elementeve që do të përmbajë. Forma e përgjithshme është:

```
tipi emriVargut[madhesia];  
Për shembull: int notat[5];
```

Në këtë shembull, fjala kyçe `int` tregon se në varg do të ruhen numra të plotë. Emri i vargut është `notat`, ndërsa numri 5 tregon se vargu përmban pesë elemente. Megjithatë vargu ka pesë elemente, pozicionet e tyre nuk fillojnë nga 1, por nga 0. Për këtë arsye, elementi i parë ndodhet në pozicionin 0, i dyti në pozicionin 1, i treti në pozicionin 2, i katërti në pozicionin 3 dhe i pesti në pozicionin 4. Rregull i përgjithshëm: në një varg me madhësi n , elementi i parë ndodhet në pozicionin 0, ndërsa elementi i fundit ndodhet në pozicionin $n - 1$.

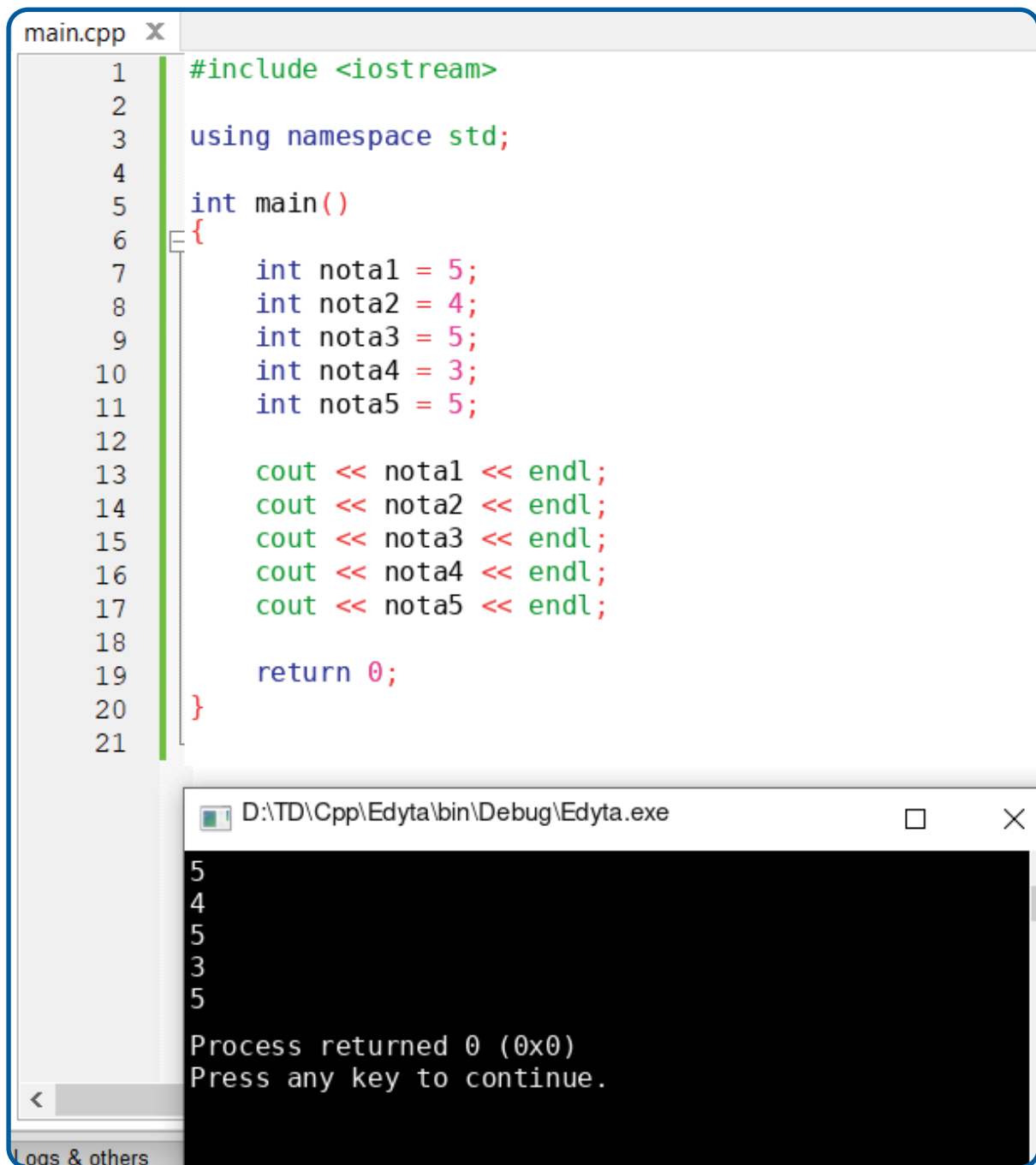
Për vargun:

```
int notat[5];  
pozicionet e elementeve janë:  
notat[0] notat[1] notat[2] notat[3] notat[4]
```

Për këtë arsye, gjatë qasjes në elemente duhet pasur gjithmonë kujdes që të mos përdoret një pozicion më i madh se vlera e fundit e lejuar, sepse kjo mund të shkaktojë funksionim të parregullt të programit.

Shembulli 1: Problemi pa përdorimin e vargut

Të paraqiten notat e pesë nxënësve.



```
main.cpp X
1  #include <iostream>
2
3  using namespace std;
4
5  int main()
6  {
7      int nota1 = 5;
8      int nota2 = 4;
9      int nota3 = 5;
10     int nota4 = 3;
11     int nota5 = 5;
12
13     cout << nota1 << endl;
14     cout << nota2 << endl;
15     cout << nota3 << endl;
16     cout << nota4 << endl;
17     cout << nota5 << endl;
18
19     return 0;
20 }
21
```

D:\TD\Cpp\Edyta\bin\Debug\Edyta.exe

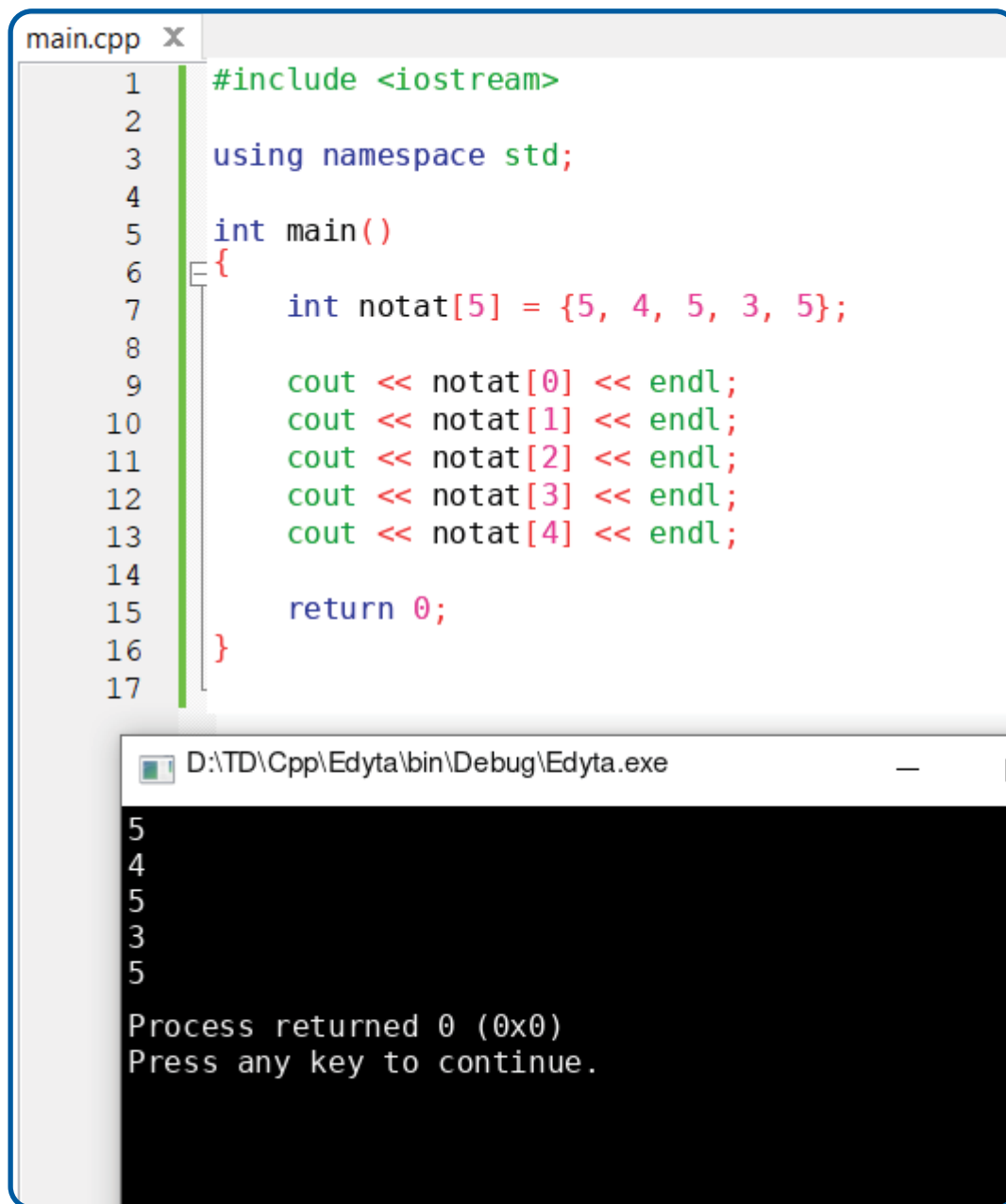
5
4
5
3
5

Process returned 0 (0x0)
Press any key to continue.

Në shembullin, për secilën notë është përdorur një ndryshore e veçantë. Zgjidhja është e pranueshme kur numri i të dhënave është i vogël, por bëhet jopraktike nëse duhet të përpunohen dhjetëra ose qindra vlera.

Kur disa të dhëna kanë të njëjtin tip dhe të njëjtin qëllim, ato mund të organizohen në një varg. Në vend që të përdoren shumë emra të ndryshëm për ndryshore, të gjitha vlerat grupohen nën një emër të përbashkët. Në këtë mënyrë, programi bëhet më i shkurtër dhe më i rregullt. Për më tepër, të dhënat mund të përpunohen lehtësisht duke përdorur cikle. Pikërisht kjo mundësi përbën një nga përparësitë më të mëdha të vargjeve dhe një nga arsyet kryesore të përdorimit të tyre të gjerë në programim.

Shembulli 2: Të njëjtat të dhëna të organizuara në varg



```
main.cpp x
1  #include <iostream>
2
3  using namespace std;
4
5  int main()
6  {
7      int notat[5] = {5, 4, 5, 3, 5};
8
9      cout << notat[0] << endl;
10     cout << notat[1] << endl;
11     cout << notat[2] << endl;
12     cout << notat[3] << endl;
13     cout << notat[4] << endl;
14
15     return 0;
16 }
17
```

D:\TD\Cpp\Edyta\bin\Debug\Edyta.exe

5
4
5
3
5

Process returned 0 (0x0)
Press any key to continue.

Në shembullin, të gjitha notat janë ruajtur në një varg me emrin notat. Në këtë mënyrë, disa vlera organizohen nën një emër të përbashkët, gjë që e bën programin më të thjeshtë dhe më të lehtë për mirëmbajtje.

Elementet e një vargu vendosen në lokacione të njëpasnjëshme të memories. Kjo do të thotë se kompjuteri i ruan ato njëri pas tjetrit në memorie. Falë kësaj mënyre organizimi, qasja te elementet është shumë e shpejtë dhe efikase. Megjithëse të gjithë elementet lidhen me një emër të përbashkët, secili prej tyre përfaqëson një vlerë të veçantë, e cila mund të lexohet ose të ndryshohet në mënyrë të pavarur. Kjo mënyrë e organizimit të të dhënave mundëson që një numër i madh vlerash të përpunohen si një tërësi e vetme. Pikërisht për këtë arsye, vargjet kanë një rol të rëndësishëm në programim dhe përbëjnë bazën për shumë struktura më të ndërlikuara të të dhënave. Gjatë punës me vargje duhet të kemi gjithmonë parasysh se të gjithë elementet duhet të jenë të të njëjtit tip. Nuk lejohet që në të njëjtin varg të ruhen njëkohësisht numra të plotë, numra realë dhe vlera tekstuale.

Shembulli 3: Numri i vizitorëve në muze

Gjatë pesë ditëve janë regjistruar numri i vizitorëve në një muze. Të dhënat duhet të ruhen në një varg dhe të shfaqen në ekran.

```
#include <iostream>

using namespace std;

int main()
{
    int vizitoret[5] = {120, 145, 132, 160, 151};

    cout << "Dita 1: " << vizitoret[0] << endl;
    cout << "Dita 2: " << vizitoret[1] << endl;
    cout << "Dita 3: " << vizitoret[2] << endl;
    cout << "Dita 4: " << vizitoret[3] << endl;
    cout << "Dita 5: " << vizitoret[4] << endl;

    return 0;
}
```

Në shembullin, të dhënat për të pesë ditët janë vendosur në një varg. Në vend që të përdoren pesë ndryshore të ndryshme, të gjitha vlerat organizohen nën një emër të përbashkët. Në këtë mënyrë, programi bëhet më i rregullt dhe më i lehtë për t'u mirëmbajtur.

Vargjet përdoren sa herë që duhet të përpunohet një numër i madh të dhënash të të njëjtit lloj. Ato mund të përdoren për ruajtjen e notave, temperaturave, të dhënave financiare, rezultateve sportive, matjeve nga sensorët dhe shumë llojeve të tjera të informacionit. Pa vargje, zhvillimi i programeve më të mëdha do të ishte dukshëm më i vështirë. Për këtë arsye, mësimi i vargjeve përbën një hap të rëndësishëm në zhvillimin e aftësive të programimit. Pasi të përvetësohet koncepti bazë, mund të kalohet në futjen, përpunimin dhe analizimin e elementeve të vargjeve. Pikërisht këto veprime përbëjnë bazën për një numër të madh algoritmesh dhe zgjidhjesh programuese që përdoren në praktikë.

PYETJE DHE DETYRA PËR PËRSËRITJE



PYETJE

1. Çfarë përfaqëson një varg njëdimensional?
2. Kur justifikohet përdorimi i një vargu në vend të disa ndryshoreve të veçanta?
3. Çfarë lloji të dhënash mund të përmbajë një varg?
4. Si organizohen elementet e një vargu në memorie?
5. Cilat janë përparësitë e përdorimit të vargjeve?



DETYRA

1. Në një bibliotekë duhet të ruhen numrat e librave të huazuar gjatë shtatë ditëve. Shkruaj një program që do t'i vendosë të dhënat në një varg dhe do t'i shfaqë në ekran.
2. Në një klub sportiv regjistrohet numri i pikëve të fituara në pesë ndeshje. Shkruaj një program që do t'i ruajë të dhënat në një varg dhe do t'i shfaqë ato.
3. Në një stacion meteorologjik regjistrohen temperaturat ditore për gjashtë ditë. Shkruaj një program që do t'i ruajë temperaturat në një varg dhe do t'i shfaqë ato.
4. Në një dyqan ndiqet shitja e katër produkteve. Shkruaj një program që do t'i ruajë sasinë në një varg dhe do t'i shfaqë në ekran.
5. Në një shkollë mbahet evidenca e numrit të nxënësve në pesë paralele. Shkruaj një program që do t'i ruajë të dhënat në një varg dhe do t'i shfaqë ato.



MBAJ MEND ÇFARË MËSOVE

- Vargu është një bashkësi elementesh të të njëjtit tip.
- Të gjithë elementet e vargut kanë një emër të përbashkët.
- Elementet vendosen në lokacione të njëpasnjëshme të memories.
- Vargjet mundësojnë ruajtjen e organizuar të një numri të madh të dhënash.
- Vargjet përbëjnë një nga strukturat më të rëndësishme të të dhënave në programim.



PËR ATA QË DËSHIROJNË TË DINË MË SHUMË

Në sistemet kompjuterike moderne, sasi të mëdha të të dhënave organizohen duke përdorur struktura që bazohen në vargje. Listat e përdoruesve në rrjetet sociale, rezultatet e kërkimeve në internet, të dhënat nga eksperimentet shkencore dhe analizat statistikore shpesh fillojnë pikërisht me vargje të thjeshta.

Edhe pse në pamje të parë duken si një strukturë e thjeshtë, vargjet përfaqësojnë një bazë mbi të cilën ndërtohen shumë struktura më të ndërlikuara, të cilat përdoren në programimin modern dhe në zhvillimin e sistemeve softuerike.



DEKLARIMI DHE INICIALIZIMI I VARGJEVE NJËDIMENSIONALE

TASHMË DI ...

- të shpjegosh se çfarë përfaqëson një varg njëdimensional;
- të dallosh situatat në të cilat nevojitet përdorimi i një vargu;
- të dallosh një ndryshore nga një varg;
- të shpjegosh se çfarë përfaqëson madhësia e një vargu;
- të dallosh se elementet e vargut numërohen duke filluar nga 0.

DETYRA PËR PËRSËRITJE

1. Çfarë përfaqëson një varg njëdimensional?
2. Pse është më mirë të përdoret një varg në vend të një numri të madh ndryshoresh të veçanta?
3. Çfarë tregon numri në kllapa gjatë deklarimit të një vargu?
4. Cili është pozicioni i elementit të parë në një varg?
5. Cili është pozicioni i elementit të fundit në një varg me madhësi 8?

NË PËRMBAJTJEN QË VIJON DO TË MËSOSH:

- të deklarojsh vargje njëdimensionale;
- të inicializosh vargjet gjatë deklarimit;
- të dallosh deklarimin nga inicializimi;
- të përdorësh mënyra të ndryshme për inicializimin e vargjeve;
- të dallosh gabimet më të zakonshme gjatë punës me vargje.

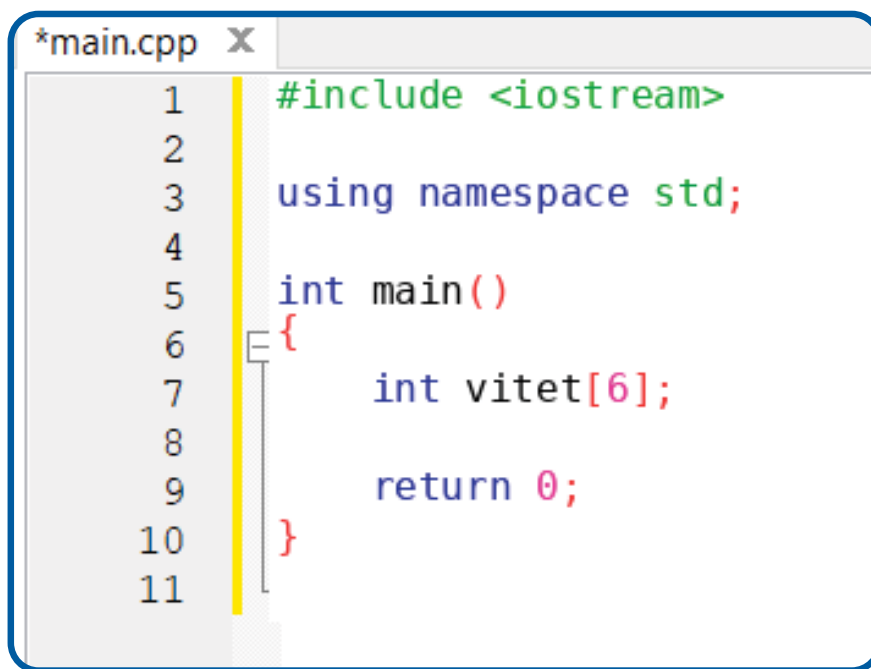
Para se të vendosen të dhënat në një varg, fillimisht duhet të krijohet hapësirë në memorien e kompjuterit. Kjo procedurë quhet deklarimi i vargut. Gjatë deklarimit përcaktohen tipi i të dhënave që do të ruhen dhe numri i elementeve që do të përmbajë vargu.

Pas deklarimit, elementeve të vargut mund t'u caktohen vlera fillestare. Kjo procedurë quhet inicializimi. Në gjuhën programuese C++ ekzistojnë disa mënyra për inicializimin e vargjeve dhe zgjedhja e mënyrës së duhur kontribuon në një paraqitje më të qartë dhe në punë më të thjeshtë me të dhënat.

Deklarimi i një vargu është procedura me të cilën rezervohet hapësirë në memorie për një numër të caktuar elementesh të të njëjtit tip. Gjatë deklarimit duhet të përcaktohen tipi i të dhënave, emri i vargut dhe madhësia e tij. Madhësia përcakton se sa elemente mund të përmbajë vargu.

Pasi të jetë deklaruar, vargu mund të përdoret për ruajtjen dhe përpunimin e të dhënave. Nëse gjatë deklarimit elementeve nuk u caktohen vlera, ato do të përmbajnë vlera të papërcaktuara. Për këtë arsye, në shumë raste rekomandohet që vargu të inicializohet menjëherë. Në këtë mënyrë, programi bëhet më i qartë dhe mundësia për shfaqjen e gabimeve zvogëlohet ndjeshëm.

Shembulli 1: Deklarimi i një vargu

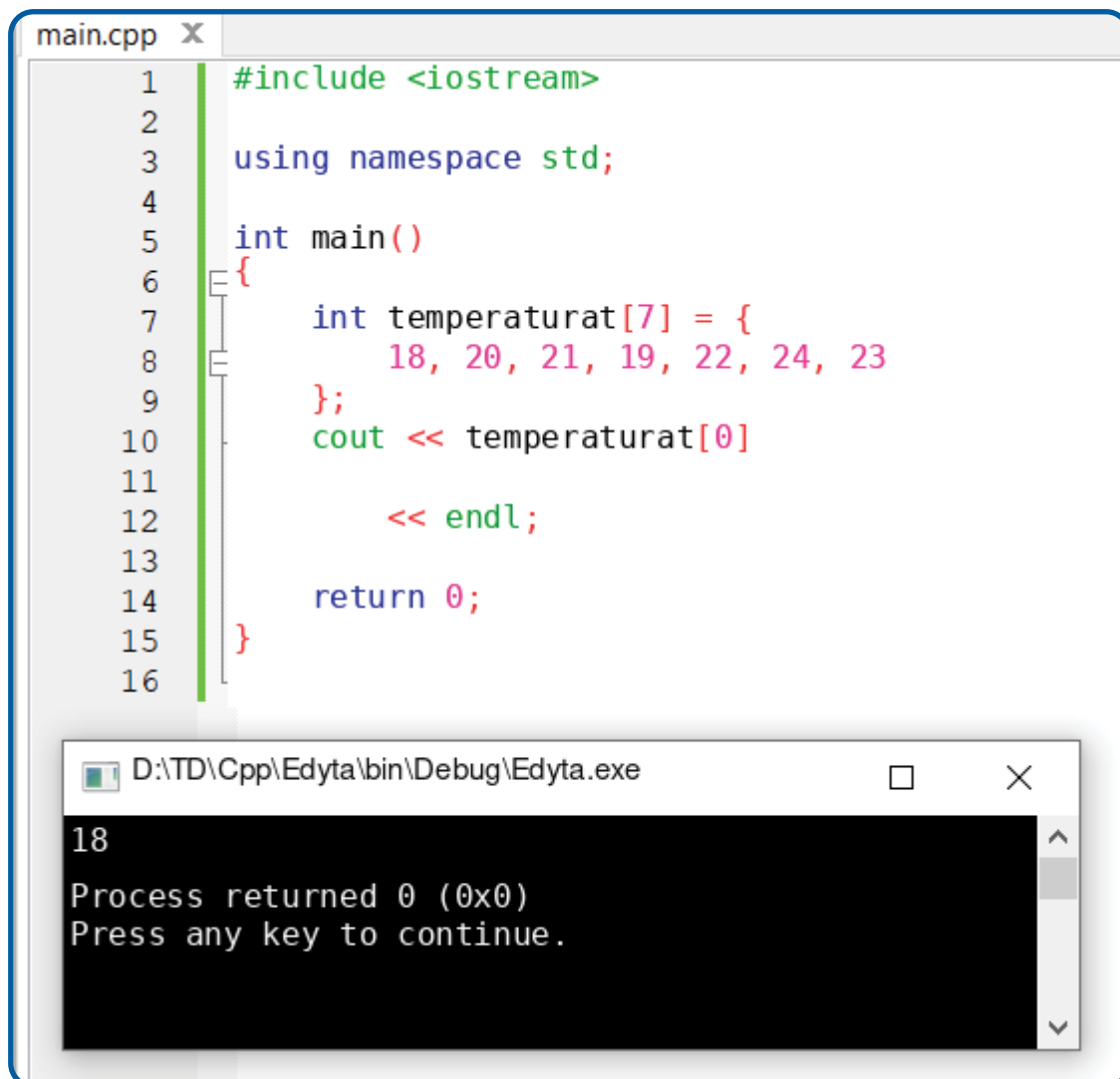


```
*main.cpp x
1  #include <iostream>
2
3  using namespace std;
4
5  int main()
6  {
7      int vitet[6];
8
9      return 0;
10 }
11
```

Në shembullin është deklaruar një varg me emrin godini. Në të mund të ruhen gjashtë numra të plotë. Në këtë moment është rezervuar hapësirë në memorie, por elementeve ende nuk u janë caktuar vlera.

Kur vlerat njihen paraprakisht, ato mund t'u caktohen elementeve që gjatë deklarimit të vargut. Kjo procedurë quhet inicializim. Gjatë inicializimit, vlerat shkruhen në kllapa kaçurrela dhe ndahen me presje. Numri i vlerave zakonisht është i barabartë me madhësinë e vargut. Kjo mënyrë e punës përdoret shpesh gjatë testimit të programeve ose kur të dhënat njihen paraprakisht. Vargjet e inicializuara i mundësojnë programit të disponojë menjëherë me të dhëna që mund të përpunohen dhe analizohen.

Shembulli 2: Inicializimi i një vargu



```
main.cpp x
1  #include <iostream>
2
3  using namespace std;
4
5  int main()
6  {
7      int temperaturat[7] = {
8          18, 20, 21, 19, 22, 24, 23
9      };
10     cout << temperaturat[0]
11
12         << endl;
13
14     return 0;
15 }
16
```

D:\TD\Cpp\Edyta\bin\Debug\Edyta.exe

18

Process returned 0 (0x0)

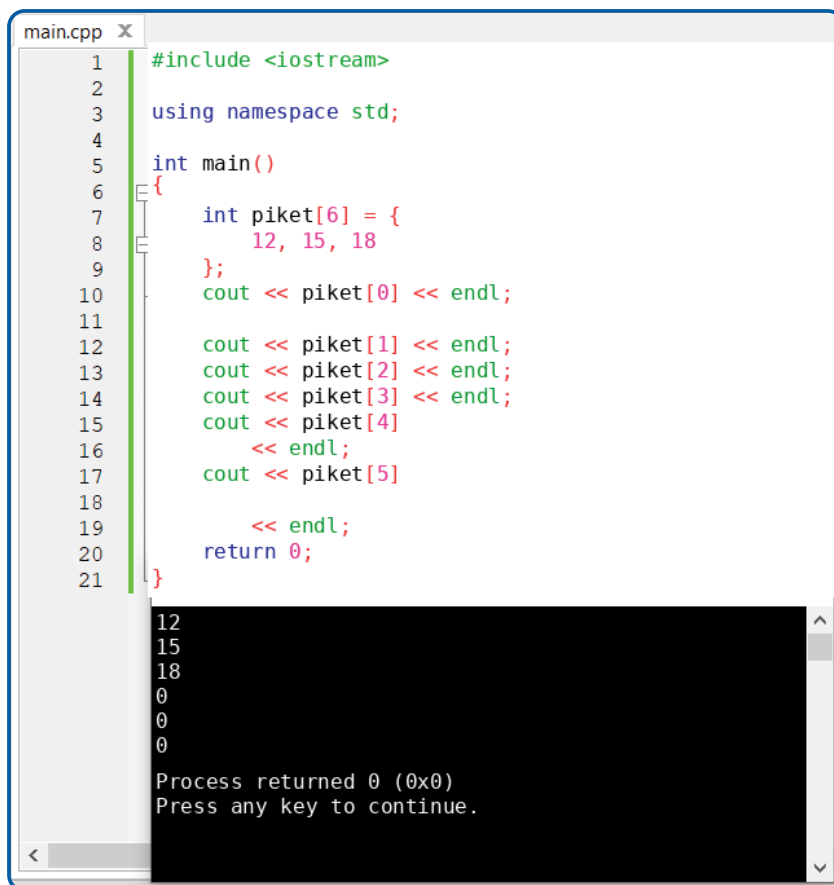
Press any key to continue.

Në shembullin, vargu është inicializuar që gjatë deklarimit. Secilit element i është caktuar një vlerë fillestare dhe vargu është menjëherë i gatshëm për përdorim.

Gjatë inicializimit të vargjeve nuk është e detyrueshme të caktohen vlera për të gjithë elementet. Nëse jepen më pak vlera sesa madhësia e vargut, elementet e para do të marrin vlerat e caktuara, ndërsa elementeve të mbetura do t'u caktohet automatikisht vlera zero.

Kjo mënyrë e inicializimit quhet inicializim i pjesshëm. Kjo qasje mund të jetë e dobishme kur vetëm një pjesë e të dhënave dihet në momentin e krijimit të vargut. Megjithatë, gjatë punës me vargje të tilla duhet të kontrollohet me kujdes se cilat elemente përmbajnë të dhëna të përcaktuara dhe cilat kanë marrë automatikisht vlerën zero.

Njohja e mënyrës së inicializimit është e rëndësishme, sepse ajo ndikon drejtpërdrejt në rezultatet që do të përftohen pas ekzekutimit të programit.



```

main.cpp x
1  #include <iostream>
2
3  using namespace std;
4
5  int main()
6  {
7      int piket[6] = {
8          12, 15, 18
9      };
10     cout << piket[0] << endl;
11
12     cout << piket[1] << endl;
13     cout << piket[2] << endl;
14     cout << piket[3] << endl;
15     cout << piket[4]
16         << endl;
17     cout << piket[5]
18         << endl;
19     return 0;
20 }
21
12
15
18
0
0
0
Process returned 0 (0x0)
Press any key to continue.

```

Në shembull, tre elementet e para marrin vlerat 12, 15 dhe 18. Tre elementet e mbetura automatikisht marrin vlerën 0. Falë kësaj mundësie, nuk është e nevojshme që gjithmonë të përcaktohen vlera për të gjithë elementet e vargut.

Në gjuhën programuese C++ ekziston mundësia që madhësia e vargut të përcaktohet automatikisht në bazë të numrit të vlerave të dhëna. Në raste të tilla, numri i elementeve nuk shënohet brenda kllapave katrore. Kompilatori e përcakton vetë numrin e elementeve, bazuar në numrin e vlerave të dhëna. Kjo qasje e bën programin më të lexueshëm dhe zvogëlon mundësinë e gabimeve gjatë numërimit të elementeve.

Megjithatë, kjo mënyrë pune është e përshtatshme vetëm kur të gjitha vlerat njihen që gjatë deklarimit të vargut. Nëse të dhënat do të futen më vonë, madhësia e vargut duhet të përcaktohet paraprakisht. Prandaj, zgjedhja e mënyrës së deklarimit varet nga problemi konkret që zgjidhet dhe nga disponueshmëria e të dhënave në momentin e krijimit të vargut.

Përcaktimi automatik i madhësisë

```

int muajt[] =
{
    31, 28, 31, 30,
    31, 30, 31, 31,
    30, 31, 30, 31
};

```

Në këtë shembull nuk është përcaktuar madhësia e vargut. Meqenëse janë dhënë dymbëdhjetë vlera, kompajleri automatikisht krijon një varg me dymbëdhjetë elemente.

PYETJE DHE DETYRA PËR PËRSËRITJE



PYETJE

1. Çfarë përfaqëson deklarimi i një vargu?
2. Çfarë përfaqëson inicializimi i një vargu?
3. Cili është dallimi midis deklarimit dhe inicializimit të një vargu?
4. Çfarë ndodh me elementet të cilave nuk u është caktuar vlerë gjatë inicializimit të pjesshëm?
5. Në cilat situata mund të përdoret përcaktimi automatik i madhësisë së vargut?



DETYRA

1. Shkruaj një program në të cilin do të deklarohet një varg me dhjetë numra të plotë, pa inicializim.
2. Shkruaj një program në të cilin do të inicializohet një varg me shtatë temperatura të matura gjatë një jave.
3. Shkruaj një program në të cilin do të krijohet një varg me pesë vlera që përfaqësojnë rezultatet e një sportisti në pesë gara.
4. Shkruaj një program në të cilin do të përdoret një varg i inicializuar pjesërisht me tetë elemente, ku do të përcaktohen vetëm katër vlerat e para.
5. Shkruaj një program në të cilin madhësia e vargut do të përcaktohet automatikisht në bazë të numrit të vlerave të dhëna.



MBAJ MEND ÇFARË MËSOVE

- Deklarimi rezervon hapësirë në memorie për elementet e vargut.
- Inicializimi u cakton vlera fillestare elementeve.
- Madhësia e vargut përcakton numrin e elementeve që mund të ruhen.
- Gjatë inicializimit të pjesshëm, elementet e mbetura marrin vlerën 0.
- Madhësia e vargut mund të përcaktohet automatikisht kur të gjitha vlerat jepen gjatë deklarimit.



PËR ATA QË DËSHIROJNË TË DINË MË SHUMË

Në gjuhët moderne të programimit ekzistojnë struktura që mund ta ndryshojnë automatikisht madhësinë e tyre gjatë ekzekutimit të programit. Struktura të tilla mundësojnë shtimin dhe heqjen e elementeve pa qenë e nevojshme që numri i tyre të përcaktohet paraprakisht. Megjithatë, vargjet klasike mbeten një nga strukturat më të shpejta dhe më efikase për ruajtjen e të dhënave. Për shkak të thjeshtësisë dhe efikasitetit të tyre, ato përbëjnë bazën për një numër të madh strukturash më të ndërlikuara që përdoren në programimin modern.



2.16

INDEKSI I ELEMENTIT DHE QASJA NË ELEMENTET E VARGUT

TASHMË DI ...

- të shpjegosh se çfarë përfaqëson një varg njëdimensional;
- të deklarosh një varg;
- të inicializosh një varg;
- të dallosh deklarin nga inicializimi;
- të kuptosh se elementet e vargut indeksohen duke filluar nga 0.

DETYRA PËR PËRSËRITJE

1. Çfarë përfaqëson deklarimi i një vargu?
2. Çfarë përfaqëson inicializimi i një vargu?
3. Sa elemente përmban vargu `int numrat[8];`?
4. Cili është pozicioni i elementit të fundit në një varg me madhësi 10?
5. Çfarë ndodh me elementet e mbetura gjatë inicializimit të pjesshëm?

NË PËRMBAJTJEN QË VIJON DO TË MËSOSH:

- të shpjegosh se çfarë përfaqëson indeksi i një elementi;
- të qasësh te elementet individuale të një vargu;
- të ndryshosh vlerat e elementeve në një varg;
- të dallosh indeksat e pavlefshme;
- të shmangësh gabimet më të zakonshme gjatë punës me indeksat.

Edhe pse të gjithë elementet e një vargu lidhen me një emër të përbashkët, secili element duhet të mund të aksesohet dhe përpunohet veçmas. Për këtë qëllim përdoren indekset. Indeksi përcakton pozicionin e elementit në varg dhe i mundëson programit të aksesojë një vlerë të caktuar.

Përdorimi i saktë i indekseve përbën bazën për një punë të suksesshme me vargjet. Shumë gabime programimi shfaqen pikërisht për shkak të përdorimit të gabuar të indekseve. Prandaj, është e nevojshme të kuptohet me kujdes se si funksionon numërimi i elementeve dhe cilat vlera përfaqësojnë indekse të lejuara.

Çdo element në varg ka pozicionin e vet, i cili quhet indeks. Indeksi mundëson përcaktimin e elementit konkret me të cilin do të punohet. Në gjuhën programuese C++, numërimi fillon nga 0. Për këtë arsye, elementi i parë ka indeksin 0, i dyti indeksin 1, i treti indeksin 2, e kështu me radhë. Nëse vargu përmban pesë elemente, elementi i fundit nuk ka indeksin 5, por indeksin 4. Ky rregull vlen për të gjithë vargjet, pavarësisht nga madhësia e tyre. Aksesimi i elementeve bëhet duke shënuar emrin e vargut dhe indeksin brenda kllapave katrore. Në këtë mënyrë, programi e di saktësisht se me cilin element duhet të punojë.

Shembulli 1: Aksesimi i një elementi të vargut

```
#include <iostream>

using namespace std;

int main()
{
    int piket[5] =
    {
        10, 15, 20, 25, 30
    };

    cout << piket[2] << endl;

    return 0;
}
```

Në shembullin paraqitet elementi me indeksin 2. Meqenëse numërimi fillon nga 0, ai është elementi i tretë në varg dhe vlera e tij është 20.

Përveç leximit të vlerave, indekset mundësojnë edhe ndryshimin e elementeve në varg. Për të ndryshuar vlerën e një elementi të caktuar, duhet të specifikohet indeksi i tij dhe t'i caktohet një vlerë e re. Në këtë mënyrë, elementet individuale mund të ndryshohen në mënyrë të pavarur, pa ndikuar te elementet e tjera të vargut.

Kjo mundësi është veçanërisht e rëndësishme gjatë përpunimit të të dhënave, sepse mundëson përditësimin e informacioneve të caktuara pa pasur nevojë të krijohet përsëri i gjithë vargu.

Shembulli 2: Ndryshimi i një vlere në varg

```
#include <iostream>

using namespace std;

int main()
{
    int stoqet[4] =
    {
        12, 18, 25, 30
    };

    stoqet[1] = 22;

    cout << stoqet[1] << endl;

    return 0;
}
```

Në shembullin, vlera e elementit me indeksin 1 ndryshon nga 18 në 22.

Gjatë punës me vargje, është veçanërisht e rëndësishme të përdoren indekse të lejuara. Çdo element mund të aksesohet vetëm përmes një indeksi që i përket vargut. Nëse vargu përmban pesë elemente, indekset e lejuara janë 0, 1, 2, 3 dhe 4. Përpjekja për të aksesuar një element me indeksin 5 ose më të madh përbën gabim.

Gjithashtu, nuk lejohet përdorimi i indekseve negative. Gabime të tilla mund të shkaktojnë funksionim të parregullt të programit ose shfaqjen e rezultateve të papritura.

Prandaj, gjithmonë duhet të kontrollohet me kujdes nëse indeksi i përdorur ndodhet brenda diapazonit të lejuar. Kuptimi i këtij rregulli është një nga parakushtet më të rëndësishme për punë të suksesshme me vargje.

Shembulli 3: Aksesimi i disa elementeve

Në një klub sportiv janë regjistruar pikët e fituara në pesë ndeshje. Të shfaqen rezultati i parë, i tretë dhe i fundit.

```
#include <iostream>

using namespace std;

int main()
{
    int piket[5] =
    {
        12, 18, 15, 20, 25
    };

    cout << "Rezultati i parë: "
         << piket[0]
         << endl;
```

```
cout << "Rezultati i tretë: "
    << piket[2]
    << endl;

cout << "Rezultati i fundit: "
    << piket[4]
    << endl;

return 0;
}
```

Në shembullin aksesohen tre elemente të ndryshme nga i njëjti varg. Elementi i parë ka indeksin 0, elementi i tretë indeksin 2, ndërsa elementi i fundit indeksin 4. Kjo tregon se pozicioni i elementit nuk është gjithmonë i barabartë me numrin e tij rendor në numërimin e përditshëm.

Indekset përfaqësojnë një mekanizëm themelor përmes të cilit aksesohet te elementet e vargjeve. Pa to, nuk do të ishte e mundur që vlerat individuale të lexoheshin, të ndryshoheshin ose të përpunoheshin.

Prandaj, gjatë punës me vargje duhet të dihen gjithmonë madhësia e vargut dhe diapazoni i lejuar i indekseve. Pasi të përvetësohet aksesimi i elementeve individuale, mund të kalohet në futjen dhe përpunimin e një numri më të madh të të dhënave, që përbën hapin e radhës në studimin e vargjeve njëdimensionale.

PYETJE DHE DETYRA PËR PËRSËRITJE



PYETJE

1. Çfarë përfaqëson indeksi i një elementi?
2. Nga cili numër fillon numërimi i elementeve në një varg?
3. Cili është indeksi i elementit të fundit në një varg me madhësi 12?
4. Si aksesohet një element i vargut?
5. Pse është e rëndësishme të përdoren vetëm indekse të lejuara?



DETYRA

1. Në një bibliotekë janë regjistruar vizitat gjatë pesë ditëve, të realizuara nga nxënës me numrat e kartelave: 35, 42, 38, 50 dhe 47. Shkruaj një program që do të shfaqë elementin e parë dhe të fundit të vargut.
2. Në një dyqan ndiqet shitja e gjashtë produkteve. Shkruaj një program që do t'i ruajë të dhënat në një varg dhe do të shfaqë elementin e katërt.
3. Në një turne sportiv janë regjistruar pikët e fituara nga një ekip në pesë ndeshje. Shkruaj një program që do të ndryshojë vlerën e ndeshjes së dytë dhe do të shfaqë vlerën e re.
4. Në një stacion meteorologjik janë ruajtur shtatë temperatura në një varg. Shkruaj një program që do të shfaqë temperaturën e ditës së parë, të tretë dhe të gjashtë.
5. Në evidencën e një shkolle ruhet numri i nxënësve në pesë paralele. Shkruaj një program që do të ndryshojë vlerën e elementit të fundit dhe do të shfaqë rezultatin.

MBAJ MEND ÇFARË MËSOVE



- Indeksi përcakton pozicionin e elementit në varg.
- Numërimi i elementeve fillon nga 0.
- Elementi i fundit ka indeks të barabartë me madhësinë e vargut minus 1.
- Indekset përdoren për leximin dhe ndryshimin e elementeve.
- Duhet të përdoren vetëm indekse që i përkasin vargut.

PËR ATA QË DËSHIROJNË TË DINË MË SHUMË



Në shumë gjuhë programuese, si C++, Java dhe C#, numërimi i elementeve në vargje fillon nga 0. Ky rregull rrjedh nga mënyra se si kompjuteri i llogarit adresat e memories së elementeve. Edhe pse në fillim mund të duket e pazakontë, kjo mënyrë e numërimit mundëson llogaritje më të thjeshta dhe më të shpejta gjatë qasjes në të dhëna. Pikërisht për këtë arsye, indeksimi nga 0 është pranuar si standard në një numër të madh gjuhësh moderne programuese.



FUTJA DHE SHTYPJA E ELEMENTEVE TË VARGUT

TASHMË E DI...

- të deklarojsh një varg njëdimensional;
- të inicializojsh një varg;
- të qasësh te elementet përmes indeksit;
- të ndryshojsh vlerat e elementeve në varg;
- të përdorësh ciklin for.

DETYRA PËR PËRSËRITJE

1. Çfarë përfaqëson indeksi i një elementi?
2. Nga cili numër fillon numërimi i elementeve në një varg?
3. Cili është indeksi i elementit të fundit në një varg me madhësi 15?
4. Si ndryshohet vlera e një elementi në një varg?
5. Pse nuk duhet të përdoren indekse jashtë intervalit të lejuar?

NË PËRMBAJTJET QË VIJOJNË DO TË MËSOSH:

- të futësh vlera në varg;
- të përdorësh cikël për futjen e elementeve;
- të shfaqësh elementet e vargut;
- të kombinosh vargjet dhe ciklet;
- të shmangësh gabimet më të zakonshme gjatë futjes së të dhënave.

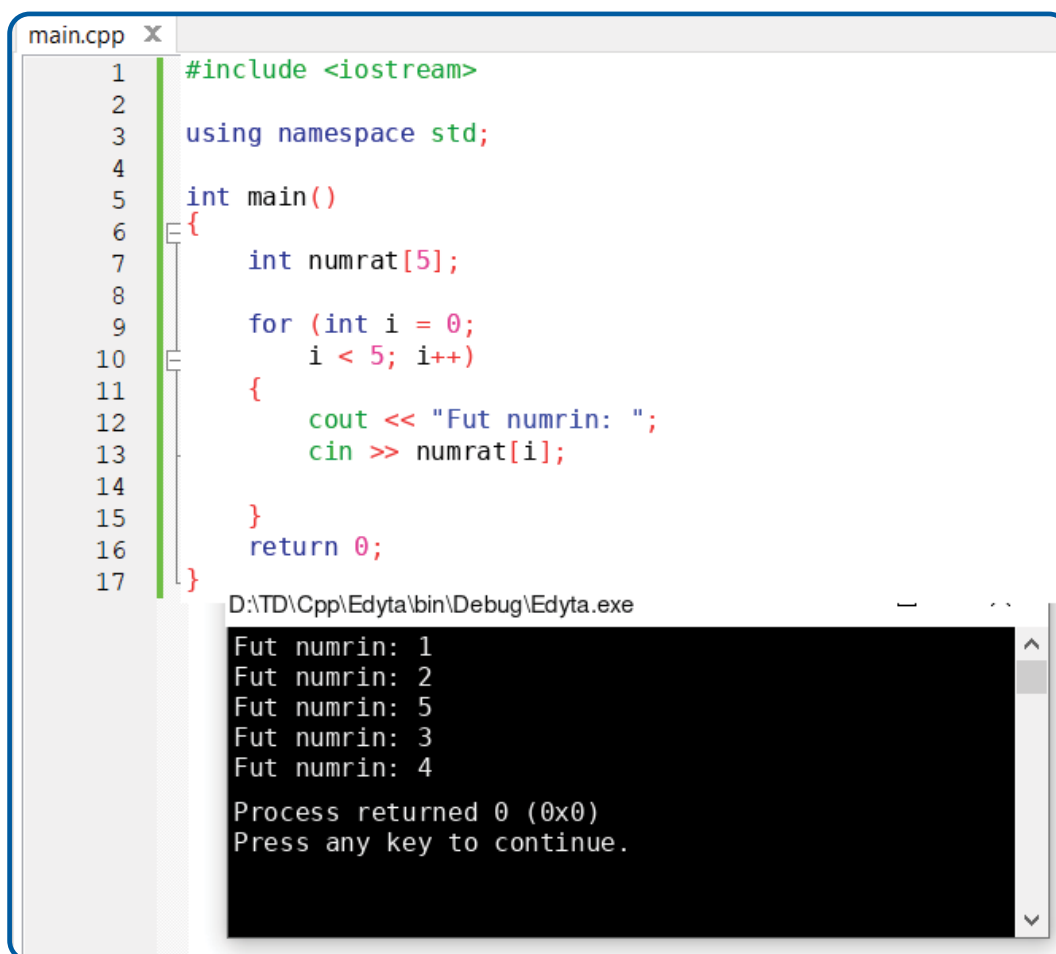
Një nga përparësitë më të rëndësishme të vargjeve është mundësia e përpunimit të një numri të madh të dhënash me një numër të vogël udhëzimesh programuese. Nëse është e nevojshme të futen dhjetë, njëzet

ose njëqind vlera, nuk është praktike që për secilën vlerë të përdoret një ndryshore e veçantë. Në vend të kësaj, të dhënat mund të ruhen në një varg.

Gjatë punës me vargje, shpesh është e nevojshme që të dhënat të futen nga përdoruesi. Për këtë qëllim përdoren ciklet, të cilat mundësojnë futjen e disa elementeve duke përsëritur të njëjtat udhëzime. Në mënyrë të ngjashme mund të realizohet edhe shfaqja e përmbajtjes së vargut.

Gjatë futjes së të dhënave në një varg, çdo element duhet të marrë një vlerë. Në vend që të shkruhet një urdhër i veçantë për çdo element, më së shpeshti përdoret cikli for. Numëruesi i ciklit përfaqëson pozicionin e elementit që po përpunohet në atë moment. Në këtë mënyrë mund të realizohet lehtësisht futja e të gjithë elementeve, pavarësisht nëse vargu përmban pesë, dhjetë apo njëqind vlera. Kombinimi i vargjeve dhe cikleve përbën një nga teknikat më të rëndësishme në programim, sepse mundëson përpunim efikas të një numri të madh të të dhënave.

Shembulli 1: Futja e pesë numrave në një varg



```
main.cpp x
1  #include <iostream>
2
3  using namespace std;
4
5  int main()
6  {
7      int numrat[5];
8
9      for (int i = 0;
10         i < 5; i++)
11      {
12          cout << "Fut numrin: ";
13          cin >> numrat[i];
14      }
15
16      return 0;
17 }
```

D:\TD\Cpp\Edyta\bin\Debug\Edyta.exe

Fut numrin: 1
Fut numrin: 2
Fut numrin: 5
Fut numrin: 3
Fut numrin: 4

Process returned 0 (0x0)
Press any key to continue.

Në shembull, cikli ekzekutohet pesë herë. Gjatë çdo ekzekutimi futet një vlerë në një element të ndryshëm të vargut.

Pasi të jenë futur të dhënat, shpesh është e nevojshme paraqitja e tyre. Edhe në këtë situatë përdoret cikli for. Në vend që të shkruhet një urdhër i veçantë për çdo element, cikli qaset në mënyrë të njëpasnjëshme të të gjitha pozicionet e vargut. Kjo qasje e bën programin më të shkurtër dhe më të kuptueshëm. Për më tepër, i njëjti kod mund të zbatohet edhe për vargje shumë më të mëdha, pa ndryshime të konsiderueshme.

Shembulli 2: Inicializimi i një vargu

The image shows a code editor window titled 'main.cpp' with the following C++ code:

```

1  #include <iostream>
2
3  using namespace std;
4
5  int main()
6  {
7      int numrat[5] = {
8          10, 20, 30, 40, 50
9      };
10     for (int i = 0; i < 5; i++)
11     {
12         cout << numrat[i]
13             << " ";
14     }
15
16     return 0;
17 }
18
19

```

Below the code editor is a console window titled 'D:\TD\Cpp\Edyta\bin\Debug\Edyta.exe'. It displays the output of the program:

```

10 20 30 40 50
Process returned 0 (0x0)
Press any key to continue.

```

Në shembull, të gjithë elementet e vargut shfaqen me ndihmën e një cikli. Pa cikël do të nevojiteshin pesë urdhra të veçantë cout.

Në zgjidhjet praktike programuese, zakonisht nuk mjafton vetëm futja e të dhënave. Pas futjes, zakonisht vijon përpunimi, kontrolli ose shfaqja e tyre. Për këtë arsye, shumë shpesh në një program përdoren dy cikle. Cikli i parë shërben për futjen e të dhënave në varg, ndërsa i dyti për shfaqjen e tyre. Kjo qasje mundëson që të gjitha të dhënat fillimisht të ruhen në memorie dhe më pas të përpunohen në mënyra të ndryshme. Me ritjen e numrit të elementeve, përparësitë e kësaj mënyre organizimi bëhen edhe më të theksuara. Në vend që të shkruhen dhjetëra ose qindra urdhra për futjen dhe shfaqjen e të dhënave, nevojiten vetëm dy cikle të shkurtra. Pikërisht për këtë arsye, kombinimi i vargjeve dhe cikleve përbën një nga teknikat më të përdorura në programim.

Shembulli 3: Futja dhe shfaqja e numrit të vizitorëve

Në një muze regjistrohet numri i vizitorëve gjatë pesë ditëve. Të futen të dhënat dhe më pas të shfaqen.

```
main.cpp x
1  #include <iostream>
2
3  using namespace std;
4
5  int main()
6  {
7      int vizitoret[5];
8      for (int i = 0; i < 5; i++)
9      {
10         cout << "Numri ditor i vizitorëve "
11             << i + 1
12             << ": ";
13
14         cin >> vizitoret[i];
15     }
16     cout << "Të dhënat e futura:"
17         << endl;
18     for (int i = 0;
19         i < 5; i++)
20     {
21         cout << "Dita "
22             << i + 1
23             << ": " << vizitoret[i]
24             << endl;
25     }
26     return 0;
27 }
28
```

```
Numri ditor i vizitorëve 1: 21
Numri ditor i vizitorëve 2: 45
Numri ditor i vizitorëve 3: 8
Numri ditor i vizitorëve 4: 105
Numri ditor i vizitorëve 5: 55
Të dhënat e futura:
Dita 1: 21
Dita 2: 45
Dita 3: 8
Dita 4: 105
Dita 5: 55
Process returned 0 (0x0)
Press any key to continue.
```

Në shembullin, cikli i parë i fut të dhënat në varg, ndërsa cikli i dytë i shfaq ato. Në këtë mënyrë, të gjitha vlerat mund të kontrollohen lehtësisht pas përfundimit të futjes.

Gjatë punës me vargje dhe cikle, është veçanërisht e rëndësishme që numri i përsëritjeve të përcaktohet saktë. Nëse cikli ekzekutohet më shumë herë se numri i elementeve në varg, do të ndodhë një përpjekje për t'iu qasur elementeve që nuk ekzistojnë. Nga ana tjetër, nëse cikli ekzekutohet më pak herë, një pjesë e elementeve nuk do të përpunohet. Për këtë arsye, kufiri i sipërm i ciklit duhet të përputhet me madhësinë e vargut. Ky rregull do të zbatohet më tej për të kryer llogaritje dhe analiza të ndryshme mbi elementet e vargjeve.

PYETJE DHE DETYRA PËR PËRSËRITJE



PYETJE

1. Pse zakonisht përdoret cikli gjatë futjes së elementeve në varg?
2. Cili është roli i numëruesit në cikël?
3. Pse është e rëndësishme që numri i përsëritjeve të përputhet me madhësinë e vargut?
4. Cilat janë përparësitë e përdorimit të ciklit për shfaqjen e elementeve?
5. Pse vargjet dhe ciklet përdoren shpesh së bashku?



DETYRA

1. Në një lokalitet arkeologjik janë gjetur shtatë objekte me vjetërsi të ndryshme, të shprehura në vite. Shkruaj një program që do t'i fusë vjetërsitë në një varg dhe më pas do t'i shfaqë sipas radhës së futjes.
2. Në një observator astronomik regjistrohet numri i meteorëve të vërejtur gjatë dhjetë netëve. Shkruaj një program që do t'i fusë të dhënat në një varg dhe më pas do t'i shfaqë të gjitha vlerat.
3. Në një fabrikë biçikletash evidentohet numri i biçikletave të prodhuara gjatë gjashtë ndërrimeve të punës. Shkruaj një program që do t'i fusë të dhënat në një varg dhe do t'i shfaqë në ekran.
4. Në një park kombëtar ndiqet numri i vizitorëve në pesë shtigje të ndryshme turistike. Shkruaj një program që do t'i fusë të dhënat në një varg dhe më pas do t'i shfaqë me një njoftim për secilin shteg.
5. Në një stacion hapësinor maten sasisë e ujit (në litra) të shpenzuara gjatë tetë ditëve të njëpasnjëshme. Shkruaj një program që do t'i fusë vlerat në një varg dhe do t'i shfaqë pas përfundimit të futjes.



MBAJ MEND ÇFARË MËSOVE

- Futja e elementeve në varg zakonisht bëhet me ciklin for.
- Numëruesi i ciklit përdoret si indeks i elementeve.
- Shfaqja e elementeve gjithashtu mund të realizohet me cikël.
- Madhësia e vargut duhet të përputhet me numrin e përsëritjeve të ciklit.
- Kombinimi i vargjeve dhe cikleve mundëson përpunimin efikas të një numri të madh të të dhënave.



PËR ATA QË DËSHIROJNË TË DINË MË SHUMË

Në një numër të madh aplikacionesh moderne, të dhënat vijnë nga burime të ndryshme, si sensorë, shërbime interneti, baza të të dhënave ose pajisje mobile. Pavarësisht nga origjina e tyre, ato zakonisht ruhen në struktura që mundësojnë përpunimin e disa vlerave njëkohësisht. Vargjet përfaqësojnë një nga strukturat më të thjeshta të këtij lloji. Falë cikleve, mijëra të dhëna mund të përpunohen me një numër relativisht të vogël udhëzimesh programuese. Kjo mundësi përbën bazën për zhvillimin e sistemeve për analizë statistikore, kërkime shkencore, përpunim të imazheve, mësim makinerik dhe shumë fusha të tjera moderne të informatikës.



TASHMË DI ...

- ## DETYRA PËR PËRSËRITJE

1. Çfarë përfaqëson indeksi i një elementi?
2. Nga cili numër fillon numërimi i elementeve?
3. Pse përdoret cikli for gjatë punës me vargje?
4. Si futen disa elemente në një varg?
5. Çfarë do të ndodhë nëse përdoret një indeks jashtë kufijve të vargut?

NË PËRMBAJTJET NË VIJIM DO TË MËSOSH:

- të llogaritësh shumën e elementeve të një vargu;
- të llogaritësh prodhimin e elementeve të një vargu;
- të përdorësh ndryshore ndihmëse gjatë përpunimit të të dhënave;
- të kombinosh veprimet aritmetike me vargjet;
- të zgjidhësh detyra praktike me përpunimin e të gjithë elementeve.

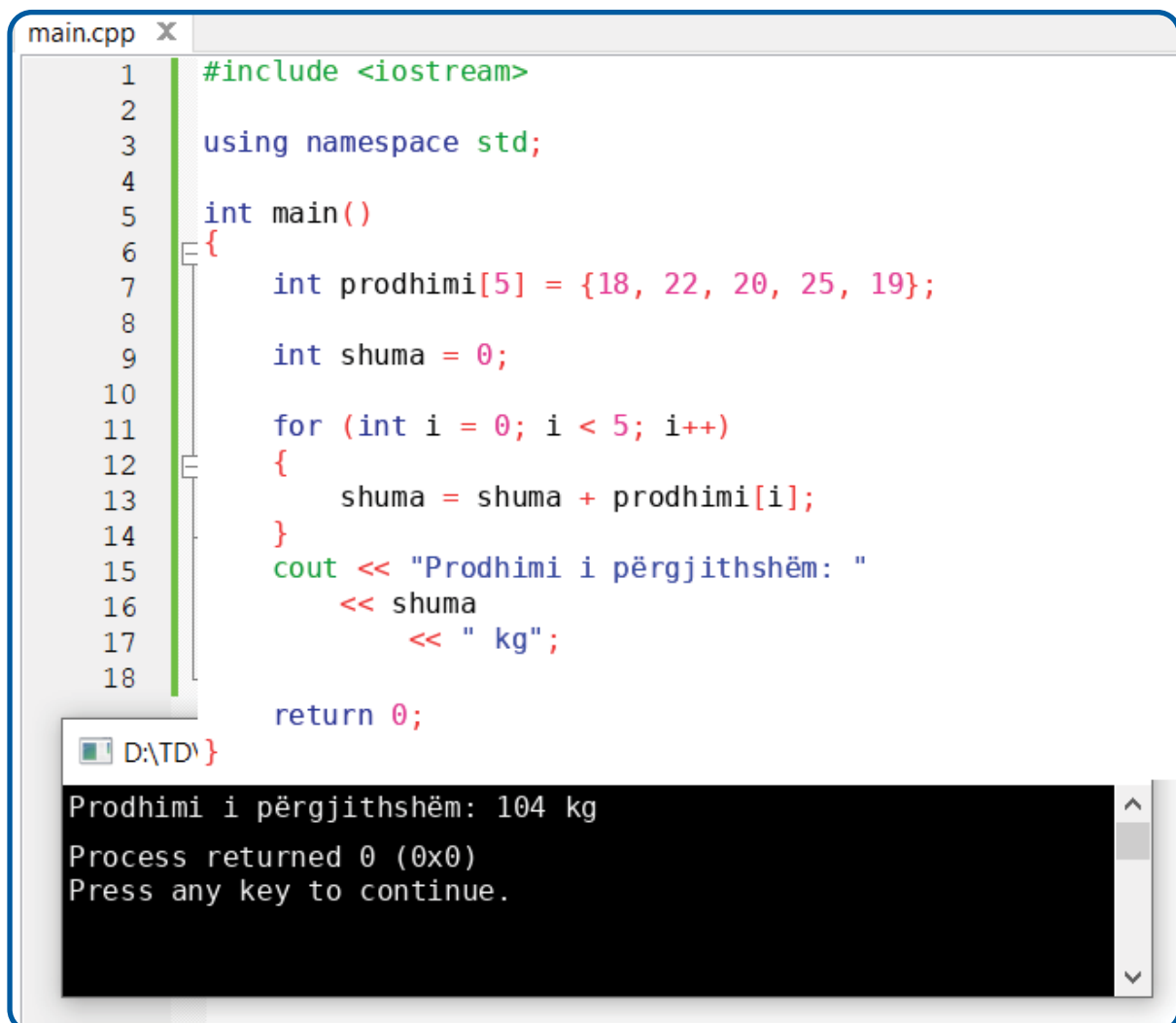
Për zgjidhjen e problemeve të tilla zakonisht përdoret një cikël që përpunon në mënyrë të njëpasnjëshme të gjithë elementet e vargut. Në çdo përsëritje të ciklit përditësohet rezultati i llogaritjes.

Në këtë mënyrë, një numër i madh të dhënash mund të përpunohen lehtësisht me një numër relativisht të vogël udhëzimesh programuese.

Gjatë përpunimit të elementeve të një vargu, shpesh është e nevojshme të llogaritet vlera e tyre totale. Për këtë qëllim përdoret një ndryshore ndihmëse, e cila në fillim merr vlerën 0. Më pas, gjatë çdo përsëritjeje të ciklit, elementit aktual të vargut i shtohet kjo ndryshore. Pas përpunimit të elementit të fundit, ndryshorja përmban shumën totale të të gjitha vlerave. Kjo mënyrë pune zbatohet në një numër të madh situatash praktike dhe përbën një nga teknikat themelore për përpunimin e të dhënave në vargje.

Shembulli 1: Prodhimi i mjaltit

Në një bletari janë matur sasi të mjaltit të prodhuara nga pesë koshere. Të llogaritet prodhimi i përgjithshëm.



The image shows a code editor window titled 'main.cpp' with a line number margin on the left. The code is as follows:

```

1  #include <iostream>
2
3  using namespace std;
4
5  int main()
6  {
7      int prodhimi[5] = {18, 22, 20, 25, 19};
8
9      int shuma = 0;
10
11     for (int i = 0; i < 5; i++)
12     {
13         shuma = shuma + prodhimi[i];
14     }
15     cout << "Prodhimi i përgjithshëm: "
16          << shuma
17          << " kg";
18
19     return 0;

```

Below the code editor, a terminal window shows the output of the program:

```

D:\TD\ }
Prodhimi i përgjithshëm: 104 kg
Process returned 0 (0x0)
Press any key to continue.

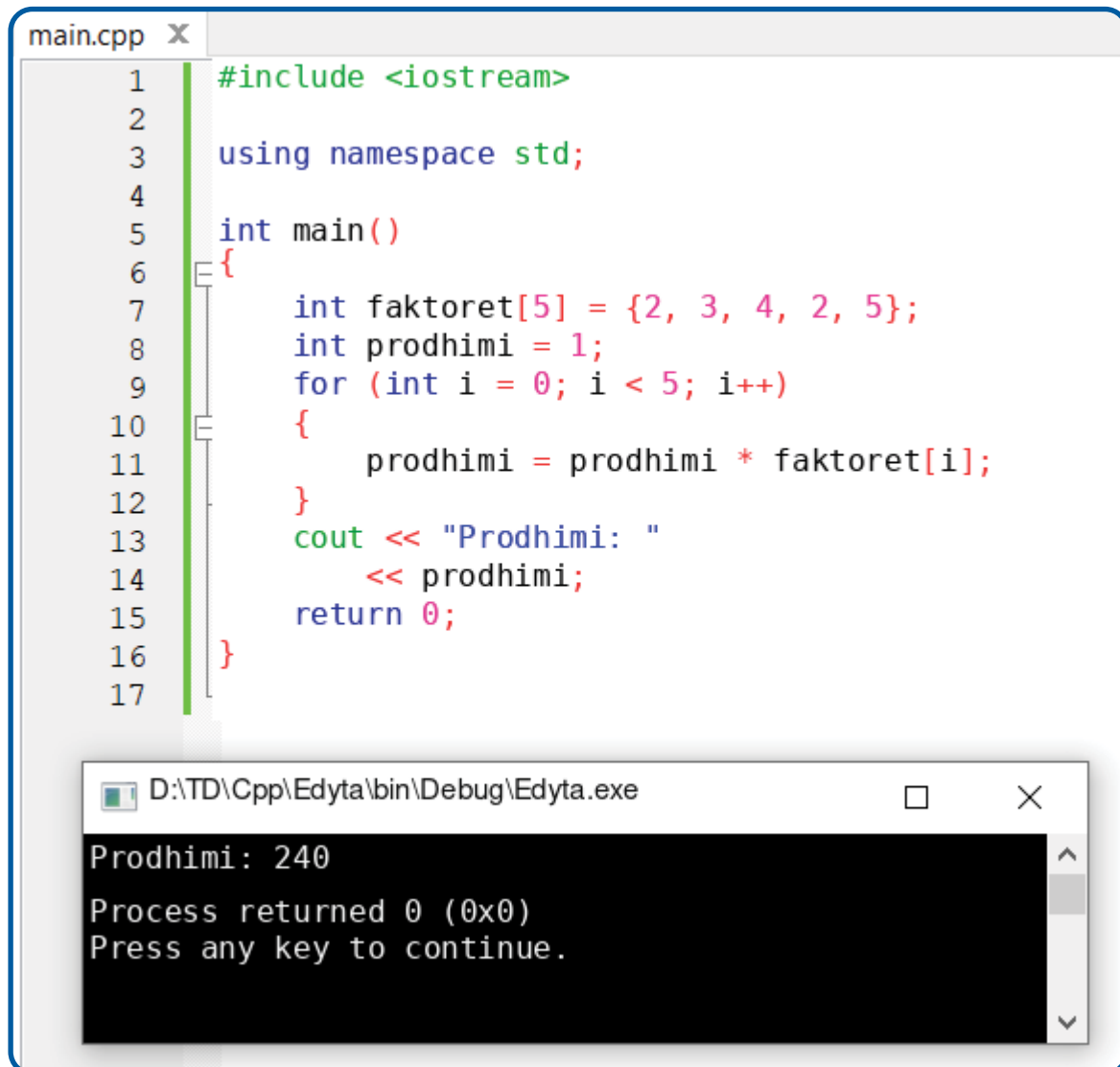
```

Në shembullin, të gjitha sasi të mjaltit të prodhuar i shtohen gradualisht njëra-tjetrës. Pas përfundimit të ciklit, merret prodhimi i përgjithshëm.

Jo gjithmonë është e nevojshme të llogaritet shuma. Në situata të caktuara duhet të llogaritet prodhimi i disa vlerave. Në detyra të tilla përdoret një ndryshore ndihmëse, e cila në fillim merr vlerën 1. Arsyeja është se shumëzimi me 1 nuk e ndryshon vlerën e numrave, ndërsa shumëzimi me 0 gjithmonë jep rezultatin 0. Në çdo përsëritje të ciklit, elementi aktual shumëzohet me rezultatin e mëparshëm.

Shembulli 2: Koeficienti i përgjithshëm i transmetimit

Në një eksperiment shkencor janë matur pesë koeficientë të përforcimit. Të llogaritet prodhimi i tyre.



The image shows a code editor window titled 'main.cpp' with a C++ program that calculates the product of an array of numbers. The code is as follows:

```
1  #include <iostream>
2
3  using namespace std;
4
5  int main()
6  {
7      int faktoret[5] = {2, 3, 4, 2, 5};
8      int prodhimi = 1;
9      for (int i = 0; i < 5; i++)
10     {
11         prodhimi = prodhimi * faktoret[i];
12     }
13     cout << "Prodhimi: "
14          << prodhimi;
15     return 0;
16 }
17
```

Below the code editor is a console window titled 'D:\TD\Cpp\Edyta\bin\Debug\Edyta.exe'. It displays the output of the program:

```
Prodhimi: 240
Process returned 0 (0x0)
Press any key to continue.
```

Në shembullin, ndryshorja proizvod përditësohet gradualisht gjatë përpunimit të çdo elementi të vargut. Pas përfundimit të ciklit, merret prodhimi i të gjitha vlerave.

Gjatë përpunimit të të dhënave të ruajtura në një varg, jo gjithmonë është e nevojshme të llogaritet vetëm shuma ose prodhimi. Në shumë situata praktike është e nevojshme që mbi secilin element të kryhet një llogaritje e caktuar dhe rezultatet e përfituara të kombinohen gradualisht.

Kjo mënyrë pune mundëson zgjidhjen e problemeve të ndryshme nga jeta e përditshme dhe nga fusha të ndryshme shkencore. Pavarësisht nga lloji i llogaritjes, ideja themelore mbetet e njëjtë. Cikli i përpunon në mënyrë të njëpasnjëshme të gjithë elementet e vargut, ndërsa mbi secilin element zbatohet veprimi

i nevojshëm. Në këtë mënyrë mund të llogaritet konsumi i përgjithshëm, energjia e prodhuar, shuma financiare ose shumë vlera të tjera.

Shembulli 3: Konsumi i ujit

Në një kamp kërkimor evidentohet konsumi ditor i ujit për pesë ekspedita. Çmimi i një litri ujë është 2 denarë. Të llogaritet shuma totale që duhet të paguhet.

The image shows a C++ program in a code editor and its execution output in a console window.

Code Editor (main.cpp):

```

1  #include <iostream>
2
3  using namespace std;
4
5  int main()
6  {
7      int uji[5] = {120, 150, 110, 140, 130};
8      int shuma = 0;
9      for (int i = 0; i < 5; i++)
10     {
11         shuma += uji[i] * 2;
12     }
13     cout << "Shuma totale: " << shuma << " denarë" << endl;
14     return 0;
15 }
16

```

Console Window (D:\TD\Cpp\Edyta\bin\Debug\Edyta.exe):

```

Shuma totale: 1300 denarë
Process returned 0 (0x0)
Press any key to continue.

```

Në shembull nuk mbledhen vetëm vlerat nga vargu. Për secilin element fillimisht llogaritet çmimi i tij, dhe më pas kjo vlerë i shtohet shumës totale. Në këtë mënyrë kombinohen veprimet aritmetike dhe përpunimi i elementeve të vargut.

Gjatë përpunimit të vargjeve, zgjedhja e veprimit që do të kryhet varet nga problemi që zgjidhet. Në disa situata llogaritet shuma, në të tjera prodhimi, ndërsa në raste të tjera një shprehje e caktuar matematikore për secilin element. Karakteristikë e përbashkët e të gjitha këtyre detyrave është se çdo element i vargut përpunohet saktësisht një herë. Falë kësaj, programet mbeten të thjeshta dhe të lehta për t'u kuptuar, edhe kur punojnë me një numër të madh të dhënash. Kjo mënyrë përpunimi përbën bazën për një numër të madh algoritmesh që do të studiohen në mësimet e ardhshme.

PYETJE DHE DETYRA PËR PËRSËRITJE



PYETJE

1. Pse gjatë përpunimit të një vargu më së shpeshti përdoret cikli for?
2. Me cilën vlerë fillestare më së shpeshti fillon ndryshorja për shumën?
3. Pse ndryshorja për prodhimin më së shpeshti fillon me vlerën 1?
4. A mund të kryhet mbi secilin element të vargut një veprim aritmetik i ndryshëm?
5. Cila është përparësia e përpunimit të të gjithë elementeve me një cikël të vetëm?



DETYRA

1. Në një park me turbina me erë janë matur sasitë e energjisë elektrike të prodhuara nga gjashtë turbina me erë. Shkruaj një program që do të llogarisë sasinë totale të energjisë elektrike të prodhuar.
2. Në një fabrikë prodhohen pjesë që pakëtohen në grupe. Shkruaj një program që do të llogarisë prodhimin (shumëzimin) e numrit të grupeve të ruajtura në një varg.
3. Në një aksion ekologjik regjistrohet numri i pemëve të mbjella në pesë vendbanime. Për çdo pemë ndahen 50 denarë për mirëmbajtje. Shkruaj një program që do të llogarisë shumën totale të nevojshme për mirëmbajtjen e pemëve.
4. Në laboratorin robotik regjistrohet numri i detyrave të përfunduara me sukses nga katër robotë. Shkruaj një program që do të llogarisë numrin e përgjithshëm të detyrave të përfunduara me sukses.
5. Në një stacion kërkimor matet sasia e mostrave të mbledhura gjatë shtatë ditëve. Për çdo mostër llogariten 3 pikë. Shkruaj një program që do të llogarisë numrin e përgjithshëm të pikëve të fituara.



MBAJ MEND ÇFARË MËSOVE

- Elementet e një vargu mund të përpunohen njëri pas tjetrit me ciklin for.
- Për llogaritjen e vlerës së përgjithshme zakonisht përdoret një ndryshore për shumën.
- Për llogaritjen e prodhimit përdoret një ndryshore që fillon me vlerën 1.
- Mbi çdo element të vargut mund të zbatohet një operacion aritmetik.
- Përpunimi i të gjithë elementeve me një cikël mundëson zgjidhje programuese të thjeshta dhe efektive.



PËR ATA QË DËSHIROJNË TË DINË MË SHUMË

Në sistemet moderne të informacionit shpesh përpunohen sasi të mëdha të dhënash. Për shembull, mund të analizohen matje nga sensorët, transaksione financiare, eksperimente shkencore ose të dhëna statistikore. Edhe pse sisteme të tilla punojnë me mijëra ose miliona vlera, ideja themelore mbetet e njëjtë – të dhënat përpunohen në mënyrë të njëpasnjëshme dhe mbi çdo element kryhen operacionet e nevojshme. Njohja e këtyre teknikave themelore përbën një bazë të rëndësishme për të mësuar algoritme dhe struktura më të ndërlikuara të të dhënave.





Gjatë futjes së elementeve në një varg, më së shpeshti përdoret cikli for. Numëruesi i ciklit përcakton pozicionin e elementit që po futet në atë moment. Pasi të futen të gjitha vlerat, ato mund të përpunohen me një cikël tjetër ose gjatë vetë procesit të futjes. Falë kësaj mundësie, një numër i madh të dhënash mund të përpunohet shpejt dhe në mënyrë efikase. Kjo qasje ka zbatim të gjerë në sisteme të ndryshme informatike dhe përbën një nga teknikat themelore gjatë punës me vargje njëdimensionale.

Shembull 1: Gjatësia totale e shtigjeve për biçikleta

Në një park kombëtar futen gjatësitë e pesë shtigjeve për biçikleta. Duhet të llogaritet gjatësia e tyre totale.

The image shows a C++ program in a code editor and its execution in a console window. The code defines an array of 5 integers to store the lengths of bicycle paths. It uses a for loop to read each value from the user and calculate the total. The output shows the individual path lengths and the total distance of 50 km.

```

main.cpp x
1  #include <iostream>
2
3  using namespace std;
4
5  int main()
6  {
7      int shtigjet[5];
8      int shuma = 0;
9
10     for (int i = 0; i < 5; i++)
11     {
12         cout << "Gjatësia e shtegut " << i + 1 << ": ";
13         cin >> shtigjet[i];
14         shuma += shtigjet[i];
15     }
16     cout << "Gjatësia totale: " << shuma << " km" << endl;
17     return 0;
18 }
19
D:\TD\Cpp\Edyta\bin\Debug\Edyta.exe
Gjatësia e shtegut 1: 14
Gjatësia e shtegut 2: 7
Gjatësia e shtegut 3: 2
Gjatësia e shtegut 4: 9
Gjatësia e shtegut 5: 18
Gjatësia totale: 50 km
Process returned 0 (0x0)
Press any key to continue.
  
```

Në këtë shembull, të dhënat futen nga tastiera dhe, me çdo futje të një vlere të re, ajo i shtohet shumës totale. Pas përfundimit të ciklit, merret gjatësia totale e të gjitha shtigjeve.

Në zgjidhjet praktike programuese, jo gjithmonë është e nevojshme të llogaritet shuma. Në disa raste duhet të përcaktohet prodhimi i disa vlerave. Në raste të tilla përdoret një ndryshore shtesë, e cila në fillim merr vlerën 1. Me çdo futje të një vlere të re, ajo shumëzohet me rezultatin e mëparshëm. Në këtë mënyrë, gradualisht përfitohet prodhimi i të gjithë elementeve të futura.

Shembull 2: Prodhimi i faktorëve

Në një laborator shkencor futen pesë faktorë zmadhimi. Duhet të llogaritet prodhimi i tyre.



```
main.cpp x
1  #include <iostream>
2
3  using namespace std;
4
5  int main()
6  {
7      int faktoret[5];
8      int prodhimi = 1;
9      for (int i = 0; i < 5; i++)
10     {
11         cout << "Faktori " << i + 1 << ": ";
12         cin >> faktoret[i];
13         prodhimi *= faktoret[i];
14     }
15     cout << "Prodhimi: "
16          << prodhimi
17          << endl;
18
19     return 0;
20 }
```

D:\TD }

```
Faktori 1: 2
Faktori 2: 1
Faktori 3: 2
Faktori 4: 4
Faktori 5: 3
Prodhimi: 48

Process returned 0 (0x0)
Press any key to continue.
```

Gjatë përpunimit të të dhënave të futura shpesh lind nevoja të përcaktohet vlera e tyre mesatare. Për këtë qëllim fillimisht duhet të llogaritet shuma e përgjithshme e të gjithë elementeve të vargut. Më pas rezultati i përfutur pjesëtohet me numrin e elementeve të futura.

Megjithatë rezultati i pjesëtimit mund të përmbajë pjesë dhjetore, gjatë llogaritjes së mesatares aritmetike zakonisht përdoret një ndryshore e tipit double. Në këtë mënyrë përfitohet një rezultat më i saktë dhe shmanget humbja e vlerave dhjetore.

Kjo mënyrë e punës ka zbatim të gjerë në analizat statistikore, kërkimet shkencore dhe përpunimin e llojeve të ndryshme të të dhënave.

Shembull 3: Ekspozimi mesatar ditor ndaj diellit

Në një observator astronomik regjistrohet numri i orëve të dritës së diellit gjatë pesë ditëve. Duhet të llogaritet numri i përgjithshëm i orëve dhe ekspozimi mesatar ditor ndaj diellit.

The screenshot shows a C++ program in a text editor and its execution in a console window. The program calculates the average daily exposure to sunlight based on five days of observations.

```

main.cpp x
1  #include <iostream>
2
3  using namespace std;
4
5  int main()
6  {
7      int oret[5];
8      int shuma = 0;
9      double mesatarja;
10     for (int i = 0; i < 5; i++)
11     {
12         cout << "Orët e dritës së diellit " << i + 1 << ": ";
13         cin >> oret[i];
14         shuma += oret[i];
15     }
16     mesatarja = (double)shuma / 5;
17     cout << "Orët e përgjithshme: " << shuma << endl;
18     cout << "Vlera mesatare: " << mesatarja << endl;
19     return 0;
20 }
21
D:\TD\Cpp\Edyta\bin\Debug\Edyta.exe
Orët e dritës së diellit 1: 2
Orët e dritës së diellit 2: 3
Orët e dritës së diellit 3: 4
Orët e dritës së diellit 4: 1
Orët e dritës së diellit 5: 3
Orët e përgjithshme: 13
Vlera mesatare: 2.6
Process returned 0 (0x0)
Press any key to continue.

```

Në këtë shembull, të dhënat fillimisht futen në një varg. Gjatë futjes së tyre, gradualisht llogaritet shuma e tyre. Pas përfundimit të ciklit, përcaktohet mesatarja aritmetike duke pjesëtuar shumën totale me numrin e elementeve. Për ruajtjen e rezultetit përdoret një ndryshore e tipit double, e cila mundëson paraqitjen edhe të pjesës dhjetore.

Gjatë përpunimit të të dhënave të futura nga përdoruesi mund të kryhen llogaritje të ndryshme, në varësi të problemit të përcaktuar. Në disa raste llogaritet shuma, në raste të tjera prodhimi, ndërsa në raste të tjera mesatarja aritmetike.

Karakteristikë e përbashkët e të gjitha këtyre detyrave është se të dhënat fillimisht futen në një varg, e më pas përpunohen me ndihmën e një cikli. Falë kësaj procedure, një program mund të përdoret për përpunimin e grupeve të ndryshme të të dhënave, pa pasur nevojë të ndryshohet. Kjo qasje përbën bazën për një numër të madh zgjidhjesh programuese që do të mësohen në mësimet e ardhshme.

PYETJE DHE DETYRA PËR PËRSËRITJE



PYETJE

1. Cila është përparësia e përdorimit të ciklit for gjatë futjes së më shumë elementeve në një varg?
2. Pse gjatë llogaritjes së mesatares aritmetike shpesh përdoret tipi double?
3. Cila është përparësia e futjes së të dhënave nga tastiera në vend të përcaktimit të tyre paraprakisht?



DETYRA

1. Në një kopsht botanik futen lartësitë e gjashtë pemëve të sapombjella. Shkruaj një program që do të llogarisë lartësinë e tyre totale.
2. Në një qendër robotike futet numri i detyrave të përfunduara me sukses nga pesë robotë. Shkruaj një program që do të llogarisë prodhimin e vlerave të futura.
3. Në një stacion oqeanografik futen pesë matje ditore të lartësisë së valëve. Shkruaj një program që do të llogarisë mesataren e tyre aritmetike.
4. Në një ekspeditë arkeologjike futet numri i objekteve të gjetura në shtatë lokalitete. Shkruaj një program që do të llogarisë numrin e përgjithshëm të objekteve.
5. Në një qendër për energji të rinovueshme futet energjia e prodhuar nga pesë sisteme diellore. Shkruaj një program që do të llogarisë energjinë totale dhe energjinë mesatare të prodhuar.



MBAJ MEND ÇFARË MËSOVE

- Elementet e një vargu mund të futen nga tastiera me ndihmën e ciklit for.
- Të dhënat e futura mund të përpunohen menjëherë.
- Për llogaritjen e vlerës totale përdoret një ndryshore për shumën.
- Për llogaritjen e prodhimit përdoret një ndryshore me vlerë fillestare 1.
- Mesatarja aritmetike përfitohet duke pjesëtuar shumën totale me numrin e elementeve.
- Gjatë llogaritjes së mesatares, më së shpeshti përdoret tipi double.



PËR ATA QË DËSHIROJNË TË DINË MË SHUMË

Në sistemet moderne informatike, një pjesë e madhe e të dhënave nuk dihen paraprakisht, por futen gjatë punës së programit. Për shembull, matjet nga sensorët, transaksionet financiare, analizat mjekësore dhe rezultatet sportive plotësohen vazhdimisht me informacione të reja.

Për këtë arsye, programet duhet të jenë në gjendje të pranojnë të dhëna nga përdoruesi dhe t'i përpunojnë menjëherë. Pikërisht kjo mundësi përbën një nga zbatimet më të rëndësishme të vargjeve njëdimensionale dhe cikleve në programimin modern.



PËRCAKTIMI I ELEMENTIT MË TË MADH DHE MË TË VOGËL NË NJË VARG

TASHMË DI ...

- të deklarosh dhe të inicializosh një varg njëdimensional;
- të futësh elemente në një varg;
- të përpunosh të dhënat me ciklin for;
- të llogaritësh shumën, prodhimin dhe mesataren aritmetike;
- të kombinosh futjen dhe përpunimin e të dhënave.

DETYRA PËR PËRSËRITJE

1. Si futen më shumë elemente në një varg?
2. Pse gjatë punës me vargje më së shpeshti përdoret cikli for?
3. Me cilën vlerë fillestare fillon ndryshorja për shumën?
4. Si llogaritet mesatarja aritmetike?
5. Pse është e dobishme që të dhënat të përpunohen menjëherë pas futjes së tyre?

NË PËRMBAJTJET QË VIJOJNË DO TË MËSOSH:

- të përcaktosh elementin më të madh në një varg;
- të përcaktosh elementin më të vogël në një varg;
- të krahasosh elementet e një vargu;
- të përdorësh ndryshore shtesë gjatë krahasimit;
- të përpunosh të dhënat duke përdorur konstrukione kushtore.

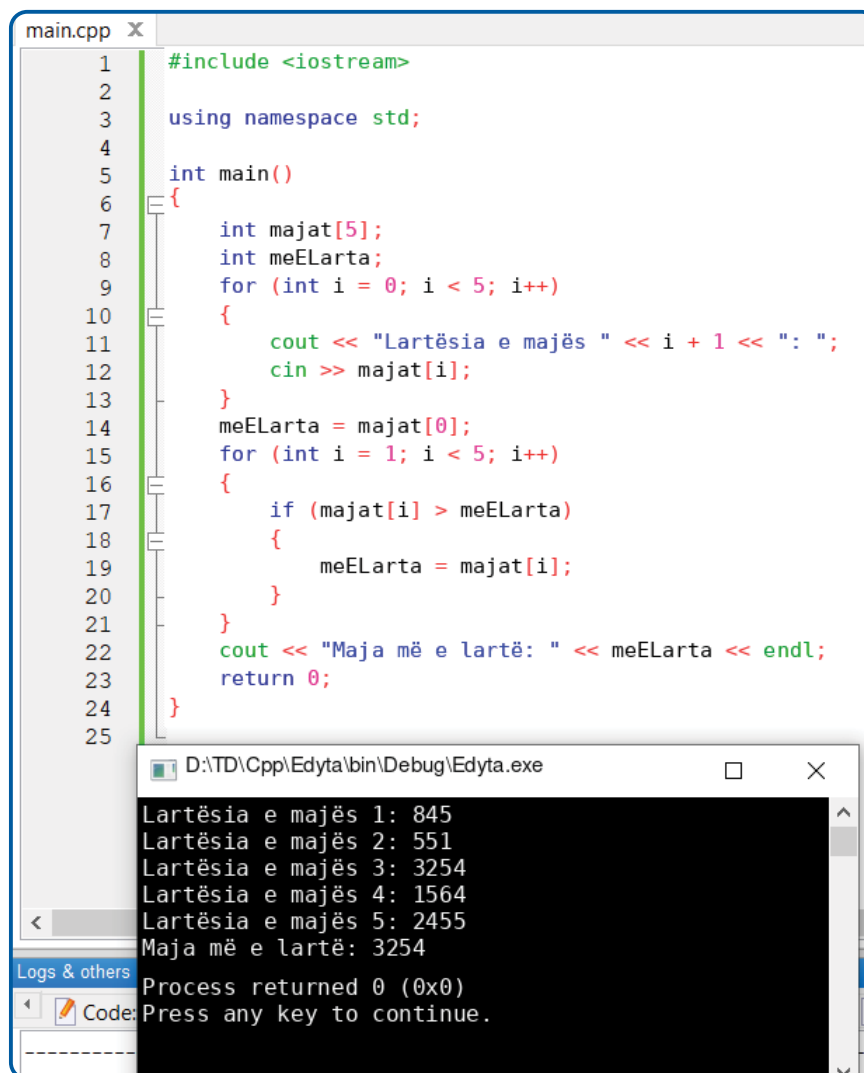
Gjatë përpunimit të të dhënave shpesh lind nevoja të përcaktohet vlera më e madhe ose më e vogël nga një bashkësi e caktuar të dhënash. Për shembull, mund të jetë e nevojshme të përcaktohet lartësia më e madhe e matur, sasia më e vogël e reshjeve ose shpejtësia më e lartë e arritur.

Për zgjidhjen e detyrave të tilla përdoret krahasimi i elementeve të vargut. Fillimisht një vlerë zgjidhet si vlerë fillestare, pastaj elementet e tjera krahasohen me të njëri pas tjetrit. Nëse gjendet një vlerë më e madhe ose më e vogël, vlera fillestare zëvendësohet me vlerën e re.

Gjatë përcaktimit të elementit më të madh ose më të vogël, fillimisht duhet të zgjidhet një vlerë nga vargu si vlerë fillestare. Më së shpeshti merret elementi i parë i vargut. Pastaj, me ndihmën e një cikli, shqyrtohen njëri pas tjetrit elementet e mbetur. Nëse ndonjëri prej tyre është më i madh se vlera më e madhe e ruajtur deri në atë moment, ajo zëvendësohet. Në të njëjtën mënyrë mund të përcaktohet edhe vlera më e vogël. Falë kësaj procedure, të gjitha të dhënat përpunohen me një kalim të vetëm nëpër varg.

Shembull 1: Maja më e lartë malore

Në një udhëzues malor futen lartësitë mbidetare të pesë majave malore. Duhet të përcaktohet maja më e lartë.



The screenshot shows a C++ program in a text editor and its execution in a console window. The program, named `main.cpp`, includes `<iostream>` and uses the `std` namespace. It defines a `main` function that initializes an array `majat` of size 5 and a variable `meELarta`. A loop from `i = 0` to `i = 4` prompts the user to enter the height of each mountain. After the loop, another loop from `i = 1` to `i = 4` finds the maximum value in the array. Finally, it prints the maximum height and returns 0.

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main()
6 {
7     int majat[5];
8     int meELarta;
9     for (int i = 0; i < 5; i++)
10    {
11        cout << "Lartësia e majës " << i + 1 << ": ";
12        cin >> majat[i];
13    }
14    meELarta = majat[0];
15    for (int i = 1; i < 5; i++)
16    {
17        if (majat[i] > meELarta)
18        {
19            meELarta = majat[i];
20        }
21    }
22    cout << "Maja më e lartë: " << meELarta << endl;
23    return 0;
24 }
```

The console window shows the execution of `D:\TD\Cpp\Edyta\bin\Debug\Edyta.exe`. It displays the prompts and inputs for the five mountains, followed by the output: "Maja më e lartë: 3254". The process returns 0 (0x0) and prompts the user to press any key to continue.

```
Lartësia e majës 1: 845
Lartësia e majës 2: 551
Lartësia e majës 3: 3254
Lartësia e majës 4: 1564
Lartësia e majës 5: 2455
Maja më e lartë: 3254
Process returned 0 (0x0)
Press any key to continue.
```

Në shembull, elementi i parë zgjidhet si vlerë fillestare. Më pas, të gjithë elementet e tjerë krahasohen me të. Nëse gjendet një vlerë më e madhe, ajo bëhet vlera e re më e madhe.

Gjatë përcaktimit të elementit më të vogël përdoret e njëjta procedurë. Dallimi është se, në vend që të kërkohet një vlerë më e madhe, kontrollohet nëse elementi aktual është më i vogël se vlera më e vogël e ruajtur deri në atë moment. Në këtë mënyrë mund të përcaktohet vlera më e vogël në një grup të dhënash pa llogaritje shtesë të ndërlikuara.

Shembull 2: Sasia më e vogël e karburantit

Në një ekspeditë hulumtuese regjistrohen sasitë e karburantit në gjashtë rezervuarë. Të përcaktohet rezervuari me sasinë më të vogël të karburantit.

The image shows a C++ program in a code editor and its execution output in a console window. The code defines an array of fuel amounts for six reservoirs and finds the minimum value.

```

main.cpp x
1  #include <iostream>
2
3  using namespace std;
4
5  int main()
6  {
7      int karburanti[6];
8      int meEVogla;
9      for (int i = 0; i < 6; i++)
10     {
11         cout << "Rezervuari " << i + 1 << ": ";
12         cin >> karburanti[i];
13     }
14     meEVogla = karburanti[0];
15     for (int i = 1; i < 6; i++)
16     {
17         if (karburanti[i] < meEVogla)
18         {
19             meEVogla = karburanti[i];
20         }
21     }
22     cout << "Sasia më e vogël: " << meEVogla << endl;
23     return 0;
24 }
25
D:\TD\Cpp\Edyta\bin\Debug\Edyta.exe
Rezervuari 1: 230
Rezervuari 2: 540
Rezervuari 3: 155
Rezervuari 4: 652
Rezervuari 5: 210
Rezervuari 6: 145
Sasia më e vogël: 145
Process returned 0 (0x0)
Press any key to continue.

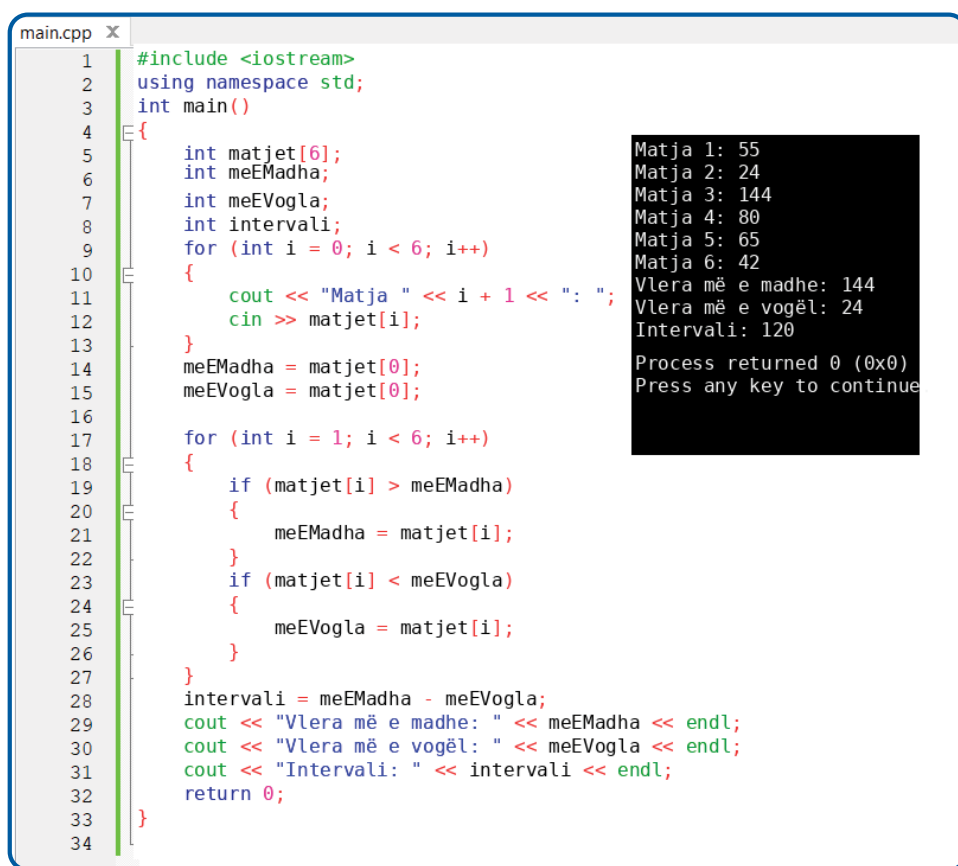
```

Në shembullin zbatohet e njëjta ide programore, por këtë herë përcaktohet vlera më e vogël nga të dhënat e futura.

Në detyrat praktike programore shpesh lind nevoja që njëkohësisht të përcaktohen vlera më e madhe dhe më e vogël nga një grup të dhënash. Në këtë rast, nuk është e nevojshme që vargu të përpunohet dy herë. Gjatë një kalimi nëpër elementet e vargut, mund të përditësohen njëkohësisht si vlera më e madhe, ashtu edhe vlera më e vogël. Pas përfundimit të përpunimit, mbi rezultatet e shembullin mund të kryhen edhe llogaritje shtesë. Njëra prej tyre është përcaktimi i shtrirjes së të dhënave, e cila llogaritet si diferenca midis vlerës më të madhe dhe vlerës më të vogël.

Shembulli 3: Shtrirja e vlerave të matura

Në një stacion shkencor futen gjashtë vlera të matura. Të përcaktohen vlera më e madhe, vlera më e vogël dhe shtrirja e matjeve.



```
main.cpp X
1  #include <iostream>
2  using namespace std;
3  int main()
4  {
5      int matjet[6];
6      int meEMadha;
7      int meEVogla;
8      int intervali;
9      for (int i = 0; i < 6; i++)
10     {
11         cout << "Matja " << i + 1 << ": ";
12         cin >> matjet[i];
13     }
14     meEMadha = matjet[0];
15     meEVogla = matjet[0];
16
17     for (int i = 1; i < 6; i++)
18     {
19         if (matjet[i] > meEMadha)
20         {
21             meEMadha = matjet[i];
22         }
23         if (matjet[i] < meEVogla)
24         {
25             meEVogla = matjet[i];
26         }
27     }
28     intervali = meEMadha - meEVogla;
29     cout << "Vlera më e madhe: " << meEMadha << endl;
30     cout << "Vlera më e vogël: " << meEVogla << endl;
31     cout << "Intervali: " << intervali << endl;
32     return 0;
33 }
34
```

Matja 1: 55
Matja 2: 24
Matja 3: 144
Matja 4: 80
Matja 5: 65
Matja 6: 42
Vlera më e madhe: 144
Vlera më e vogël: 24
Intervali: 120
Process returned 0 (0x0)
Press any key to continue

Në shembull, me një kalim nëpër varg përcaktohen njëkohësisht vlera më e madhe dhe vlera më e vogël. Më pas, duke përdorur këto vlera, llogaritet shtrirja e të dhënave.

Në përpunimin e të dhënave, përcaktimi i vlerave më të mëdha dhe më të vogla ka zbatim të gjerë. Për shembull, mund të përcaktohet temperatura më e lartë, konsumi më i ulët i energjisë ose shpejtësia më e madhe e matur. Pavarësisht nga lloji i të dhënave, procedura bazë mbetet e njëjtë. Fillimisht zgjidhet një vlerë fillestare, më pas të gjithë elementet e tjerë krahasohen në mënyrë të njëpasnjëshme me të dhe, sipas nevojës, përditësohen rezultatet e krahasimit.

PYETJE DHE DETYRA PËR PËRSËRITJE



PYETJE

1. Pse gjatë përcaktimit të vlerës më të madhe si vlerë fillestare më së shpeshti zgjidhet elementi i parë?
2. Si përcaktohet elementi më i vogël në një varg?
3. A mund të përcaktohen vlera më e madhe dhe më e vogël me një kalim nëpër varg?
4. Çfarë paraqet shtrirja e të dhënave?
5. Pse në këto detyra më së shpeshti përdoret cikli for?



DETYRA

1. Në një stacion meteorologjik futen sasinë e borës të matura në pesë qendra malore. Shkruaj një program që do të përcaktojë sasinë më të madhe të matur.
2. Në një qendër për mbrojtjen e kafshëve futen peshat e gjashtë arinjve të rinj. Shkruaj një program që do të përcaktojë peshën më të vogël të matur.
3. Në një ekspeditë oqeanografike futen thellësitë e shtatë matjeve. Shkruaj një program që do të përcaktojë thellësinë më të madhe dhe më të vogël.
4. Në një hulumtim të cilësisë së ajrit futen vlerat ditore nga pesë stacione matëse. Shkruaj një program që do të llogarisë shtrirjen e vlerave të matura.
5. Në një ekspeditë arkeologjike futen moshat e gjashtë objekteve të gjetura. Shkruaj një program që do të përcaktojë moshën më të madhe, moshën më të vogël dhe diferencën midis tyre.



MBAJ MEND ÇFARË MËSOVE

- Vlera më e madhe dhe më e vogël mund të përcaktohen duke krahasuar elementet e vargut.
- Vlera fillestare më së shpeshti është elementi i parë i vargut.
- Gjatë gjetjes së një vlere më të madhe ose më të vogël, rezultati përditësohet.
- Vlera më e madhe dhe më e vogël mund të përcaktohen me një kalim nëpër varg.
- Shtrirja e të dhënave merret si diferenca midis vlerës më të madhe dhe vlerës më të vogël.



PËR ATA QË DËSHIROJNË TË DINË MË SHUMË

Përcaktimi i vlerës më të madhe dhe më të vogël përbën një nga procedurat më të përdorura gjatë analizës së të dhënave. Në shkencë mund të përcaktohet temperatura më e lartë e matur, në mjekësi vlera më e madhe e matur e një treguesi të caktuar, ndërsa në sport rezultati më i mirë i arritur. Edhe pse sistemet moderne informatike përpunojnë sasi të mëdha të të dhënave, ideja themelore mbetet e njëjtë – të dhënat krahasohen në mënyrë të njëpasnjëshme dhe rezultatet e përfituara përditësohen kur gjendet një vlerë më e madhe ose më e vogël.



2.21

NUMËRIMI I ELEMENTEVE NË NJË VARG

TASHMË DI ...

- të futësh elemente në një varg njëdimensional;
- të përpunosh elementet me ciklin for;
- të llogarisësh shumën dhe prodhimin;
- të përcaktosh elementin më të madh dhe më të vogël;
- të përdorësh struktura kushtore gjatë përpunimit të të dhënave.

DETYRA PËR PËRSËRITJE

1. Pse gjatë përpunimit të një vargu më së shpeshti përdoret cikli for?
2. Si përcaktohet elementi më i madh në një varg?
3. Si përcaktohet elementi më i vogël?
4. A është e mundur që më shumë vlera të përpunohen me një kalim nëpër varg?
5. Çfarë paraqet shtrirja e të dhënave?

NË PËRMBAJTJET QË VIJOJNË DO TË MËSOSH:

- të numërosh elementet që plotësojnë një kusht të caktuar;
- të përdorësh një ndryshore për numërim;
- të kombinosh ciklet dhe strukturat kushtore;
- të zbatosh shprehje logjike gjatë përpunimit të vargjeve;
- të analizosh të dhënat sipas kriterëve të ndryshme.

Gjatë përpunimit të të dhënave shpesh nuk mjafton të përcaktohet vetëm shuma e tyre ose vlera më e madhe. Në shumë situata praktike është e nevojshme të përcaktohet sa elemente kanë një veti të caktuar. Për shembull, mund të jetë e nevojshme të përcaktohet numri i ditëve me reshje, numri i automjeteve që e kanë tejkaluar shpejtësinë e lejuar ose numri i matjeve të kryera me sukses.

Për zgjidhjen e detyrave të tilla përdoret një ndryshore për numërim. Në fillim asaj i jepet vlera 0. Gjatë përpunimit të çdo elementi kontrollohet nëse plotësohet kushti i dhënë. Nëse kushti plotësohet, vlera e

ndryshores rritet për 1. Pas përpunimit të të gjithë elementeve, ajo përmban numrin e përgjithshëm të elementeve që e plotësojnë kushtin.

Gjatë numërimit të elementeve të një vargu, fillimisht krijohet një ndryshore për numërim, së cilës i jepet vlera fillestare 0. Më pas, të gjithë elementet përpunohen njëri pas tjetrit me ndihmën e një cikli. Për çdo element kontrollohet nëse e plotëson kushtin e dhënë. Nëse kushti plotësohet, vlera e ndryshores për numërim rritet për 1. Pas përpunimit të elementit të fundit, ndryshorja e përmban rezultatin përfundimtar. Kjo mënyrë pune ka zbatim të gjerë në përpunimin statistikor dhe analizën e llojeve të ndryshme të të dhënave.

Shembulli 1: Bimë të larta

Në një kopsht botanik futen lartësitë e shtatë bimëve. Të përcaktohet sa bimë janë më të larta se 100 cm.

The image shows a C++ program in a code editor and its execution in a console window. The code defines an array of heights and counts how many are greater than 100 cm.

```

main.cpp x
1  #include <iostream>
2
3  using namespace std;
4
5  int main()
6  {
7      int lartesia[7];
8      int numeruesi = 0;
9      for (int i = 0; i < 7; i++)
10     {
11         cout << "Lartësia " << i + 1 << ": ";
12         cin >> lartesia[i];
13         if (lartesia[i] > 100)
14         {
15             numeruesi++;
16         }
17     }
18     cout << "Numri i bimëve: " << numeruesi << endl;
19     return 0;
20 }
21

```

The console window shows the following output:

```

D:\TD\Cpp\Edyta\bin\Debug\Edyta.exe
Lartësia 1: 105
Lartësia 2: 92
Lartësia 3: 56
Lartësia 4: 213
Lartësia 5: 84
Lartësia 6: 155
Lartësia 7: 142
Numri i bimëve: 4
Process returned 0 (0x0)
Press any key to continue.

```

Në shembullin, çdo vlerë e futur kontrollohet. Nëse bima është më e lartë se 100 cm, ndryshorja për numërim rritet për 1.

Në detyrat praktike, kushti nuk duhet domosdoshmërisht të lidhet me madhësinë e vlerave. Shpesh ka nevojë të numërohen elementet që kanë një veti të caktuar matematikore. Një nga vetitë më të zakonshme është pariteti i numrave. Për këtë qëllim përdoret operatori %, i cili mundëson përcaktimin e mbetjes gjatë pjesëtimit.

Shembulli 2: Rezultate çift

Në një qendër sportive futen gjashtë rezultate. Të përcaktohet sa prej tyre janë numra çift.



The image shows a C++ program in a code editor and its execution output in a console window. The code defines an array of 6 results, reads them from the user, and counts how many are even. The output shows the results and the final count of even numbers.

```
main.cpp x
1  #include <iostream>
2
3  using namespace std;
4
5  int main()
6  {
7      int rezultatet[6];
8      int numeruesi = 0;
9      for (int i = 0; i < 6; i++)
10     {
11         cout << "Rezultati " << i + 1 << ": ";
12         cin >> rezultatet[i];
13         if (rezultatet[i] % 2 == 0)
14         {
15             numeruesi++;
16         }
17     }
18     cout << "Rezultate çift: " << numeruesi << endl;
19     return 0;
20 }
21
```

```
D:\TD\Cpp\Edyta\bin\Debug\Edyta.exe
Rezultati 1: 2
Rezultati 2: 3
Rezultati 3: 4
Rezultati 4: 5
Rezultati 5: 6
Rezultati 6: 4
Rezultate çift: 4
Process returned 0 (0x0)
Press any key to continue.
```

Në shembull, për çdo vlerë të futur kontrollohet nëse është numër çift. Nëse kushti plotësohet, rritet numri i vlerave çift.

Gjatë përpunimit të të dhënave, shpesh ka nevojë të përcaktohet numri i elementeve që ndodhen në një interval të caktuar. Për shembull, mund të jetë e nevojshme të përcaktohet numri i temperaturave midis dy vlerave, numri i nxënësve me nota në një interval të caktuar ose numri i produkteve që i plotësojnë kushtet e vendosura. Në situata të tilla, për çdo element kontrollohen disa kushte njëkohësisht. Nëse plotësohen

të gjitha kushtet, ndryshorja për numërim rritet për 1. Në këtë mënyrë mund të përcaktohet lehtësisht numri i elementeve që i përkasin intervalit të dhënë.

Shembulli 3: Vlerat në një interval të caktuar

Në një stacion kërkimor futen shtatë vlera të matura. Të përcaktohet sa prej tyre ndodhen në intervalin nga 20 deri në 50.

The screenshot shows a C++ program in a code editor and its execution output in a console window.

Code Editor (main.cpp):

```

1  #include <iostream>
2
3  using namespace std;
4
5  int main()
6  {
7      int vlerat[7];
8      int numeruesi = 0;
9      for (int i = 0; i < 7; i++)
10     {
11         cout << "Vlera " << i + 1 << ": ";
12         cin >> vlerat[i];
13         if (vlerat[i] >= 20 && vlerat[i] <= 50)
14         {
15             numeruesi++;
16         }
17     }
18     cout << "Numri i elementeve: " << numeruesi << endl;
19     return 0;
20 }
21

```

Console Window (D:\TD\Cpp\Edyta\bin\Debug\Edyta.exe):

```

Vlera 1: 18
Vlera 2: 22
Vlera 3: 87
Vlera 4: 54
Vlera 5: 35
Vlera 6: 40
Vlera 7: 11
Numri i elementeve: 3
Process returned 0 (0x0)
Press any key to continue.

```

Në shembullin, për çdo vlerë të futur kontrollohet nëse i përket intervalit të dhënë. Nëse kushti plotësohet, ndryshorja për numërim rritet për 1. Pas përpunimit të të gjithë elementeve, merret rezultati përfundimtar.

Gjatë numërimit të elementeve, kushtet që duhet të plotësohen mund të jenë shumë të ndryshme. Në disa detyra kontrollohet nëse vlera është më e madhe se një numër i caktuar, në të tjera nëse është çift, ndërsa në të tretat nëse i përket një intervali të caktuar. Pavarësisht nga problemi i parashtruar, procedura themelore mbetet e njëjtë. Elementet përpunohen njëri pas tjetrit, kushti kontrollohet për çdo element dhe ndryshorja për numërim rritet kur kushti plotësohet. Falë kësaj teknike mund të zgjidhen një numër i madh detyrash praktike programore.

PYETJE DHE DETYRA PËR PËRSËRITJE



PYETJE

1. Për çfarë shërben ndryshorja për numërim?
2. Me cilën vlerë fillestare zakonisht fillon ndryshorja për numërim?
3. Kur rritet vlera e saj?
4. Si kontrollohet nëse një numër është çift?
5. Si mund të kontrollohet nëse elementi i përket një intervali të caktuar?



DETYRA

1. Në një park kombëtar futen lartësitë e tetë ujëvarave. Shkruaj një program që do të përcaktojë sa ujëvara janë më të larta se 50 metra.
2. Në një laborator robotik futen numrat e detyrave të përfunduara me sukses nga gjashtë robotë. Shkruaj një program që do të përcaktojë sa prej rezultateve janë numra tek.
3. Në një qendër për energji të rinovueshme futet prodhimi ditor i shtatë sistemeve diellore. Shkruaj një program që do të përcaktojë sa vlera ndodhen midis 100 dhe 300.
4. Në një ekspeditë oqeanografike futen thellësitë e gjashtë matjeve. Shkruaj një program që do të përcaktojë sa matje janë më të mëdha se 500 metra.
5. Në një observator astronomik futen numrat e meteorëve të vëzhguar gjatë pesë netëve. Shkruaj një program që do të përcaktojë në sa net janë regjistruar më shumë se 20 meteorë.



MBAJ MEND ÇFARË MËSOVE

- Për numërimin e elementeve përdoret një ndryshore për numërim.
- Ndryshorja për numërim zakonisht fillon me vlerën 0.
- Vlera e saj rritet kur plotësohet kushti i dhënë.
- Kushti mund të jetë i thjeshtë ose i përbërë nga disa kontrolle.
- Me një kalim nëpër varg mund të përcaktohet numri i elementeve që kanë një veti të caktuar.



PËR ATA QË DËSHIROJNË TË DINË MË SHUMË

Numërimi i elementeve përfaqëson një nga teknikat më të përdorura gjatë analizës së të dhënave. Në mjekësi mund të përcaktohet numri i pacientëve që kanë një simptomë të caktuar, në ekologji numri i ditëve me reshje, ndërsa në sport numri i përpjekjeve të realizuara me sukses. Edhe pse sistemet moderne informatike përpunojnë sasi të mëdha të të dhënave, ideja themelore mbetet e pandryshuar – çdo e dhënë kontrollohet dhe, nëse e plotëson kushtin e dhënë, rezultati i numërimit përditësohet.

2.22



VARGJE ME KARAKTERE DHE KËRKIMI I SHENJAVE

TASHMË DI ...

- të deklarosh dhe të përdorësh vargje njëdimensionale me numra të plotë;
- të futësh dhe të përpunosh elemente në varg;
- të përdorësh ciklin for;
- të zbatosh struktura kushtore;
- të numërosh elementet që plotësojnë një kusht të caktuar.

DETYRA PËR PËRSËRITJE

1. Çfarë paraqet një varg njëdimensional?
2. Nga cili numër fillon numërimi i elementeve?
3. Pse gjatë punës me vargje përdoret cikli for?
4. Për çfarë shërben ndryshorja për numërim?
5. Si mund të kontrollohet nëse një element e plotëson një kusht të caktuar?

NË PËRMBAJTJET QË VIJOJNË DO TË MËSOSH:

- të përdorësh vargje me karaktere;
- të futësh karaktere në varg;
- të kërkosh një karakter në varg;
- të përcaktosh nëse një karakter gjendet në varg;
- të numërosh paraqitjet e një karakteri.

Përveç numrave, në programe shpesh lind nevoja të përpunohen edhe karaktere. Për shembull, mund të jetë e nevojshme të kontrollohet nëse një shkronjë e caktuar gjendet në një kod, nëse një simbol i caktuar shfaqet në një tekst ose sa herë është përdorur një karakter i caktuar.

Për ruajtjen e karaktereve të veçanta në gjuhën programuese C++ përdoret tipi char. Më shumë karaktere mund të ruhen në një varg karakteresh. Në këtë mënyrë, çdo karakter ka pozicionin e vet në varg dhe mund të përpunohet veçmas.

Vargu i karaktereve deklarohet në të njëjtën mënyrë si vargjet e numrave të plotë. Dallimi është se në vend të tipit int përdoret tipi char. Çdo element i vargut përmban nga një karakter. Elementet mund të jenë shkronja, shifra ose simbole të tjera. Gjatë përpunimit të vargut, çdo karakter mund të krahasohet me një vlerë të caktuar dhe, në bazë të rezultatit, të merret një vendim. Falë kësaj mundësie mund të zgjidhen detyra të ndryshme praktike që lidhen me përpunimin e të dhënave tekstuale.

Shembulli 1: Kontrollimi nëse ekziston një karakter

Në një sistem kontrolli futen gjashtë karaktere. Të kontrollohet nëse midis tyre gjendet shkronja A.



```
main.cpp X
1  #include <iostream>
2
3  using namespace std;
4
5  int main()
6  {
7      char etiketat[6];
8      bool uGjet = false;
9      for (int i = 0; i < 6; i++) {
10         cout << "Etiketa "
11             << i + 1
12             << ": ";
13
14         cin >> etiketat[i];
15     }
16     for (int i = 0; i < 6; i++) {
17         if (etiketat[i] == 'A') {
18             uGjet = true;
19         }
20     }
21     if (uGjet) {
22         cout << "Shkronja ekziston.";
23     } else {
24         cout << "Shkronja nuk u gjet.";
25     }
26     return 0;
27 }
28
29
30
31
```

D:\TD\Cpp\Edyta\bin\Debug\Edyta.exe

Etiketa 1: B
Etiketa 2: A
Etiketa 3: C
Etiketa 4: D
Etiketa 5: E
Etiketa 6: F
Shkronja ekziston.

Logs & others

Code: Executing: Process returned 0 (0x0)
Press any key to continue.

Në shembullin, të gjithë karakteret fillimisht futen në varg. Më pas kontrollohet njëri pas tjetrit nëse ndonjëri prej tyre është i barabartë me shkronjën A. Nëse ajo gjendet, rezultati ruhet.

Gjatë përpunimit të karaktereve, shpesh nuk mjafton vetëm të përcaktohet nëse ekziston një karakter i caktuar. Në shumë situata është e nevojshme të përcaktohet sa herë shfaqet ai. Për këtë qëllim përdoret një ndryshore për numërim, e cila rritet sa herë që gjendet karakteri i kërkuar.

Shembulli 2: Numërimi i paraqitjeve

Në një sistem komunikimi futen tetë karaktere. Të përcaktohet sa herë shfaqet shkronja K.

```

main.cpp x
1  #include <iostream>
2
3  using namespace std;
4
5  int main()
6  {
7      char shenjat[8];
8      int numeruesi = 0;
9
10     for (int i = 0; i < 8; i++)
11     {
12         cout << "Shenja " << i + 1 << ": ";
13         cin >> shenjat[i];
14     }
15     for (int i = 0; i < 8; i++)
16     {
17         if (shenjat[i] == 'K')
18         {
19             numeruesi++;
20         }
21     }
22     cout << "Numri i përsëritjeve: "
23          << numeruesi << endl;
24     return 0;
25 }

```

D:\TD\Cpp\Edyta\bin\Debug\Edyta.exe

```

Shenja 1: K
Shenja 2: A
Shenja 3: B
Shenja 4: R
Shenja 5: K
Shenja 6: X
Shenja 7: C
Shenja 8: D
Numri i përsëritjeve: 2
Process returned 0 (0x0)
Press any key to continue.

```

Logs & others

Code::

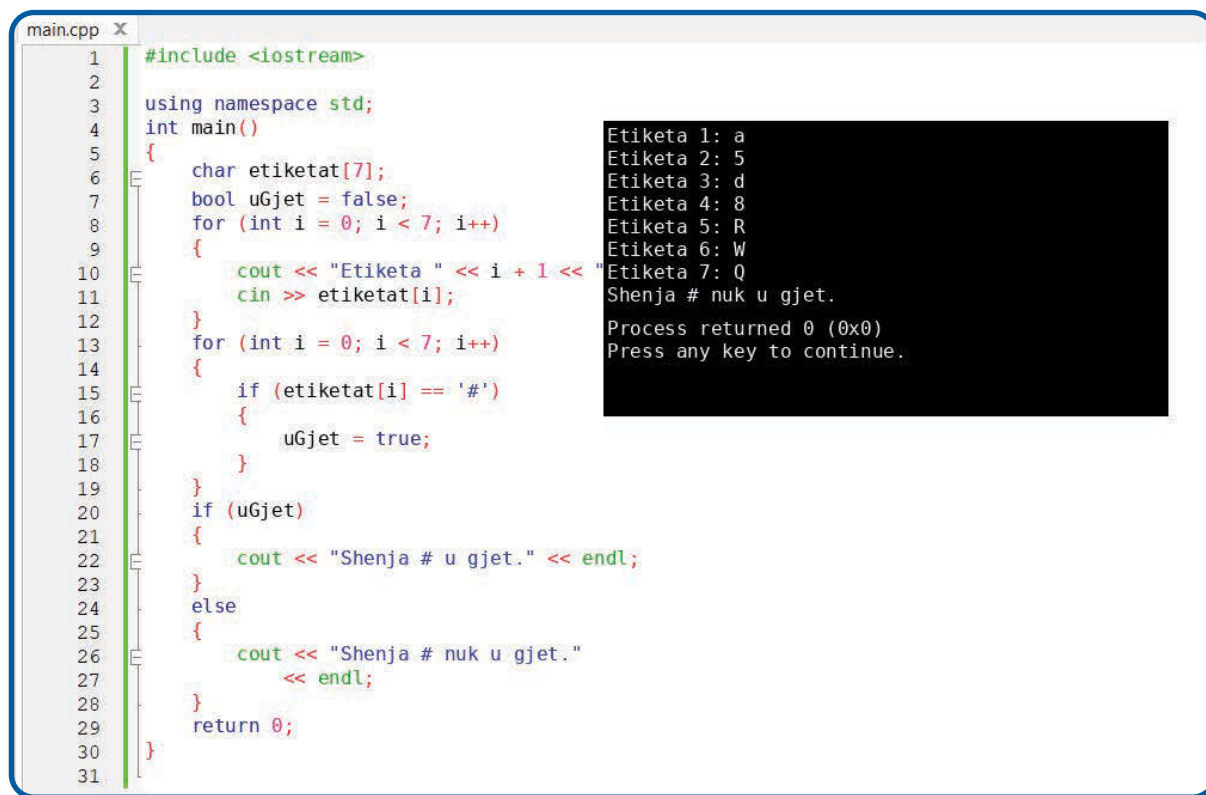
Executing:

Në shembullin përdoret teknika e numërimit. Sa herë që gjendet shkronja K, ndryshorja për numërim rritet për 1.

Gjatë punës me vargje karakteresh, nuk është gjithmonë e nevojshme të kërkohe vetëm një shkronjë e caktuar. Në detyrat praktike programuese shpesh kontrollohet nëse shfaqet një simbol i caktuar. Simbole të tilla mund të jenë #, @, * ose shenja të tjera që kanë kuptim të veçantë në sisteme të caktuara. Procedura e përpunimit mbetet e pandryshuar. Çdo element i vargut krahasohet njëri pas tjetrit me shenjën e kërkuar dhe, nëse ajo gjendet, nxirret përfundimi përkatës.

Shembulli 3: Kontrollimi i një simboli të veçantë

Në një sistem për qasje futen shtatë karaktere. Të kontrollohet nëse midis tyre gjendet simboli #.



```
main.cpp x
1  #include <iostream>
2
3  using namespace std;
4  int main()
5  {
6      char etiketat[7];
7      bool uGjet = false;
8      for (int i = 0; i < 7; i++)
9      {
10         cout << "Etiketa " << i + 1 << " ";
11         cin >> etiketat[i];
12     }
13     for (int i = 0; i < 7; i++)
14     {
15         if (etiketat[i] == '#')
16         {
17             uGjet = true;
18         }
19     }
20     if (uGjet)
21     {
22         cout << "Shenja # u gjet." << endl;
23     }
24     else
25     {
26         cout << "Shenja # nuk u gjet."
27             << endl;
28     }
29     return 0;
30 }
31
```

Etiketa 1: a
Etiketa 2: 5
Etiketa 3: d
Etiketa 4: 8
Etiketa 5: R
Etiketa 6: W
Etiketa 7: Q
Shenja # nuk u gjet.
Process returned 0 (0x0)
Press any key to continue.

Në shembullin, të gjithë karakteret e futura krahasohen njëri pas tjetrit me simbolin #. Nëse simboli gjendet, programi e njofton përdoruesin se simboli ndodhet në varg.

Gjatë përpunimit të një vargu karakteresh mund të vendosen kushte të ndryshme për kërkim. Në disa detyra kërkohe prania e një shkronje të caktuar, në të tjera numërohet numri i paraqitjeve të një karakteri të caktuar, ndërsa në të tretat kontrollohet nëse shfaqet një simbol i caktuar. Pavarësisht nga lloji i kushtit, procedura themelore mbetet e njëjtë. Çdo element i vargut përpunohet, bëhet krahasimi dhe, në bazë të rezultatit, nxirret një përfundim. Kjo teknikë përbën bazën për përpunimin e të dhënave tekstuale në gjuhët programuese.

PYETJE DHE DETYRA PËR PËRSËRITJE



PYETJE

1. Çfarë paraqet një varg karakteresh?
2. Cili lloj i të dhënave përdoret për ruajtjen e një shenje?
3. Si mund të kontrollohet nëse një shenje e caktuar gjendet në një varg?
4. Si mund të përcaktohet sa herë shfaqet një shenje e caktuar?
5. A mund të ruhen në një varg karakteresh shkronja, shifra dhe simbole speciale?



DETYRA

1. Në një sistem kontrolli futen tetë shenja. Shkruaj një program që do të kontrollojë nëse midis tyre gjendet shkronja M.
2. Në një qendër komunikimi futen gjashtë shenja. Shkruaj një program që do të përcaktojë se sa herë shfaqet shifra '5'.
3. Në një laborator informatik futen dhjetë shenja. Shkruaj një program që do të kontrollojë nëse është futur shenja @.
4. Në një sistem evidencimi futen shtatë shenja. Shkruaj një program që do të përcaktojë se sa herë shfaqet shkronja S.
5. Në një stacion kërkimor futen nëntë simbole. Shkruaj një program që do të kontrollojë nëse shfaqet shenja *.



MBAJ MEND ÇFARË MËSOVE

- Vargu i karaktereve paraqet një bashkësi të përbërë nga shenja të veçanta.
- Për ruajtjen e një shenje përdoret tipi char.
- Shenjat mund të jenë shkronja, shifra ose simbole speciale.
- Kërkimi i një shenje kryhet duke krahasuar elementet e vargut.
- Me ndihmën e një ndryshoreje numëruese mund të përcaktohet numri i shfaqjeve të një shenje të caktuar.



PËR ATA QË DËSHIROJNË TË DINË MË SHUMË

Në sistemet moderne informatike, përpunimi i të dhënave tekstuale përbën një nga veprimtaritë më të rëndësishme në programim. Gjatë futjes së emrave të përdoruesve, fjalëkalimeve, shenjave të produkteve ose kodeve të ndryshme, programet shpesh kontrollojnë nëse janë përdorur shenja ose simbole të caktuara. Edhe pse në këtë mësim trajtohen shenja të veçanta, të njëjtat parime zbatohen më vonë edhe gjatë punës me fjalë, fjali dhe të dhëna tekstuale më të mëdha.



2.23

ZBATIMI I FUNKSIONEVE MATEMATIKORE MBI ELEMENTET E VARGUT

TASHMË DI ...

- futësh elementet në një varg njëdimensional;
- përpunosh elementet me ciklin for;
- llogarisësh shumën, prodhimin dhe mesataren;
- përcaktosh elementin më të madh dhe më të vogël;
- përdorësh funksione nga biblioteka matematikore.

DETYRA PËR PËRSËRITJE

1. Për çfarë shërben biblioteka `cmath`?
2. Cili funksion llogarit rrënjën katrore?
3. Cili funksion përdoret për fuqizim?
4. Për çfarë shërben funksioni `abs()`?
5. A mund të zbatohen funksionet nga biblioteka matematikore mbi elementet e një vargu?

NË PËRMBAJTJET QË VIJOJNË DO TË MËSOSH:

- të zbatosh funksione matematikore mbi elementet e një vargu;
- të konbinosh vargjet me bibliotekën `cmath`;
- të përpunosh të dhënat e futura duke përdorur llogaritje matematikore;
- të krahasosh rezultatet e përfituara;
- të zgjidhësh detyra praktike duke kombinuar teknikat e mësuara.

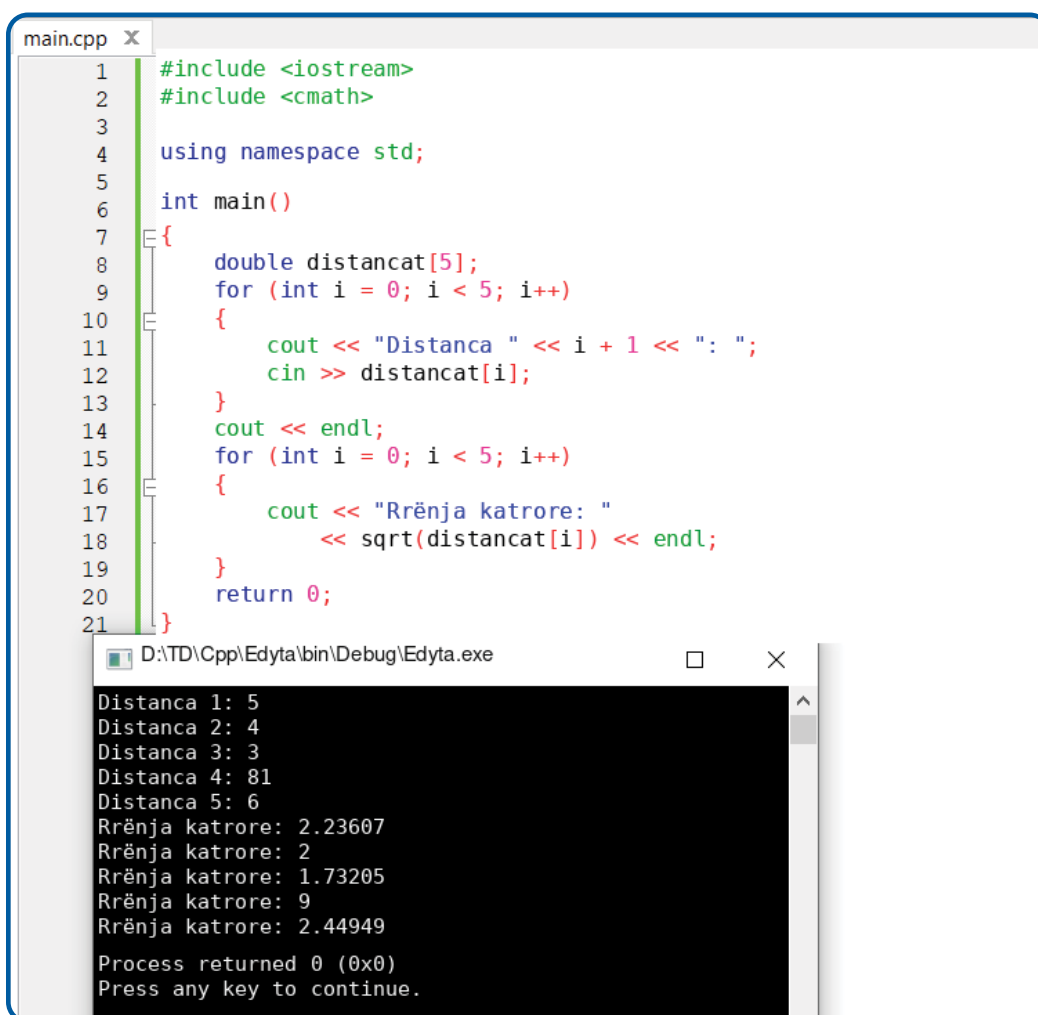
Gjatë përpunimit të të dhënave shpesh lind nevoja që mbi secilin element të një vargu të kryhet një veprim i caktuar matematikor. Në zgjidhjet praktike programuese mund të jetë e nevojshme të llogaritet rrënja katrore, katrori i një numri ose vlera absolute e të dhënave të futura. Për këtë qëllim, në gjuhën programuese C++ përdoret biblioteka matematikore `cmath`.

Funksionet e kësaj biblioteke mund të zbatohen mbi çdo element të vargut. Me ndihmën e një cikli, të gjitha vlerat përpunohen në mënyrë të njëpasnjëshme dhe mbi secilën prej tyre kryhet llogaritja e nevojshme matematikore. Në këtë mënyrë mund të zgjidhen probleme të ndryshme praktike nga shkenca, teknika dhe jeta e përditshme.

Gjatë zbatimit të funksioneve matematikore mbi elementet e një vargu, fillimisht futen të dhënat dhe më pas çdo element përpunohet në mënyrë të njëpasnjëshme. Për secilin element mund të llogaritet rrënja katrore, katrori, vlera absolute ose një funksion tjetër matematikor. Rezultatet e përfituara mund të shfaqen menjëherë ose të përpunohen më tej. Falë kësaj procedure, një numër i madh të dhënash mund të analizohen dhe përpunohen shpejt, duke përdorur një numër relativisht të vogël instruksionesh programuese.

Shembull 1: Rrënja katrore e distancave të matura

Në një laborator shkencor futen pesë distanca të matura. Llogarit rrënjën katrore të secilës distancë.



The image shows a C++ program in a code editor and its execution output in a console window. The code defines an array of distances and calculates the square root of each.

```

main.cpp
1  #include <iostream>
2  #include <cmath>
3
4  using namespace std;
5
6  int main()
7  {
8      double distancat[5];
9      for (int i = 0; i < 5; i++)
10     {
11         cout << "Distanca " << i + 1 << ": ";
12         cin >> distancat[i];
13     }
14     cout << endl;
15     for (int i = 0; i < 5; i++)
16     {
17         cout << "Rrënja katrore: "
18              << sqrt(distancat[i]) << endl;
19     }
20     return 0;
21 }
  
```

The console output shows the program's execution with the following data:

```

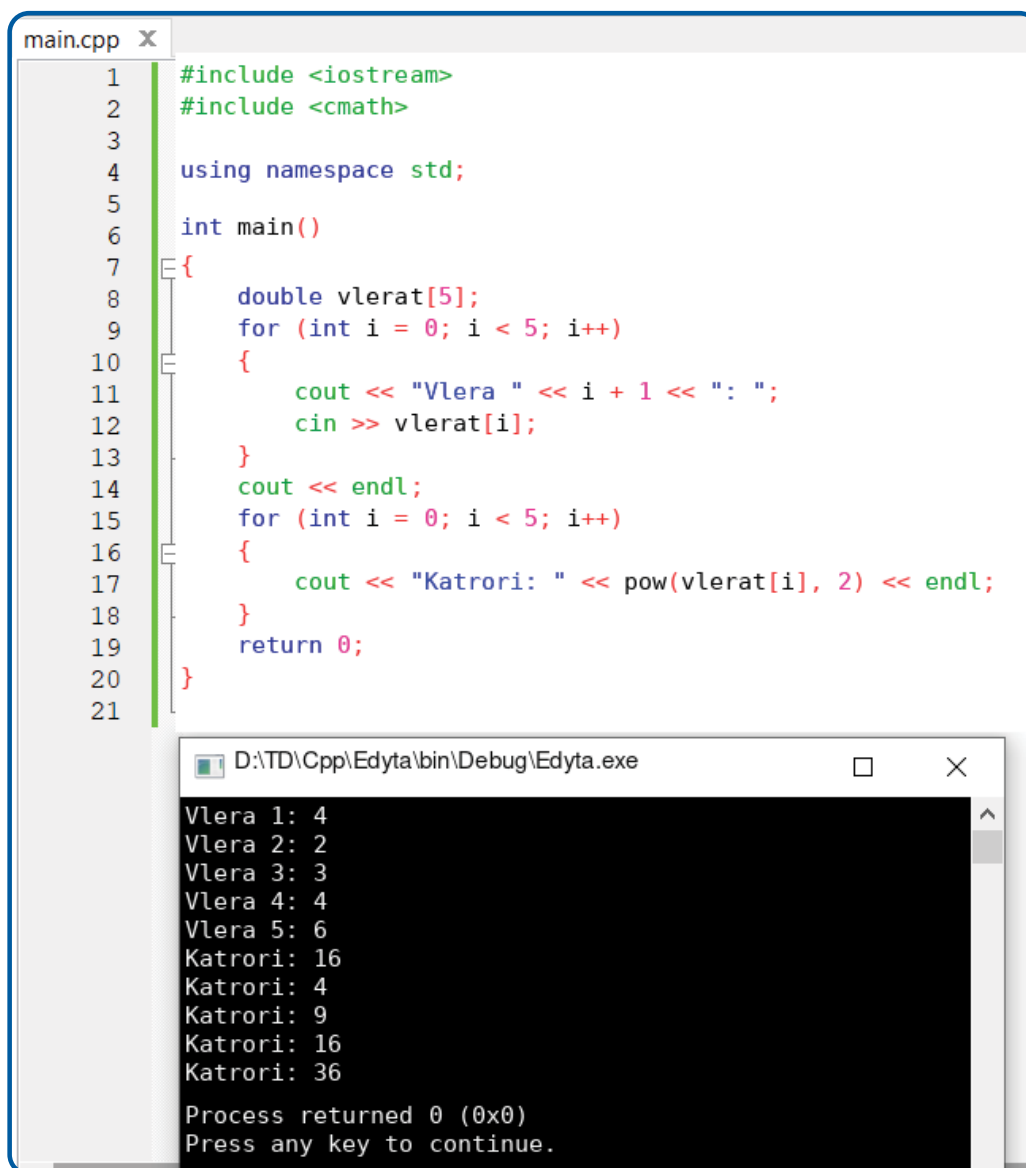
Distanca 1: 5
Distanca 2: 4
Distanca 3: 3
Distanca 4: 81
Distanca 5: 6
Rrënja katrore: 2.23607
Rrënja katrore: 2
Rrënja katrore: 1.73205
Rrënja katrore: 9
Rrënja katrore: 2.44949
Process returned 0 (0x0)
Press any key to continue.
  
```

Në këtë shembull, çdo e dhënë e futur përpunohet në mënyrë të pavarur. Me ndihmën e funksionit `sqrt()` llogaritet rrënja katrore e secilës vlerë dhe rezultati shfaqet menjëherë.

Gjatë përpunimit të të dhënave, shpesh lind nevoja që mbi të gjithë elementet e një vargu të zbatohet i njëjti veprim matematikor. Për shembull, mund të jetë e nevojshme të llogariten katrorët e të gjitha vlerave të matura ose të përcaktohet vlera absolute e secilës matje. Në këto situata, çdo element përpunohet në mënyrë të njëpasnjëshme dhe mbi të zbatohet funksioni përkatës matematikor. Falë kësaj qasjeje, një numër i madh të dhënash mund të përpunohen shpejt, duke përdorur një procedurë të thjeshtë programuese.

Shembull 2: Katrorët e vlerave të matura

Në një qendër kërkimore futen pesë vlera të matura. Llogaritni katrorin e secilës vlerë.



The image shows a C++ program in a code editor and its execution output in a console window. The code defines an array of five numbers, prompts the user to input each number, and then calculates and prints the square of each number using the `pow()` function from the `cmath` library.

```
main.cpp X
1  #include <iostream>
2  #include <cmath>
3
4  using namespace std;
5
6  int main()
7  {
8      double vlerat[5];
9      for (int i = 0; i < 5; i++)
10     {
11         cout << "Vlera " << i + 1 << ": ";
12         cin >> vlerat[i];
13     }
14     cout << endl;
15     for (int i = 0; i < 5; i++)
16     {
17         cout << "Katrori: " << pow(vlerat[i], 2) << endl;
18     }
19     return 0;
20 }
21
```

The console window shows the following output:

```
D:\TD\Cpp\Edyta\bin\Debug\Edyta.exe
Vlera 1: 4
Vlera 2: 2
Vlera 3: 3
Vlera 4: 4
Vlera 5: 6
Katrori: 16
Katrori: 4
Katrori: 9
Katrori: 16
Katrori: 36
Process returned 0 (0x0)
Press any key to continue.
```

Në shembull, çdo element i vargut përpunohet veçmas. Me ndihmën e funksionit `pow()` llogaritet katrori i secilës vlerë të futur.

Gjatë përpunimit praktik të të dhënave shpesh paraqiten edhe vlera negative. Në situata të caktuara, nuk është e rëndësishme shenja e tyre, por vetëm madhësia e vlerës. Për këtë qëllim përdoret funksioni `abs()`, i cili përcakton vlerën absolute të numrit. Pas përpunimit të të gjithë elementeve, mund të kryhet edhe një analizë shtesë e rezultateve të përfituara.

Shembull 3: Vlera absolute më e madhe

Në një laborator shkencor futen pesë matje, të cilat mund të jenë pozitive ose negative. Përcakto vlerën absolute më të madhe.

```

main.cpp x
1  #include <iostream>
2  #include <cmath>
3
4  using namespace std;
5
6  int main()
7  {
8      int matjet[5];
9      int meEMadha;
10     for (int i = 0; i < 5; i++)
11     {
12         cout << "Matja " << i + 1 << ": ";
13         cin >> matjet[i];
14     }
15     meEMadha = abs(matjet[0]);
16     for (int i = 1; i < 5; i++)
17     {
18         if (abs(matjet[i]) > meEMadha)
19         {
20             meEMadha = abs(matjet[i]);
21         }
22     }
23     cout << "Vlera absolute më e madhe: "
24          << meEMadha << endl;
25     return 0;
26 }
  
```

D:\TD\Cpp\Edyta\bin\Debug\Edyta.exe

```

Matja 1: 45
Matja 2: -23
Matja 3: 48
Matja 4: 65
Matja 5: -87
Vlera absolute më e madhe: 87
Process returned 0 (0x0)
Press any key to continue.
  
```

Logs & others

Në shembullin, mbi secilin element fillimisht zbatohet funksioni `abs()`. Më pas vlerat e përfituara krahasohen dhe përcaktohet vlera absolute më e madhe.

PYETJE DHE DETYRA PËR PËRSËRITJE



PYETJE

1. Pse në program duhet të përfshihet biblioteka cmath?
2. Si mund të zbatohet një funksion matematikor mbi të gjithë elementet e një vargu?
3. Pse është e dobishme që funksionet matematikore të kombinohen me vargjet dhe ciklet



DETYRA

1. Në një shërbim gjeodezik futen pesë distanca. Shkruaj një program që do të llogarisë rrënjën katrore të secilës distancë.
2. Në një laborator të fizikës futen gjashtë matje. Shkruaj një program që do të llogarisë katrorin e secilës vlerë.
3. Në një stacion kërkimor futen pesë matje pozitive dhe negative. Shkruaj një program që do të përcaktojë vlerën absolute më të madhe.
4. Në një observator astronomik futen shtatë distanca. Shkruaj një program që do të llogarisë kubin e secilës distancë.
5. Në një laborator inxhinierik futen gjashtë matje. Shkruaj një program që do të llogarisë vlerën absolute të secilës matje dhe do të shfaqë rezultatet e përfutuara.



MBAJ MEND ÇFARË MËSOVE

- Funksionet e bibliotekës cmath mund të zbatohen mbi elementet e një vargu.
- Duke kombinuar vargjet, ciklet dhe funksionet matematikore, mund të zgjidhen detyra të ndryshme praktike.



PËR ATA QË DËSHIROJNË TË DINË MË SHUMË

Në sistemet moderne informatike, një numër i madh të dhënash përpunohen duke përdorur funksione matematikore. Në shkencë mund të analizohen rezultatet e matjeve, në inxhinieri madhësi të ndryshme fizike, ndërsa në sistemet financiare – tregues të ndryshëm statistikorë. Edhe pse programet moderne përpunojnë sasi të mëdha informacioni, ideja themelore mbetet e njëjtë – çdo e dhënë përpunohet me funksionin përkatës matematikor, ndërsa rezultatet e përfutuara më pas mund të analizohen dhe të krahasohen.



ZBATIMI I KOMBINUAR I FUNKSIONEVE DHE VARGJEVE NJËDIMENSIONALE

TASHMË DI ...

- të përdorësh vargje njëdimensionale;
- të futësh dhe përpunosh elementet;
- të përkufizosh dhe thërrisni funksione;
- të përdorësh ciklin for;
- të kombinosh konstrukte të ndryshme programuese.

PYETJE PËR PËRSËRITJE

1. Çfarë përfaqëson argumenti i një funksioni?
2. Çfarë përfaqëson thirrja e një funksioni?
3. A mund të thirret një funksion disa herë brenda një programi?
4. Pse gjatë përpunimit të vargjeve përdoret më së shpeshti një cikël?
5. Çfarë përparësie ka kombinimi i funksioneve dhe vargjeve?

NË PËRMBAJTJET QË VIJOJNË DO TË MËSOSH:

- të zbatosh funksione mbi elementet e një vargu;
- të kombinosh ciklet dhe funksionet;
- të përpunosh të dhënat me funksione të krijuara nga përdoruesi;
- të përdorësh vlerat e kthyer gjatë punës me vargje;
- të zgjidhësh detyra praktike programuese.

Gjatë përpunimit të të dhënave shpeshherë lind nevoja që e njëjta procedurë të kryhet mbi disa elemente të një vargu. Në vend që të njëjtat instruksione programuese të përsëriten disa herë, ato mund të vendosen në një funksion. Më pas funksioni thirret njëri pas tjetrit për secilin element të vargut.

Me kombinimin e funksioneve dhe vargjeve njëdimensionale, programet bëhen më të qarta dhe më të lehta për t'u kuptuar. I njëjti funksion mund të përdoret për përpunimin e të gjithë elementeve, pa pasur nevojë të rishkruhen të njëjtat instruksione programuese.

Gjatë kombinimit të funksioneve dhe vargjeve, zakonisht funksioni përpunon një element, ndërsa cikli mundëson që ai të thirret për të gjithë elementet e vargut. Në këtë mënyrë, çdo vlerë përpunohet në të njëjtën mënyrë, ndërsa programi mbetet i thjeshtë dhe i qartë. Kjo qasje ka zbatim të gjerë në zgjidhjen e detyrave të ndryshme programuese dhe paraqet vazhdim të natyrshëm të të mësuarit për funksionet dhe vargjet njëdimensionale.

Shembull 1: Katrorët e elementeve

Në një laborator kërkimor futen pesë vlera. Shfaqni katrorin e secilës vlerë duke përdorur një funksion.



```
main.cpp x
1  #include <iostream>
2
3  using namespace std;
4
5  int katrori(int x)
6  {
7      return x * x;
8  }
9
10 int main()
11 {
12     int numrat[5];
13     for (int i = 0; i < 5; i++) {
14         cout << "Numri " << i + 1 << ": ";
15         cin >> numrat[i];
16     }
17     cout << endl;
18     for (int i = 0; i < 5; i++) {
19         cout << katrori(numrat[i])
20             << endl;
21     }
22     return 0;
23 }
24
```

D:\TD\Cpp\Edyta\bin\Debug\Edyta.exe

```
Numri 1: 2
Numri 2: 4
Numri 3: 3
Numri 4: 8
Numri 5: 15
4
16
9
64
225
Process returned 0 (0x0)
Press any key to continue.
```

Logs & others

Code::

Executing:

CppCh

Vtora\bi

C/C++

Në shembullin, funksioni përpunon një element, ndërsa cikli mundëson që i njëjti funksion të thirret për të gjithë elementet e vargut.

Gjatë kombinimit të funksioneve dhe vargjeve, nuk është e nevojshme që funksioni të llogarisë vetëm një vlerë matematikore. Në shumë situata është e nevojshme të kontrollohet nëse një element i caktuar plotëson një kusht të dhënë. Në raste të tilla, funksioni mund të kthejë një vlerë logjike. Nëse kushti plotësohet, funksioni kthen true, ndërsa në të kundërtën false. Me ndihmën e ciklit, i njëjti funksion mund të thirret për të gjithë elementet e vargut, ndërsa rezultatet e përfituara mund të përpunohen më tej.

Shembull 2: Kontrollimi i numrave çift

Në një qendër kërkimore futen gjashtë numra. Përcakto sa prej tyre janë çift, duke përdorur një funksion.

```

main.cpp x
1  #include <iostream>
2
3  using namespace std;
4
5  bool eshteCift(int x)
6  {
7      if (x % 2 == 0)
8      {
9          return true;
10     }
11     return false;
12 }
13
14 int main()
15 {
16     int numrat[6];
17     int numeruesi = 0;
18     for (int i = 0; i < 6; i++)
19     {
20         cout << "Numri " << i + 1 << ": ";
21         cin >> numrat[i];
22     }
23     for (int i = 0; i < 6; i++)
24     {
25         if (eshteCift(numrat[i]))
26         {
27             numeruesi++;
28         }
29     }
30     cout << "Numrat çift: " << numeruesi << endl;
31     return 0;
32 }
33

```

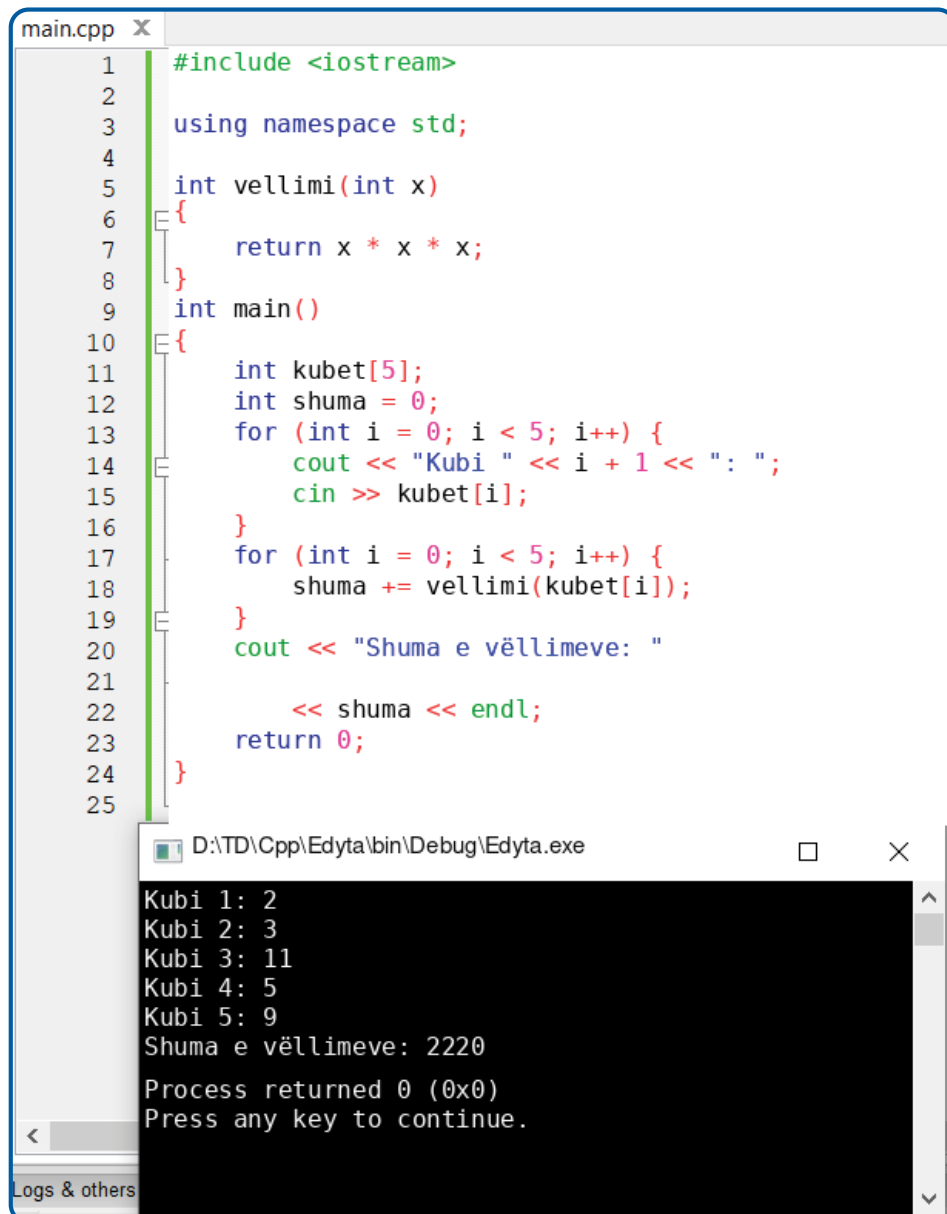
Numri 1: 2
 Numri 2: 7
 Numri 3: 45
 Numri 4: 62
 Numri 5: 188
 Numri 6: 145
 Numrat çift: 3
 Process returned 0 (0x0)
 Press any key to continue

Në shembullin, funksioni kontrollon nëse një numër është çift. Më pas i njëjti funksion thirret për secilin element të vargut, ndërsa numërohen të gjithë numrat që e plotësojnë kushtin.

Gjatë punës me funksione, shpesh lind nevoja që një vlerë e caktuar të llogaritet për secilin element të vargut dhe më pas rezultatet e përfituara të përpunohen më tej. Në këtë mënyrë mund të kombinohen teknika të ndryshme programuese që më parë janë mësuar veçmas. Falë kësaj qasjeje, programet bëhen më të organizuara dhe më të lehta për t'u mirëmbajtur.

Shembull 3: Vëllimi i kubeve dhe shuma e tyre

Në një laborator teknik duhet të optimizohet hapësira përreth 5 objekteve që do të dërgohen në hapësirë. Të gjitha 5 objektet kanë formën e kubit. Janë të nevojshme pesë matje të vëllimit të trupave në formë kubi. Të llogaritet shuma e vëllimeve të të gjitha matjeve, duke përdorur një funksion.



```
main.cpp X
1  #include <iostream>
2
3  using namespace std;
4
5  int vëllimi(int x)
6  {
7      return x * x * x;
8  }
9  int main()
10 {
11     int kubet[5];
12     int shuma = 0;
13     for (int i = 0; i < 5; i++) {
14         cout << "Kubi " << i + 1 << ": ";
15         cin >> kubet[i];
16     }
17     for (int i = 0; i < 5; i++) {
18         shuma += vëllimi(kubet[i]);
19     }
20     cout << "Shuma e vëllimeve: "
21
22         << shuma << endl;
23     return 0;
24 }
25
```

D:\TD\Cpp\Edyta\bin\Debug\Edyta.exe

```
Kubi 1: 2
Kubi 2: 3
Kubi 3: 11
Kubi 4: 5
Kubi 5: 9
Shuma e vëllimeve: 2220
Process returned 0 (0x0)
Press any key to continue.
```

Logs & others

```

main.cpp X
1  #include <iostream>
2
3  using namespace std;
4
5  int kubi(int x)
6  {
7      return x * x * x;
8  }
9  int main()
10 {
11     int matjet[5];
12     int shuma = 0;
13     for (int i = 0; i < 5; i++)
14     {
15         cout << "Matja " << i + 1 << ": ";
16         cin >> matjet[i];
17     }
18     for (int i = 0; i < 5; i++)
19     {
20         shuma += kubi(matjet[i]);
21     }
22     cout << "Shuma e kubeve: " << shuma << endl;
23     return 0;
24 }
25
D:\TD\Cpp\Edyta\bin\Debug\Edyta.exe
Matja 1: 2
Matja 2: 5
Matja 3: 14
Matja 4: 22
Matja 5: 4
Shuma e kubeve: 13589
Process returned 0 (0x0)
Press any key to continue.

```

Në shembullin, funksioni llogarit vëllimin që përfitohet nga vlera e një elementi, ndërsa cikli mundëson që ai të zbatohet mbi të gjithë elementet e vargut. Rezultatet e përfituara mblidhen gradualisht dhe në fund shfaqet shuma e tyre.

PYETJE DHE DETYRA PËR PËRSËRITJE



PYETJE

1. Çfarë detyre mund të kryejë një funksion që përpunon një element të një vargu?
2. Si mundëson cikli që i njëjti funksion të zbatohet mbi disa elemente?
3. Në cilat situata funksioni mund të kthejë një vlerë logjike?
4. Pse është e dobishme që përpunimi i të dhënave të vendoset në një funksion?
5. Cilat janë përparësitë e kombinimit të funksioneve dhe vargjeve njëdimensionale?



DETYRA

1. Në një stacion meteorologjik futen gjashtë temperatura. Shkruaj një program në të cilin do të përdorësh një funksion që llogarit vlerën e trefishtë (të shumëzuar me 3) të temperaturës dhe i shfaq rezultatet.
2. Në një qendër robotike futen pesë rezultate. Shkruaj një program në të cilin do të përdorësh një funksion që kontrollon nëse numri është pozitiv dhe përcakton sa vlera pozitive janë futur.
3. Në një laborator shkencor futen gjashtë matje. Shkruaj një program në të cilin do të përdorësh një funksion që llogarit mbetjen gjatë pjesëtimit me numrin 5 dhe shfaq vlerat e përfituara.
4. Në një qendër sportive futen pesë rezultate. Shkruaj një program në të cilin do të përdorësh një funksion që llogarit gjysmën e secilit rezultat dhe shfaq vlerat e përfituara.
5. Në një qendër energjetike futen gjashtë matje. Shkruaj një program në të cilin do të përdorësh një funksion që përcakton nëse vlera është më e madhe se 100 dhe llogarit numrin e matjeve të tilla.



MBAJ MEND ÇFARË MËSOVE

- Funksioni mund të përpunojë një element të vetëm të vargut.
- I njëjti funksion mund të thirret për të gjithë elementet e vargut.
- Funksioni mund të kthejë një vlerë numerike ose logjike.
- Cikli mundëson që funksioni të zbatohet mbi të gjithë elementet.
- Kombinimi i funksioneve dhe vargjeve mundëson që programet të jenë më të qarta dhe më të lehta për t'u mirëmbajtur.



PËR ATA QË DËSHIROJNË TË DINË MË SHUMË

Në programimin modern, funksionet përfaqësojnë një nga mënyrat më të rëndësishme për organizimin e kodit programues. Në vend që të njëjtat instruksione të përsëriten në vende të ndryshme, ato vendosen në funksione që mund të thirren disa herë. Kur kjo qasje kombinohet me vargje njëdimensionale, mund të përpunohen në mënyrë të thjeshtë dhe efikase sasi të mëdha të të dhënave. Kjo mënyrë pune zbatohet gjatë përpunimit të matjeve shkencore, analizave financiare, të dhënave statistikore dhe në shumë sisteme të tjera moderne informatike.



DETYRA PËR USHTRIME ME VARGJE

Detyra të lehta

1. Në një shtëpi malore, gjatë një jave shënohet numri i vizitorëve. Shkruaj një program që do të llogarisë numrin e përgjithshëm të vizitorëve.
2. Në një fidanishte regjistrohet lartësia e dhjetë fidanëve të rinj. Shkruaj një program që do të përcaktojë lartësinë mesatare.
3. Në një muze regjistrohet numri i vizitorëve gjatë tetë ditëve radhazi. Shkruaj një program që do të përcaktojë ditën me numrin më të madh të vizitorëve.
4. Në një central elektrik me erë regjistrohet prodhimi ditor i energjisë elektrike nga nëntë gjeneratorë. Shkruaj një program që do të përcaktojë cilin gjenerator e ka prodhimin më të vogël.
5. Në një park të kafshëve të egra shënohet numri i këlyshëve të lindur në shtatë habitate të ndryshme. Shkruaj një program që do të përcaktojë në sa habitate kanë lindur më shumë se tre këlyshë.
6. Në një bibliotekë mbërrijnë libra në numër të ndryshëm kopjesh. Në dhjetë dërgesa regjistrohet numri i librave. Shkruaj një program që do të kontrollojë nëse ndonjë dërgesë përmban saktësisht 100 libra.
7. Në një fabrikë lëngjesh shënohet numri i shisheve të prodhuara gjatë tetë orëve. Shkruaj një program që do të përcaktojë diferencën ndërmjet orës me prodhimin më të madh dhe orës me prodhimin më të vogël.
8. Në një park kombëtar shënohet numri i drerëve të vërejtur gjatë nëntë vëzhgimeve. Shkruaj një program që do të përcaktojë në sa vëzhgime janë vërejtur të paktën pesë drerë.
9. Në një kamp astronomik shënohet numri i meteorëve të vërejtur gjatë dhjetë netëve. Shkruaj një program që do të llogarisë numrin e përgjithshëm të meteorëve të vërejtur gjatë netëve me numër çift.
10. Në një akvarium ndiqet numri i peshqve të sapolindur në tetë basene. Shkruaj një program që do të përcaktojë sa basene kanë numër të peshqve të sapolindur më të madh se numri mesatar.

Detyra mesatarisht të vështira

11. Në një minierë regjistrohen sipërfaqet e tetë gërmimeve në formë katrore. Shkruaj një program që do të përcaktojë gjatësinë e brinjës së secilit gërmim.
12. Në një kopsht regjistrohen gjatësitë e brinjëve të shtatë parcelave katrore me lule. Shkruaj një program që do të llogarisë sipërfaqen e secilës parcelë.
13. Në një laborator gjeologjik regjistrohen zhvendosjet pozitive dhe negative të tokës gjatë nëntë matjeve. Shkruaj një program që do të përcaktojë se cila zhvendosje është më e madhe, pa marrë parasysh drejtimin.

- 14.** Në një qendër për përgatitje sportive regjistrohen distancat e përshkuara nga gjashtë sportistë. Duhet të llogaritet katrori i distancës së secilit sportist dhe të shfaqen vlerat e përfituara.
- 15.** Në një studio arkitekture projektohen shatërvanë rrethorë. Njihen rrezet e tetë shatërvanëve. Shkruaj një program që do të llogarisë sipërfaqen e secilit shatërvan.
- 16.** Në një kompani ndërtimi jepen gjatësitë e brinjëve të shtatë kubeve që do të përdoren si elemente dekorative. Shkruaj një program që do të llogarisë vëllimin e secilit kub.
- 17.** Në një observator astronomik regjistrohen devijimet pozitive dhe negative të teleskopit. Shkruaj një program që do të llogarisë madhësinë totale të të gjitha devijimeve, pa marrë parasysh drejtimin e tyre.
- 18.** Në një laborator shkencor jepen sipërfaqet e mostrave katrore. Shkruaj një program që do të përcaktojë se cila mostër ka brinjën më të gjatë.
- 19.** Në një qendër për energji të rinovueshme njihen rrezet e gjashtë platformave rrethore diellore. Shkruaj një program që do të llogarisë sipërfaqen totale të të gjitha platformave.
- 20.** Në një shërbim gjeodezik regjistrohen gabime pozitive dhe negative gjatë matjeve. Shkruaj një program që do të llogarisë madhësinë mesatare të gabimeve, pa marrë parasysh nëse janë pozitive apo negative.

Detyra të vështira

- 21.** Në një qendër bletarie çdo koshere merr një vlerësim për produktivitet. Një koshere konsiderohet e suksesshme nëse prodhon të paktën 25 kg mjaltë. Gjatë sezonit regjistrohet prodhimi i tetë koshereve. Shkruaj një program që do të përcaktojë sa koshere të suksesshme ka.
- 22.** Në një klub alpinistësh regjistrohen lartësitë e dhjetë majave. Një majë konsiderohet e lartë nëse lartësia e saj mbidetare është të paktën 2000 metra. Shkruaj një program që do të përcaktojë sa maja të larta janë regjistruar.
- 23.** Në një akvarium monitorohet numri i peshqve të sapolindur në nëntë basene. Një basen konsiderohet i suksesshëm nëse në të kanë lindur më shumë se 50 peshq. Shkruaj një program që do të përcaktojë numrin e baseneve të suksesshme.
- 24.** Në një observator astronomik regjistrohen distancat e shtatë trupave qiellorë. Një trup qiellor konsiderohet i afërt nëse distanca është më e vogël se një vlerë e dhënë paraprakisht. Shkruaj një program që do të përcaktojë numrin e trupave të afërt.
- 25.** Në një park kombëtar regjistrohet mosha e dhjetë pemëve. Një pemë konsiderohet shekullore nëse mosha e saj është të paktën 100 vjet. Shkruaj një program që do të përcaktojë sa pemë shekullore ka.
- 26.** Në një qendër për shpëtimin e kafshëve regjistrohet masa trupore e tetë kafshëve. Për secilën kafshë duhet të përcaktohet nëse i përket grupit të kafshëve të lehta apo të rënda, sipas një kufiri të dhënë paraprakisht. Shkruaj një program që do të përcaktojë numrin e kafshëve të rënda.
- 27.** Në një kamp sportiv regjistrohen gjatësitë e kërcimeve të nëntë garuesve. Një kërcim konsiderohet i suksesshëm nëse është më i gjatë se kufiri i dhënë. Shkruaj një program që do të përcaktojë sa kërcime të suksesshme janë realizuar.
- 28.** Në një park me turbina me erë regjistrohet prodhimi i shtatë turbinave. Një turbinë konsiderohet efikase nëse prodhon më shumë se sasia e dhënë e energjisë. Shkruaj një program që do të përcaktojë numrin e turbinave efikase.
- 29.** Në një ekspeditë arkeologjike regjistrohet mosha e tetë objekteve të gjetura. Një objekt konsiderohet me rëndësi historike nëse mosha e tij është më e madhe se një vlerë e përcaktuar paraprakisht. Shkruaj një program që do të përcaktojë numrin e objekteve me rëndësi historike.

30.Në një qendër për energji të rinovueshme monitorohet prodhimi ditor i nëntë sistemeve diellore. Një sistem konsiderohet i suksesshëm nëse prodhon më shumë energji se sasia e dhënë. Shkruaj një program që do të përcaktojë sa sisteme të suksesshme ka.

31.Në një klub motoçiklistësh regjistrohen kilometrat e përshkuara nga tetë drejtues. Një drejtues merr statusin e aventurierit nëse ka përshkuar të paktën 300 km. Shkruaj një program në të cilin përpunimi për një drejtues të jetë i veçantë në një funksion, ndërsa programi të përcaktojë sa drejtues kanë fituar statusin e aventurierit.

32.Në një bletore monitorohet sasia e mjaltit nga shtatë koshere. Një koshere konsiderohet produktive nëse prodhon të paktën 25 kg mjaltë. Shkruaj një program në të cilin kontrolli për një koshere të realizohet me anë të një funksioni, ndërsa të dhënat të përpunohen përmes një vargu.

33.Në një mision hapësinor regjistrohen madhësitë e nëntë asteroidëve. Një asteroid konsiderohet i rrezikshëm nëse madhësia e tij është më e madhe se kufiri i dhënë. Shkruaj një program që do t'i përpunojë të gjithë asteroidët në një varg dhe me anë të një funksioni do të përcaktojë nëse një asteroid është i rrezikshëm.

34.Në një ekspeditë arkeologjike futet mosha e tetë objekteve. Një objekt konsiderohet i rëndësishëm nëse është më i vjetër se 1000 vjet. Shkruaj një program në të cilin funksioni do të përcaktojë rëndësinë e një objekti, ndërsa programi do të shfaqë numrin e objekteve të rëndësishme.

35.Në një fabrikë regjistrohet numri i pjesëve të prodhuara nga gjashtë robotë. Një robot konsiderohet efikas nëse ka prodhuar më shumë se 120 pjesë. Shkruaj një program në të cilin një funksion do të kontrollojë efikasitetin e një roboti, ndërsa vargu do të ruajë rezultatet për të gjithë robotët.

36.Në një qendër shpëtimi futet masa trupore e shtatë kafshëve. Një kafshë konsiderohet e rëndë nëse masa e saj është më e madhe se një kufi i dhënë paraprakisht. Shkruaj një program në të cilin një funksion do të kryejë kontrollin për një kafshë, ndërsa programi do të llogarisë sa kafshë janë të rënda.

37.Në një qytet të zgjuar monitorohet konsumi ditor i energjisë elektrike të tetë objekteve. Një objekt konsiderohet konsumator i madh nëse konsumi i tij është mbi 500 njësi. Shkruaj një program në të cilin një funksion do të përcaktojë nëse një objekt është konsumator i madh.

38.Në një kamp sportiv futen rezultatet e nëntë kërcimeve në distancë. Një kërcim konsiderohet i suksesshëm nëse është më i gjatë se kufiri i dhënë. Shkruaj një program në të cilin një funksion do të kontrollojë nëse një kërcim është i suksesshëm, ndërsa programi do të shfaqë numrin e kërcimeve të suksesshme.

39.Një patrullë ekologjike regjistron sasinë e mbetjeve të mbledhura në tetë lokacione. Një lokacion konsiderohet kritik nëse sasia e mbetjeve është më e madhe se kufiri i dhënë. Shkruaj një program në të cilin një funksion do të kontrollojë gjendjen e një lokacioni, ndërsa programi do të përcaktojë sa lokacione kritike ka.

40.Në një mision nënujor futen thellësitë e shtatë zhytjeve. Një zhytje konsiderohet e thellë nëse thellësia është më e madhe se 40 metra. Shkruaj një program në të cilin një funksion do të kontrollojë nëse një zhytje është e thellë, ndërsa vargu do të ruajë të gjitha matjet.

Për temën

Në shoqërinë moderne, çdo ditë krijohen dhe përpunohen sasi të mëdha të dhënash. Të dhënat përdoren në shkolla, banka, spitale, biblioteka, dyqane, institucione shtetërore dhe në shumë fusha të tjera. Për ruajtjen, organizimin dhe përpunimin e tyre në mënyrë efikase përdoren bazat e të dhënave, të cilat mundësojnë gjetjen, përditësimin dhe menaxhimin e shpejtë të informacionit. Falë përdorimit të tyre, puna e organizatave të ndryshme bëhet më efikase, ndërsa qasja në informacionet e nevojshme është më e shpejtë dhe më e sigurt.

Në këtë tërësi tematike trajtohen konceptet themelore që lidhen me bazat e të dhënave dhe sistemet për menaxhimin e bazave të të dhënave. Vëmendje e veçantë i kushtohet organizimit të të dhënave në tabela, regjistrave dhe fushave, krijimit të lidhjeve ndërmjet të dhënave, si dhe kërkimit, futjes dhe përditësimit të informacionit. Përmes aktiviteteve praktike fitohen njohuri dhe aftësi themelore për përdorimin e bazave të të dhënave në fusha të ndryshme të jetës së përditshme dhe në punën bashkëkohore.

KONCEPTE TË REJA

- bazë të dhënash;
- sistem për menaxhimin e bazës së të dhënave (DBMS);
- çelës primar;
- lidhje;
- kërkim;
- përditësim;
- renditje;
- filtrim;
- pyetje (query);
- raport.

TASHMË DI ...

- të futësh dhe të redaktosh të dhëna;
- të organizosh informacionet në tabela;
- të përdorësh programe të ndryshme kompjuterike;
- të kërkosh informacione;
- të analizosh dhe të krahasosh të dhënat;
- të zbatosh teknologjitë digjitale në zgjidhjen e detyrave praktike;
- të përdorësh mjete informatike për përpunimin e informacionit.



TEMA 3





3.1

KONCEPTET THEMELORE TË BAZAVE TË TË DHËNAVE

TASHMË DI ...

- të përdorësh programe të ndryshme kompjuterike;
- të futësh dhe të përpunosh të dhëna;
- të organizosh informacionin në formë tabelare;
- të kërkosh informacione në internet;
- të përdorësh shërbime të ndryshme digjitale;
- të analizosh dhe të krahasosh të dhëna.

PYETJE PËR TË MENDUAR

- Si i ruajnë bankat të dhënat për klientët e tyre?
- Si i mban shkolla të dhënat për nxënësit dhe mësuesit?
- Si e dinë dyqanet online se cilat produkte janë të disponueshme?
- Si mund t'i përdorin mijëra përdorues njëkohësisht të njëjtat informacione?
- Pse është e rëndësishme që të dhënat të gjenden dhe të përditësohen lehtësisht?

NË PËRMBAJTJET QË VIJOJNË, DO TË MËSOSH:

- të shpjegosh se çfarë është një bazë të dhënash;
- të japësh shembuj të përdorimit të bazave të të dhënave;
- të shpjegosh pse përdoren bazat e të dhënave;
- të analizosh përparësitë e përdorimit të tyre;
- të kuptosh rëndësinë e tyre në sistemet moderne informatike.

Shoqëria moderne çdo ditë krijon sasi të mëdha informacioni. Çdo blerje në një dyqan, çdo tërheqje parash nga bankomati, çdo kërkim në internet dhe çdo mesazh i dërguar përmes rrjeteve sociale përbën një të dhënë që duhet të ruhet dhe të përpunohet. Përveç individëve, sasi të mëdha informacioni krijohen edhe nga shkollat, universitetet, bankat, spitalet, institucionet shtetërore dhe shumë organizata të tjera. Për funksionimin e tyre të duhur, është e nevojshme që të dhënat të jenë të disponueshme në çdo moment dhe të mund të gjenden dhe përpunohen shpejt.

Me zhvillimin e teknologjive të informacionit janë rritur ndjeshëm edhe kërkesat për menaxhimin e informacionit. Nuk mjafton që të dhënat vetëm të ruhen, por ato duhet të jenë të sigurt, të sakta dhe lehtësisht të qasshme për përdoruesit që kanë të drejtë t'i përdorin. Përveç kësaj, është e nevojshme që informacionet e reja të mund të shtohen lehtësisht, të dhënat ekzistuese të ndryshohen, ndërsa të dhënat e panevojshme të hiqen. Për këto arsye, sistemet moderne informatike përdorin zgjidhje të posaçme për organizimin dhe menaxhimin e sasive të mëdha të të dhënave.

3.1.1 NGA SKEDARËT TEK SISTEMET MODERNE

Në fazat e para të zhvillimit të sistemeve informatike, të dhënat zakonisht ruheshin në skedarë të veçantë. Çdo program përdorte mënyrën e vet për organizimin e informacionit, ndërsa të dhënat e nevojshme për punë regjistroheshin në dokumente të ndryshme. Kjo mënyrë pune ishte plotësisht e mjaftueshme për sa kohë që sasia e informacionit ishte relativisht e vogël dhe sistemet informatike përdoreshin nga një numër i kufizuar përdoruesish.

Me kalimin e kohës, sasia e të dhënave u rrit ndjeshëm. Të njëjtat informacione filluan të përdoren në disa programe të ndryshme dhe shpesh ishte e nevojshme që i njëjti informacion të futej në disa vende. Për shembull, nëse një përdorues ndryshonte adresën ose numrin e telefonit, ndryshimi duhej të bëhej në të gjithë skedarët ku ndodheshin këto informacione. Nëse ndonjë skedar nuk përditësohej, mund të shfaqeshin vlera të ndryshme për të njëjtën të dhënë, gjë që shkaktonte gabime dhe pasiguri në punë.

Problem përbënte edhe kërkimi i informacionit. Me rritjen e numrit të të dhënave, gjetja e tyre kërkonte gjithnjë e më shumë kohë dhe burime. Për më tepër, përsëritja e shpeshtë e të njëjtit informacion zinte në mënyrë të panevojshme hapësirë ruajtëse dhe e vështirësonte përpunimin e tij. Mirëmbajtja e sistemeve të tilla bëhej gjithnjë e më e ndërlikuar dhe më e kushtueshme.

Zhvillimi i teknologjive të informacionit krijoi nevojën për një mënyrë tjetër të organizimit të të dhënave. Në vend që të shpërndaheshin në një numër të madh skedarësh të pavarur, të dhënat filluan të organizoheshin në sisteme të integruara, të cilat mundësojnë ruajtjen e centralizuar, përditësimin më të lehtë dhe kërkimin më të shpejtë të informacionit. Kjo qasje përbën bazën e zhvillimit të bazave moderne të të dhënave.



3.1.2 ÇFARË PËRFAQËSON BAZA E TË DHËNAVE?

Një bazë të dhënash përfaqëson një koleksion të organizuar të të dhënave, i cili mundëson ruajtjen e sigurt, kërkimin e lehtë dhe përpunimin efikas të tyre. Detyra e saj themelore është që informacioni të jetë i disponueshëm atëherë kur nevojitet dhe të mundësojë shtimin, ndryshimin dhe përdorimin e tij në mënyrë të thjeshtë. Ndryshe nga ruajtja klasike e të dhënave në skedarë të pavarur, bazat e të dhënave mundësojnë organizimin e informacionit në një mënyrë që përmirëson saktësinë dhe qasshmërinë e tij.

Një nga përparësitë më të rëndësishme të bazave të të dhënave është mundësia që të njëjtat informacione të përdoren nga shumë përdorues dhe sisteme të ndryshme informatike. Nëse shtohet një e dhënë ose ndryshohet një informacion ekzistues, ndryshimi bëhet menjëherë i disponueshëm për të gjithë përdoruesit që kanë të drejtë qasjeje. Në këtë mënyrë shmanget përsëritja e panevojshme e të dhënave dhe zvogëlohet mundësia e shfaqjes së gabimeve.

Bazat e të dhënave nuk përdoren vetëm për ruajtjen e informacionit. Ato mundësojnë edhe kërkimin, krahasimin, renditjen dhe analizimin e të dhënave. Falë këtyre mundësive, sasi të mëdha të dhënash mund të përpunohen shpejt dhe të përdoren për marrjen e vendimeve të ndryshme. Për këtë arsye, bazat e të dhënave përbëjnë një nga komponentët më të rëndësishëm të sistemeve moderne informatike.

3.1.3 PËRDORIMI I BAZAVE TË TË DHËNAVE

Bazat e të dhënave sot përdoren pothuajse në të gjitha fushat e veprimtarisë njerëzore. Në arsim, ato përdoren për mbajtjen e evidencës për nxënësit, mësuesit, lëndët dhe aktivitetet mësimore. Në institucionet shëndetësore përdoren për organizimin e informacionit mjekësor dhe të dhënave për pacientët, ndërsa në banka mundësojnë menaxhimin e sigurt të shërbimeve financiare dhe transaksioneve.

Bazat e të dhënave kanë një përdorim të gjerë edhe në sistemet moderne informatike. Ato përdoren në dyqanet online, rrjetet sociale, sistemet e postës elektronike, platformat për mësim në distancë dhe shumë shërbime të tjera digjitale. Sa herë që një përdorues hyn në një faqe interneti, kërkon informacion ose kryen një pagesë elektronike, në prapavijë zakonisht funksionon një bazë të dhënash.

Zhvillimi i teknologjive cloud, aplikacioneve mobile dhe inteligjencës artificiale e rrit edhe më tej rëndësinë e bazave të të dhënave. Sistemet moderne informatike përpunojnë miliarda të dhëna në sekondë, prandaj siguria, saktësia dhe shpejtësia e përpunimit të tyre përbëjnë faktorë kyç për funksionimin e suksesshëm të shërbimeve digjitale që përdorim çdo ditë.



3.1.4 PËRPARËSITË E BAZAVE TË TË DHËNAVE

Përdorimi i bazave të të dhënave ofron përparësi të shumta krahasuar me mënyrën klasike të ruajtjes së informacionit. Të dhënat organizohen në mënyrë sistematike, gjë që mundëson gjetjen më të shpejtë dhe përditësimin më të lehtë të tyre. Në të njëjtën kohë, zvogëlohet mundësia e shfaqjes së gabimeve dhe përsëritjes së panevojshme të informacionit.

Një përparësi tjetër e rëndësishme është mundësia që disa përdorues të punojnë njëkohësisht me të njëjtat të dhëna. Ndërsa disa përdorues fusin informacione të reja, të tjerë mund të kërkojnë ose të analizojnë të dhënat ekzistuese. Kjo është veçanërisht e rëndësishme në sistemet moderne informatike që u shërbejnë një numri të madh përdoruesish.

Bazat e të dhënave mundësojnë gjithashtu siguri më të madhe të informacionit. Qasja në të dhëna mund të kufizohet sipas nevojave dhe të drejtave të përdoruesve, ndërsa sistemet moderne ofrojnë mekanizma për mbrojtjen dhe ruajtjen e informacionit. Falë këtyre veçorive, bazat e të dhënave përbëjnë një pjesë të pazëvendësueshme të teknologjive moderne të informacionit dhe për këtë arsye përdorimi i tyre vazhdimisht zgjerohet.

Të shqyrtojmë rastin e një shkolle të mesme, në të cilën çdo vit shkollor regjistrohen nxënës të rinj. Përveç të dhënave personale të nxënësve, është e nevojshme të mbahen të dhëna për klasat, mësuesit, lëndët mësimore, orarin e mësimi, suksesin dhe shumë informacione të tjera. Gjatë vitit disa të dhëna ndryshojnë, regjistrohen nxënës të rinj, ndërsa të tjerë përfundojnë arsimin. Këto të dhëna të regjistruara duhet të jenë të disponueshme edhe në të ardhmen. Çdo person që e ka përfunduar arsimin, pas një kohe, duhet ta paraqesë këtë të dhënë në sisteme ose kompani të ardhshme. Nëse humbet ndonjë dokument, ai duhet të mund të lëshohet përsëri, pavarësisht se kur është përfunduar arsimimi.



Të supozojmë se të gjitha këto informacione ruhen në një numër të madh dokumentesh të ndryshme. Nëse është e nevojshme të gjendet numri i telefonit të një nxënësi, lista e nxënësve në një klasë të caktuar ose të dhënat për një mësues të caktuar, do të ishte e nevojshme të kërkohej nëpër disa dokumente të ndryshme. Përveç kësaj, çdo ndryshim do të duhej të bëhej në disa vende, duke rritur kështu mundësinë e shfaqjes së gabimeve.

Me përdorimin e një baze të të dhënave, të gjitha informacionet mund të organizohen në një sistem të vetëm. Të dhënat e reja futen lehtësisht, informacionet ekzistuese përditësohen, ndërsa të dhënat e nevojshme mund të gjenden shpejt. Në këtë mënyrë përmirësohet ndjeshëm organizimi i punës dhe zvogëlohet koha e nevojshme për përpunimin e informacionit.

E njëjta logjikë zbatohet edhe në shumë fusha të tjera. Spitalet mbajnë evidencë për pacientët, bankat për klientët dhe transaksionet e tyre, bibliotekat për librat, ndërsa shërbimet moderne online për miliona llogari përdoruesish dhe përmbajtje të ndryshme digjitale. Pa bazat e të dhënave, funksionimi i këtyre sistemeve do të ishte shumë më i vështirë.

Bazat e të dhënave mundësojnë që informacioni të centralizohet dhe të jetë i disponueshëm për përdoruesit që kanë nevojë për të. Ndryshimet mund të kryhen lehtësisht, ndërsa të dhënat mund të gjenden dhe të përpunohen shpejt. Falë këtyre mundësive, bazat e të dhënave përbëjnë një pjesë themelore të sistemeve moderne informatike. Ruajtja e organizuar e informacionit nuk kontribuon vetëm në përsheptimin e punës, por edhe në rritjen e sigurisë dhe saktësisë së të dhënave. Teknologjitë moderne të informacionit mbështeten në sisteme të tilla për të siguruar funksionimin e pandërprerë të shërbimeve të ndryshme digjitale që përdoren çdo ditë.

Zhvillimi i teknologjive të informacionit ka çuar në krijimin dhe përpunimin e sasive jashtëzakonisht të mëdha të të dhënave. Mënyra klasike e ruajtjes së tyre në skedarë të pavarur, me kalimin e kohës, ka shfaqur shumë mangësi, si përsëritja e informacionit, përditësimi i ndërlikuar dhe vështirësitë gjatë kërkimit të të dhënave. Për këtë arsye, lindi nevoja për një mënyrë më efikase të organizimit dhe menaxhimit të tyre.

Bazat e të dhënave përfaqësojnë një zgjidhje moderne për ruajtjen dhe përpunimin e sigurt të informacionit. Ato mundësojnë që të dhënat të shtohen dhe të ndryshohen lehtësisht, si dhe të përdoren nga shumë përdorues, duke përmirësuar saktësinë dhe efikasitetin e punës. Falë këtyre veçorive, bazat e të dhënave sot kanë përdorim të gjerë në fusha të ndryshme të shoqërisë moderne. Përvetësimi i njohurive themelore për bazat e të dhënave përbën një pjesë të rëndësishme të shkrim-leximit informatik. Kuptimi i rëndësisë së tyre dhe i mënyrave të përdorimit përbën bazën për studimin e sistemeve moderne informatike dhe zbatimin e tyre praktik.





A E DI?

A e dini se çdo ditë në botë krijohen sasi të mëdha të dhënash të reja? Një pjesë e madhe e tyre krijohet nga përdorimi i telefonave celularë, shërbimeve të internetit, rrjeteve sociale dhe pajisjeve të ndryshme digjitale. Për organizimin dhe përpunimin e tyre përdoren baza moderne të të dhënave, të cilat mund të ruajnë dhe të menaxhojnë një numër jashtëzakonisht të madh informacioni.



MBAJ MEND ÇFARË MËSOVE

Baza e të dhënave përfaqëson një koleksion të organizuar të të dhënave.

Sistemet moderne informatike përpunojnë sasi të mëdha informacioni.

Ruajtja klasike e të dhënave në skedarë të pavarur ka disa mangësi.

Bazat e të dhënave mundësojnë ruajtjen, kërkimin dhe përditësimin më të lehtë të informacionit.

Ato kanë përdorim të gjerë në fusha të ndryshme të shoqërisë moderne.

Bazat e të dhënave përbëjnë një pjesë të rëndësishme të teknologjive moderne të informacionit.



PËR ATA QË DËSHIROJNË TË DINË MË SHUMË

Sistemet moderne informatike mund të përpunojnë sasi jashtëzakonisht të mëdha të të dhënave. Kompanitë e mëdha të internetit, qendrat shkencore dhe platformat e ndryshme digjitale çdo ditë punojnë me miliarda të dhëna që krijohen dhe përditësohen vazhdimisht. Menaxhimi i këtyre të dhënave përbën një fushë të veçantë në informatikë, ndërsa zhvillimi i teknologjive të reja mundëson vazhdimisht përpunim më të shpejtë dhe më efikas të informacionit.



KONTROLLO NJOHURITË DI

1. Përgjigju pyetjeve.

- Çfarë paraqet baza e të dhënave?
- Pse përdoren bazat e të dhënave?
- Përmend tri fusha në të cilat përdoren bazat e të dhënave.

2. E saktë apo e pasaktë.

- Bazat e të dhënave shërbejnë vetëm për ruajtjen e informacionit.
- Të dhënat në bazën e të dhënave mund të përditësohen.
- Sistemet moderne informatike nuk përdorin baza të të dhënave.

3. Mendo dhe arsyeto.

Cilat do të ishin problemet më të mëdha nëse një bankë, spital ose shkollë nuk do të përdorte bazë të të dhënave për organizimin e informacioneve të tyre?

3.2



KRIJIMI I TABELAVE NË BAZËN E TË DHËNAVE

TASHMË DI ...

- të shpjegosh se çfarë paraqet një bazë të të dhënave;
- të japësh shembuj të përdorimit të bazave të të dhënave;
- të shpjegosh pse përdoren bazat e të dhënave;
- të analizosh përparësitë e përdorimit të tyre;
- të përdorësh tabela për organizimin e informacionit.

PYETJE PËR TË MENDUAR

- Si mund të organizohet më lehtë një numër i madh informacioni?
- Pse të dhënat duhet të jenë të renditura në mënyrë sistematike?
- Si organizohet informacioni në një bazë të të dhënave?
- Çfarë ndodh nëse të dhënat nuk organizohen në mënyrë të qartë dhe të rregullt?
- A duhet të vendosen të gjitha informacionet në një tabelë?

NË PËRMBAJTIET QË VIJOJNË, DO TË MËSOSH:

- të shpjegosh se çfarë përfaqëson një bazë të dhënash relacionale;
- të dallosh tabelën, fushën dhe rekordin;
- të analizosh organizimin e të dhënave në tabela;
- të planifikosh strukturën e një table të thjeshtë;
- të përdorësh programe për punë me baza të të dhënave;
- të krijosh një tabelë të thjeshtë.

Në jetën e përditshme, informacionet shpesh organizohen në tabela. Shkollat mbajnë tabela për nxënësit, bibliotekat për librat, klubet sportive për garat, ndërsa institucione të ndryshme për përdoruesit dhe shërbimet e tyre. Paraqitja tabelare mundëson që informacionet të jenë të qarta dhe të organizuara, të krahasohen lehtësisht dhe të gjenden shpejt të dhënat e nevojshme.

E njëjta ide zbatohet edhe në bazat e të dhënave. Edhe pse sistemet moderne informatike përpunojnë sasi jashtëzakonisht të mëdha informacioni, organizimi i tyre më së shpeshti fillon me tabela. Falë kësaj mënyre të punës, të dhënat mund të futen në mënyrë sistematike, të përditësohen dhe të përdoren për qëllime të ndryshme.

Është interesante që paraqitja tabelare përfaqëson një lidhje ndërmjet punës së përditshme dhe bazave të të dhënave. Një numër i madh programesh mundësojnë që të dhënat të eksportohen në formë tabelare, ndërsa tabelat e krijuara në programe të ndryshme mund të futen në një bazë të të dhënave.

Për shembull, të dhënat e organizuara në një tabelë mund të importohen në një bazë të të dhënave, ndërsa të dhënat nga baza mund të eksportohen përsëri për përpunim dhe analizë të mëtejshme. Kjo mundësi e lehtëson ndjeshëm shkëmbimin e informacionit ndërmjet sistemeve të ndryshme informatike.

3.2.1 TABELA SI MËNYRË PËR ORGANIZIMIN E TË DHËNAVE

Kur është e nevojshme të ruhen një numër i madh informacioni, ai duhet të organizohet në mënyrë logjike. Një nga mënyrat më të thjeshta dhe më të qarta për organizimin e tij është paraqitja në formë table. Në tabelë, të dhënat renditen në mënyrë të tillë që të mund të futen, të shikohen dhe të krahasohen lehtësisht.

Të supozojmë se është e nevojshme të mbahet evidencë për kompjuterët në një shkollë. Për çdo kompjuter duhet të dihen disa informacione, si numri i inventarit, prodhuesi, modeli, sistemi operativ dhe dhoma në të cilën përdoret. Organizimi i këtyre informacioneve në një tabelë mundëson që çdo e dhënë të ketë vendin e vet dhe e lehtëson ndjeshëm përdorimin e tyre.

Në bazat moderne të të dhënave, tabelat përfaqësojnë mënyrën themelore të organizimit të informacionit. Ato mundësojnë që të dhënat të sistemohen dhe të jenë lehtësisht të qasshme, duke krijuar njëkohësisht bazë të mirë për përpunimin dhe analizimin e mëtejshëm të tyre.

Shembull praktik

Le të shqyrtojmë tabelën e mëposhtme.

Numri i inventarit	Prodhuesi	Modeli	Sistemi operativ	Lokacioni
1001	Lenovo	ThinkCentre	Windows 11	Kabinet 1
1002	Dell	OptiPlex	Windows 11	Kabinet 2
1003	HP	ProDesk	Linux	Kabinet 3
1004	Lenovo	ThinkCentre	Windows 11	Kabinet 4

Nga tabela mund të vërehet lehtësisht se çdo kompjuter përshkruhet përmes disa informacioneve të ndryshme. Nëse blihet një kompjuter i ri, të dhënat për të mund të shtohen thjesht. Nëse një kompjuter zhvendoset në një dhomë tjetër, duhet të ndryshohet vetëm informacioni përkatës.

Kjo mënyrë e organizimit mundëson që të dhënat të shqyrtohen dhe të krahasohen lehtësisht. Për shembull, mund të përcaktohet shpejt se sa kompjuterë përdorin një sistem operativ të caktuar ose cilat pajisje ndodhen në një dhomë të caktuar.

Si ta analizojmë tabelën?

Me shqyrtim të kujdesshëm të shembullit të mëparshëm mund të vërehet se tabela përbëhet nga kolona dhe rreshta. Çdo kolonë përmban një lloj të caktuar informacioni, ndërsa çdo rresht paraqet një përshkrim të plotë të një kompjuteri. Falë kësaj mënyre të organizimit, të dhënat janë të qarta dhe lehtësisht të kuptueshme.

Në bazat e të dhënave, kolonat quhen fusha, sepse secila prej tyre është e përcaktuar për një lloj të caktuar informacioni. Rreshtat quhen rekorde dhe përmbajnë të dhëna për një objekt konkret. Kjo mënyrë e organizimit përbën bazën për rregullimin e të dhënave në bazën e të dhënave.

Organizimi i mirë i informacionit mundëson futjen më të shpejtë të të dhënave, mirëmbajtjen më të lehtë dhe përdorimin më efikas të tyre. Për këtë arsye, krijimi i saktë i tabelave përbën një nga hapat më të rëndësishëm gjatë krijimit të një baze të të dhënave.

3.2.2 BAZA RELACIONALE E TË DHËNAVE

Në praktikë, shumë rrallë të gjitha të dhënat ruhen në një tabelë të vetme. Nëse në një tabelë do të regjistroheshin të gjitha informacionet për nxënësit, mësuesit, lëndët, notat, orarin dhe mungesat, ajo do të bëhej shumë e madhe, e ndërlikuar për mirëmbajtje dhe e prirur ndaj gabimeve. Prandaj, bazat moderne të të dhënave i organizojnë informacionet në disa tabela të ndërlidhura me njëra-tjetrën. Kjo mënyrë organizimi quhet model relacional, ndërsa bazat që e përdorin atë quhen baza relacionale të të dhënave.

Ideja themelore e modelit relacional është që çdo tabelë të përmbajë të dhëna për një tërësi të caktuar. Për shembull, në një sistem informatik shkollor mund të ekzistojë një tabelë e veçantë për nxënësit, një tabelë e veçantë për mësuesit, një tabelë për lëndët dhe një tabelë për notat. Në vend që të njëjtat të dhëna të përsëriten në disa vende, tabelat lidhen me njëra-tjetrën përmes fushave të përbashkëta. Në këtë mënyrë zvogëlohet mundësia e gabimeve, ndërsa të dhënat ruhen në mënyrë më të organizuar dhe më efikase.

Bazat relacionale të të dhënave sot janë lloji më i përhapur i bazave të të dhënave. Ato përdoren në institucionet arsimore, banka, institucione shëndetësore, institucione shtetërore, dyqane online dhe shumë sisteme të tjera informatike. Falë strukturës së tyre, sasi të mëdha të dhënash mund të futen, kërkohen, përditësohen dhe analizohen shpejt.

Shembull i organizimit të të dhënave në disa tabela

Le të supozojmë se është e nevojshme të mbahet evidencë për nxënësit dhe për notat e tyre.

Në vend që të gjitha informacionet të ruhen në një tabelë të vetme, mund të krijohen dy tabela.

Tabela: Nxënësit

ID	Emri	Mbiemri
1	Ana	Petrova
2	Marko	Stojanov
3	Elena	Nikolovska

Tabela: Notat

ID	NxënësiID	Lënda	Nota
1	1	Informatika	5
2	1	Matematika	4
3	2	Informatika	5
4	3	Matematika	5

Në tabelën e parë ruhen të dhënat themelore për nxënësit, ndërsa në të dytën ruhen notat e tyre. Në vend që emri dhe mbiemri i nxënësit të përsëriten çdo herë, përdoret një numër identifikues që mundëson lidhjen e dy tabelave. Në këtë mënyrë zvogëlohet përsëritja e të dhënave dhe përmirësohet organizimi i informacionit.

Pse përdoren disa tabela?

Një nga përparësitë më të rëndësishme të bazave relacionale të të dhënave është zvogëlimi i të dhënave të dyfishuara. Nëse të dhënat për nxënësin ruhen vetëm një herë, çdo ndryshim do të duhet të bëhet në një vend të vetëm. Nëse nxënësi e ndryshon mbiemrin ose nëse është e nevojshme të korrigjohet ndonjë e dhënë, ndryshimi do të jetë i thjeshtë dhe nuk do të ekzistojë rreziku që një pjesë e regjistrimeve të mbeten të pandryshuara.

Organizimi i të dhënave në disa tabela mundëson edhe kërkim më të shpejtë të informacionit. Në vend që programi të kërkojë në një tabelë të madhe me mijëra kolona dhe rreshta, ai punon me tabela më të vogla dhe më të organizuara. Kjo është veçanërisht e rëndësishme në sistemet moderne informatike që përpunojnë miliona rekorde.

Përveç kësaj, kjo mënyrë e organizimit mundëson zgjerimin më të lehtë të bazës së të dhënave. Nëse paraqitet nevoja për informacione të reja, mund të shtohet një tabelë e re pa ndryshuar ndjeshëm strukturën ekzistuese. Për këtë arsye, bazat relacionale të të dhënave përfaqësojnë një standard në sistemet moderne informatike.

3.2.3 PROGRAME PËR PUNË ME BAZA TË TË DHËNAVE

Për krijimin, organizimin dhe përpunimin e të dhënave përdoren programe të specializuara, të njohura si sisteme për menaxhimin e bazave të të dhënave. Këto programe ofrojnë mjete për futjen, redaktimin, kërkimin dhe paraqitjen e të dhënave, pa pasur nevojë që përdoruesi të merret drejtpërdrejt me mënyrën se si informacionet ruhen fizikisht në kompjuter.

Në arsim dhe gjatë mësimi të bazave të të dhënave përdoren shpesh programet Microsoft Access dhe LibreOffice Base. Ato ofrojnë një mjedis grafik pune ku përdoruesi mund të krijojë tabela, të fusë të dhëna, të vendosë lidhje ndërmjet tabelave dhe të krijojë formularë, pyetësorë dhe raporte. Falë qasjes vizuale, këto programe janë të përshtatshme për të mësuar parimet themelore të punës me baza të të dhënave.

Megjithëse ekzistojnë edhe shumë sisteme të tjera për menaxhimin e bazave të të dhënave që përdoren në kompani dhe institucione të mëdha, parimet e punës janë të ngjashme. Pavarësisht nëse përdoret një mjet i thjeshtë arsimor apo një sistem kompleks korporativ, të dhënat organizohen në tabela, vendosen lidhje ndërmjet tyre dhe mundësohet kërkimi dhe përpunimi i tyre.

Është e rëndësishme të kuptohet se programi është një mjet për të punuar me bazën e të dhënave dhe nuk është vetë baza e të dhënave. Baza e të dhënave është një koleksion i organizuar informacioni, ndërsa programi ofron mjetet përmes të cilave përdoruesi punon me këto informacione.

Krijimi i tabelave

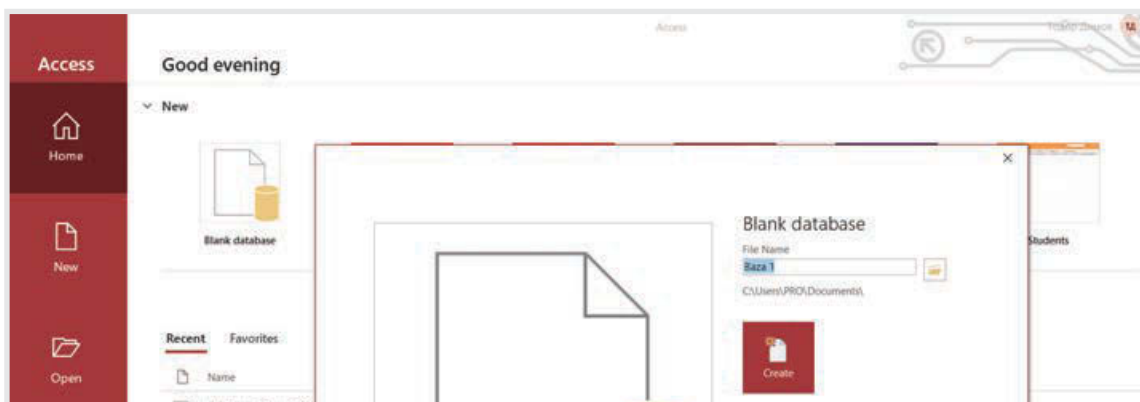
Para se të fillojë futja e të dhënave, është e nevojshme të planifikohet me kujdes se cilat informacione do të ruhen në bazën e të dhënave. Ky hap është i rëndësishëm, sepse cilësia e organizimit ndikon drejtpërdrejt në efikasitetin e punës. Nëse tabela është konceptuar keq, më vonë mund të shfaqen probleme gjatë futjes, kërkimit dhe përditësimit të të dhënave.

Gjatë krijimit të një tabele, fillimisht përcaktohen fushat që do të përmbajë ajo. Çdo fushë është e destinuar për një lloj të caktuar informacioni. Për shembull, nëse krijohet një tabelë për nxënësit, duhet të përcaktohen fushat për emrin, mbiemrin, datën e lindjes, adresën, numrin e telefonit dhe të dhëna të tjera të nevojshme.

Zgjedhja e mirë e fushave mundëson që të dhënat të jenë të organizuara, të qarta dhe të lehta për t'u përpunuar. Për këtë arsye, krijimi i një tabele nuk është vetëm një procedurë teknike, por edhe një proces i planifikimit të kujdesshëm të informacionit që do të ruhet në bazën e të dhënave.

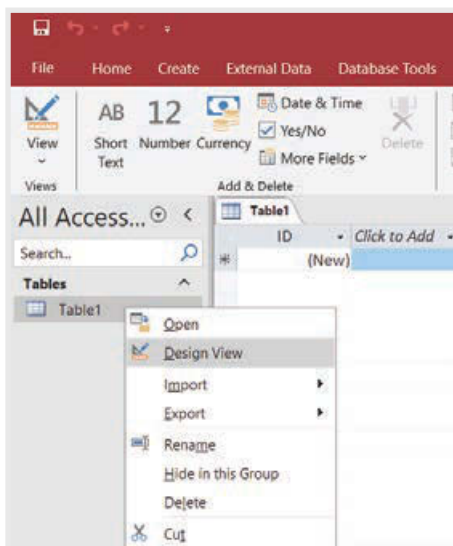
Shembull: Planifikimi i një tabele për nxënësit

Në MS Access, pasi të nisë programi, është e nevojshme të zgjidhet një bazë e zbrazët e të dhënave. Bazës së re i caktohet një emër dhe zgjidhet vendndodhja ku do të ruhet e gjithë baza e të dhënave.



Le të supozojmë se shkolla dëshiron të mbajë evidencë elektronike për nxënësit.

Para se të krijohet tabela, është e nevojshme të mendohet se cilat informacione do të nevojiten për secilin nxënë. Mund të përcaktohet se duhet të ruhen emri, mbiemri, data e lindjes, adresa, numri i telefonit dhe adresa elektronike.



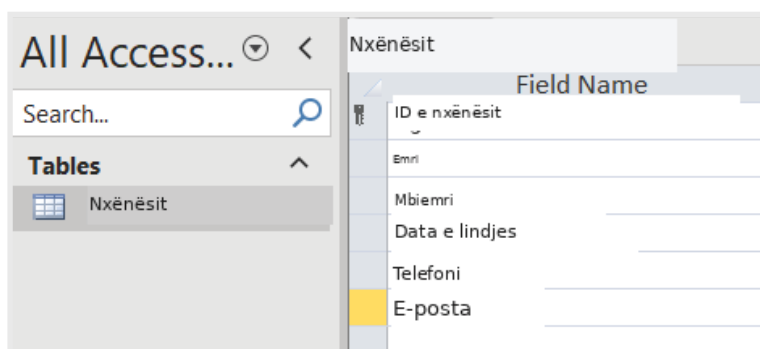
Për përcaktimin e fushave me elementet e nevojshme, qasja bëhet përmes pamjes së dizajnit të tabelës.

Menjëherë pas përzgjedhjes, tabela ruhet me emrin që i caktohet nga përdoruesi. Rekomandohet që emrat e tabelave të jenë përshkruues dhe të tregojnë llojin e të dhënave që do të përmbajë tabela.



Bazuar në këtë analizë, mund të planifikohet struktura e mëposhtme:

ID i nxënësit, Emri,	Mbiemri,	Data e lindjes,	Adresa,	Telefon,	E-mail



Nga shembullin mund të vërehet se çdo kolonë përfaqëson një lloj të caktuar të dhënash që do të futet për të gjithë nxënësit. Ky planifikim përbën hapin e parë në procesin e krijimit të një table.



Analiza e shembullit

Gjatë krijimit të një tablele, nuk mjafton vetëm të përcaktohen kolonat dhe t'u caktohen emra të ndryshëm. Është e nevojshme të përcaktohet me kujdes se cilat informacione janë vërtet të nevojshme. Shtimi i fushave të panevojshme e bën tabelën më të ndërlikuar dhe më të vështirë për mirëmbajtje. Nga ana tjetër, nëse mungojnë fusha të rëndësishme, më vonë do të jetë e nevojshme të bëhen ndryshime shtesë në strukturën e tabelës.

Planifikimi i tabelave përbën një nga hapat më të rëndësishëm gjatë projektimit të një baze të të dhënave. Në praktikë, para fillimit të zhvillimit të sistemit, zakonisht kryhet një analizë e detajuar e informacionit që duhet të ruhet, si dhe e mënyrës se si ai informacion do të përdoret. Në këtë mënyrë sigurohet një organizim më efikas dhe një punë më e lehtë me të dhënat.



PËRFUNDIM:

Tabelat përbëjnë mënyrën themelore të organizimit të të dhënave në bazat relacionale të të dhënave. Nëpërmjet tyre, informacionet paraqiten në mënyrë të strukturuar dhe të qartë, duke mundësuar futjen, përpunimin dhe menaxhimin e tyre në mënyrë më të lehtë dhe më efikase. Organizimi i duhur i tabelave përbën hapin e parë drejt krijimit të një baze të dhënash cilësore dhe funksionale.

Çdo tabelë përmban fusha dhe regjistrime që mundësojnë organizimin sistematik të informacionit. Falë kësaj strukture, të dhënat mund të kërkohen, të përditësohen dhe të përdoren më lehtë në sisteme të ndryshme informatike.



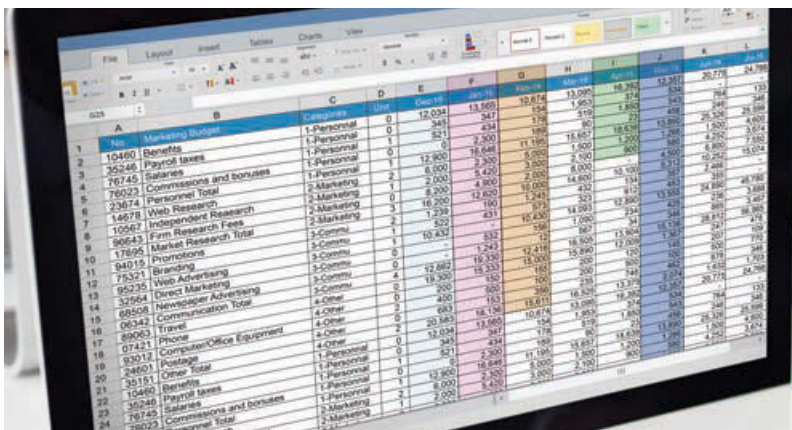
AEDI?

A e dini se pjesa më e madhe e të dhënave që përdorim në jetën e përditshme organizohet në tabela? Qoftë fjala për evidencën shkollore, sistemet bankare apo dyqanet online, baza e organizimit të informacionit, në shumicën e rasteve, është paraqitja tabelare e të dhënave.



PYETJE PËR PËRSËRITJE

1. Çfarë paraqet një bazë relacionale të të dhënave?
2. Çfarë paraqet një tabelë?
3. Cili është roli i fushave në një tabelë?
4. Çfarë paraqet një rekord?
5. Pse organizohen të dhënat në disa tabela?
6. Cilat programe mund të përdoren për të punuar me bazat e të dhënave?
7. Pse është i rëndësishëm planifikimi i tabelës para krijimit të saj?





DETYRA PRAKTIKE

Detyra 1

Harto një plan për një tabelë në të cilën do të ruhen të dhëna për librat në një bibliotekë.

Detyra 2

Përcakto cilat nga informacionet e mëposhtme përfaqësojnë fusha:

- Emri i nxënësit
- Marko Petrov
- Telefoni
- Kabineti 12
- Data e lindjes

Detyra 3

Në tabelën e mëposhtme Shpjego sa fusha dhe sa rekorde ka.

Emri	Mbiemri	Paralelja
Ana	Petrova	I-1
Marko	Stojanov	I-2
Elena	Nikolovska	I-1

Detyra 4

Shpjego pse nuk është ide e mirë që të gjitha të dhënat e një shkolle të ruhen në një tabelë të vetme.



DETYRË HULUMTUESE

Hulumto se cilat programe për punë me baza të të dhënave përdoren më së shpeshti sot. Për secilin program, gjeje:

- emrin;
- prodhuesin;
- nëse është falas apo komercial;
- fushën e përdorimit.



MBAJ MEND ÇFARË MËSOVE

Tabela është element themelor i bazës relacionale të të dhënave.

Fusha përfaqëson kolonën në tabelë.

Rekordi përfaqëson rreshtin në tabelë.

Bazat relacionale të të dhënave përmbajnë disa tabela të ndërlidhura me njëra-tjetrën.

Të dhënat mund të importohen dhe të eksportohen në formë tabelare.

Një tabelë e planifikuar mirë mundëson punë më të lehtë me të dhënat.



PËR ATA QË DËSHIROJNË TË DINË MË SHUMË

Në kompanitë dhe institucionet e mëdha, bazat e të dhënave përmbajnë mijëra tabela dhe miliona rekorde. Megjithatë, falë organizimit të mirë të tabelave, informacioni i nevojshëm mund të gjendet brenda pak sekondash.



KONTROLLO NJOHURITË E TUA

Rretho përgjigjen e saktë

Fusha përfaqëson:

- rresht
- kolonë
- tabelë
- bazë të të dhënave

Rekordi përfaqëson:

- kolonë
- fushë
- rresht
- bazë të të dhënave

E saktë apo e pasaktë

- Një bazë të dhënash mund të përmbajë disa tabela.
- Fusha dhe rekordi janë terma me të njëjtin kuptim.
- Tabelat përdoren për organizimin e të dhënave.



3.3

LLOJET E TË DHËNAVE

TASHMË DI ...

- të shpjegosh se çfarë përfaqëson një tabelë;
- të dallos fushën nga rekordi;
- të planifikosh strukturën e një table të thjeshtë;
- të krijosh një tabelë në një bazë të të dhënave;
- të organizosh informacionin në formë tabelare.

PYETJE PËR TË MENDUAR

- A mund të futen në të gjitha fushat të njëjtat lloje të të dhënave?
- Si e përcakton kompjuteri nëse një vlerë e caktuar është tekst, numër apo datë?
- Pse është e rëndësishme të ekzistojnë rregulla për futjen e të dhënave?
- Çfarë mund të ndodhë nëse përdoruesi fut një vlerë të pasaktë?
- Si mund të përmirësohet saktësia e të dhënave në bazën e të dhënave?

NË PËRMBAJTJET NË VIJIM DO TË MËSOSH:

- të dallos lloje të ndryshme të të dhënave;
- të zgjidhësh llojin e përshtatshëm të të dhënave;
- të shpjegosh vetitë e fushave;
- të zbatosh rregullat e validimit;
- të analizosh mënyrat e kontrollit të të dhënave të futura;
- të kuptosh rëndësinë e informacionit cilësor dhe të saktë.

Gjatë krijimit të një table, nuk mjafton vetëm të përcaktohen fushat që do të përmbajë ajo. Për secilën fushë duhet të përcaktohet edhe lloji i të dhënave që do të futen në të. Kjo është e rëndësishme, sepse informacionet e ndryshme regjistrohen në mënyra të ndryshme. Për shembull, emri dhe mbiemri përbëhen nga shkronja dhe karaktere, data e lindjes futet në formatin e datës, ndërsa numri i mungesave ose notat zakonisht futen si vlera numerike.

Përcaktimi i llojit të përshtatshëm të të dhënave i mundëson programit t'i përpunojë informacionet në mënyrë të saktë. Nëse në një fushë të destinuar për numra futet tekst, ose në një fushë të destinuar për datë futet një vlerë e papërshtatshme, mund të shfaqen gabime gjatë përpunimit të të dhënave. Për këtë arsye, që gjatë krijimit të tabelës përcaktohet se çfarë lloj vlerash mund të përmbajë secila fushë.

Në sistemet moderne për punën me baza të të dhënave ekzistojnë disa lloje të të dhënave. Edhe pse emërtimet mund të ndryshojnë pak nga një program në tjetrin, qëllimi i tyre është i ngjashëm. Më së shpeshti përdoren të dhënat tekstuale, të dhënat numerike, datat dhe orët, vlerat logjike dhe fushat për tekst më të gjatë.

3.3.1 LLOJET MË TË PËRDORURA TË TË DHËNAVE

TË DHËNA TEKSTUALE

Lloji tekstual i të dhënave përdoret për futjen e fjalëve, emrave, adresave, numrave të telefonit dhe informacioneve të tjera që nuk përdoren në llogaritje matematikore. Edhe pse numri i telefonit përmban shifra, ai zakonisht ruhet si tekst, sepse nuk përdoret për mbledhje, zbritje ose shumëzim.

Shembuj:

- Todor Dimov
- Sveti Nikole
- II-3
- 071123456

TË DHËNA NUMERIKE

Lloji numerik i të dhënave përdoret kur të dhënat duhet të përpunohen në mënyrë matematikore. Me to mund të kryhen llogaritje, të përcaktohen mesataret, vlerat maksimale dhe minimale, si dhe të realizohen analiza të tjera statistikore.

Shembuj:

- 5
- 18
- 125
- 98.5

DATA DHE ORA

Ky lloji i të dhënave përdoret për futjen e datave dhe të informacionit kohor. Falë tij, mund të llogaritet mosha e nxënësve, kohëzgjatja e aktiviteteve të caktuara ose diferenca kohore ndërmjet dy ngjarjeve.

Shembuj:

- 15.09.2025
- 21.05.2026
- 08:30

TË DHËNA LOGJIKE

Lloji logjik mundëson futjen e vetëm dy vlerave. Në programe të ndryshme, ato mund të paraqiten si Po/Jo, E vërtetë/E pavërtetë ose E aktivizuar/E çaktivizuar.

Shembuj:

- Po
 - Jo
- ose
- E vërtetë (True)
 - E pavërtetë (False)

TEKST MË I GJATË

Ndonjëherë është e nevojshme të futen përshkrime, shënime ose komente më të gjata. Për situata të tilla përdoret një lloj i të dhënave që mundëson ruajtjen e një sasive më të madhe teksti.

Shembull:

„Nxënësi merr pjesë aktive në gara të informatikës dhe ka fituar disa çmime në gara rajonale dhe shtetërore.“

SHEMBULL PRAKTIK

Të shqyrtohet tabela për nxënësit.

Fusha	Lloji i të dhënave
ID e nxënësit	Numër
Emri	Teksti
Mbiemri	Teksti
Data e lindjes	Data/Koha
Numri i mungesave	Numri
E-mail	Tekst
Nxënës aktiv	Logjike

Nxënësit	
Field Name	Data Type
ID e nxënësit	AutoNumber
Emri	Short Text
Mbiemri	Short Text
Data e lindjes	Date/Time
Telefoni	Short Text
Numri i mungesave	Number
E-posta	Short Text
Nxënës aktiv	Yes/No

Figura 1 Lloji i të dhënave

Nga shembulli mund të vërehet se çdo fushë ka një lloj të caktuar të dhënash. Falë këtij organizimi, programi e di se çfarë vlerash mund të futen dhe si duhet të përpunohen. Për shembull, numri i mungesave mund të mbledhet dhe të analizohet, ndërsa për emrin kjo nuk ka kuptim.

Pse është e rëndësishme të zgjidhet saktë tipi i të dhënës?

Zgjedhja e llojit të përshtatshëm të të dhënave ndikon në saktësinë dhe efikasitetin e punës me bazën e të dhënave. Nëse të gjitha informacionet do të ruheshin si tekst i zakonshëm, programi nuk do të mund të kryente siç duhet llogaritjet, të rendiste të dhënat ose të kontrollonte nëse vlerat e futura janë të sakta.

Përveç kësaj, zgjedhja e duhur kontribuon në shfrytëzimin më të mirë të memories dhe në përpunimin më të shpejtë të informacionit. Në bazat e mëdha të të dhënave që përmbajnë miliona regjistrime, këto optimizime kanë ndikim të konsiderueshëm në funksionimin e sistemit.

Me përcaktimin e kujdesshëm të llojit të të dhënave për secilën fushë krijohet një bazë e mirë për një bazë të dhënash të sigurt dhe efikase. Ky hap përbën një nga pjesët më të rëndësishme gjatë krijimit të tabelave.

Karakteristikat e fushave

Gjatë krijimit të tabelave, për secilën fushë nuk përcaktohen vetëm emri dhe lloji i të dhënave. Përveç tyre, mund të vendosen edhe karakteristika të ndryshme që përcaktojnë mënyrën e futjes, shfaqjes dhe përpunimit të informacionit. Falë këtyre karakteristikave, të dhënat futen në mënyrë më të kontrolluar dhe zvogëlohet mundësia e shfaqjes së gabimeve.

Le të supozojmë se në një tabelë ekziston një fushë për emrin e nxënësit. Nëse nuk vendosen kufizime, përdoruesi mund të fusë një vlerë boshe, një tekst tepër të gjatë ose karaktere të rastësishme që nuk kanë ndonjë kuptim. Duke përdorur karakteristikat e fushave, mund të përcaktohen paraprakisht rregulla që mundësojnë futjen më cilësore të të dhënave.

Karakteristikat e fushave përbëjnë një mekanizëm të rëndësishëm për kontrollin e informacionit në bazën e të dhënave. Ato mundësojnë standardizimin e futjes së të dhënave, organizimin më të mirë të tyre dhe siguri më të madhe gjatë përpunimit të tyre.

Karakteristikat më të përdorura

Madhësia e fushës

Te të dhënat tekstuale shpesh përcaktohet numri maksimal i karaktereve që mund të futen në fushë. Për shembull, nëse dihet se kodi i nxënësit përmban gjithmonë më së shumti dhjetë karaktere, nuk ka nevojë të lejohet futja e një teksti shumë më të gjatë.

Duke kufizuar madhësinë e fushës, zvogëlohet mundësia e futjes së të dhënave të papërshtatshme dhe sigurohet përdorim më efikas i hapësirës së memories.

Fushë e detyrueshme

Ndonjëherë, disa informacione duhet patjetër të futen. Për shembull, në një evidencë shkollare nuk do të ishte e pranueshme të kishte një nxënësit pa emër ose mbiemër. Në situata të tilla përdoret një veti që nuk lejon ruajtjen e regjistrimit nëse fusha është bosh.

Kjo veti kontribuon në përmirësimin e cilësisë së të dhënave dhe parandalon shfaqjen e regjistrimeve të paplota. Në MS Access, kjo bëhet në pjesën General të tabelës, ku një nga mundësitë është të zgjidhet “Po” (Yes) te fusha “E detyrueshme” (Required)

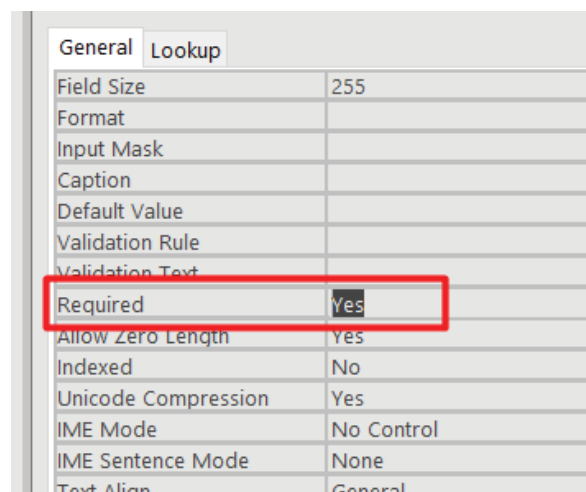


Figura 2 – Fusha e detyrueshme

General		Lookup
Field Size	255	
Format		
Input Mask		
Caption		
Default Value	2026	
Validation Rule		
Validation Text		

Figura 3 – Vlera që do të shfaqet automatikisht

Vlera e paracaktuar

Në situata të caktuara, një numër i madh regjistrimesh kanë të njëjtën vlerë fillestare. Në vend që përdoruesi ta fusë vazhdimisht të njëjtin informacion, mund të vendoset paraprakisht një vlerë e paracaktuar.

Për shembull, nëse shumica e nxënësve i përkasin vitit aktual shkollor, kjo vlerë mund të shfaqet automatikisht gjatë futjes së një regjistrimi të ri. Përdoruesi do ta ndryshojë atë vetëm nëse është e nevojshme.

Formati i të dhënave

Disa informacione duhet të paraqiten në një format standard. Datat, numrat dhe vlerat monetare zakonisht kanë një format të përcaktuar paraprakisht.

Falë kësaj vetie, të gjitha të dhënat e të njëjtit lloj paraqiten në një mënyrë të unifikuar dhe të qartë, gjë që përmirëson lexueshmërinë dhe kuptueshmërinë e informacionit.

Vlefshmëria e të dhënave

Në çdo bazë të të dhënave ekziston mundësia që përdoruesit, pa dashje, të fusin informacione të pasakta. Në vend të një numri mund të futet tekst, në vend të datës mund të futet një vlerë e rastësishme ose mund të futet një e dhënë që nuk i plotëson rregullat e sistemit. Për të parandaluar situata të tilla përdoret vlefshmëria e të dhënave (validimi).

Validimi është një proces me të cilin kontrollohet nëse vlera e futur i plotëson kushtet e përcaktuara paraprakisht. Nëse vlera nuk i plotëson rregullat, sistemi e paralajmëron përdoruesin dhe nuk lejon ruajtjen e informacionit të pasaktë.

Nëse dizajnohet një sistem për llogaritjen e pagesës së parkimit përmes një mesazhi SMS, është e nevojshme të validohet nëse targat e automjetit janë në formatin e duhur. Shpesh njerëzit bëjnë gabime gjatë futjes së numrit të targave. Në fushën „Input Mask“ të skedës „General“ mund të vendoset validimi. Nëse presim që në sistem të futen shkronja dhe numra sipas formatit të targave, duhet të përcaktohet se ajo përmban dy shkronja, të ndjekura nga katër numra dhe në fund dy shkronja.

Këtu „L“ tregon se në atë pozicion duhet të jetë vetëm shkronjë, ndërsa „0“ tregon pozicionin për shifër. Në këtë mënyrë, maska e vendosur kontrollon nëse targa e automjetit është futur saktë.

General		Lookup
Field Size	Decimal	
Format		
Precision	18	
Scale	2	
Decimal Places	2	
Input Mask	LL 0000 LL	
Caption		
Default Value	0	
Validation Rule		

Figura 4 – Vendosja e maskës në regjistrim

Ky mekanizëm ka rëndësi të madhe në sistemet informatike që përpunojnë sasi të mëdha të të dhënave. Edhe një gabim i vogël gjatë futjes së të dhënave mund të shkaktojë probleme gjatë kërkimit, analizimit ose përgatitjes së raporteve. Për këtë arsye, bazat moderne të të dhënave përdorin teknika të ndryshme për kontrollimin e informacionit të futur.

Shembuj të vlefshmërisë së të dhënave

Të shqyrtohen situatat e mëposhtme:

Shembulli 1

Fusha: Moshë

Rregulli: Vlera duhet të jetë ndërmjet 14 dhe

19.

✓ 16

✓ 18

x -5

x 120

General	Lookup
Field Size	255
Format	
Input Mask	
Caption	
Default Value	
Validation Rule	>=14 And <=19
Validation Text	
Required	Yes

Figura 5 – Përcaktimi i rregullave të vlefshmërisë

Në këtë rast, sistemi do të pranojë vetëm vlera reale të moshës së nxënësve.

Kështu duhet të duket validimi i të dhënave.

Shembulli 2

Fusha: Nota

Rregulli: Lejohen vetëm vlerat nga 1 deri në 5.

✓ 1

✓ 5

x 7

x 0

General	Lookup
Field Size	255
Format	
Input Mask	
Caption	
Default Value	
Validation Rule	>=1 And <=5
Validation Text	Lejohen vetëm vlerat nga 1 deri në 5.
Required	Yes
Allow Zero Length	Yes
Indexed	No

Figura 6 – Përcaktimi i rregullave dhe i tekstit të vlefshmërisë

Me këtë rregull parandalohet futja e notave që nuk ekzistojnë në sistemin e vlerësimit.

Teksti i validimit shpjegon më tej se çfarë duhet të përmbajë fusha. Është praktikë e mirë që, kur përdoret validimi, të vendoset edhe një tekst validimi. Kur programi sinjalizon një gabim, në vend që të dëgjohet vetëm një tingull, në këtë rast shfaqet edhe një mesazh me përshkrimin e gabimit.

Shembulli 3

Fusha: E-mail

Rregulli: Vlera duhet të përmbajë shenjë @.

✓ učenik@uciliste.mk

x učenik@gmail.com

Validimi ndihmon në zbulimin e gabimeve të

General	Lookup
Field Size	255
Format	
Input Mask	
Caption	
Default Value	
Validation Rule	Like "*"@"
Validation Text	Vlera duhet të përmbajë shenjë @.
Required	Yes
Allow Zero Length	Yes
Indexed	No
Unicode Compression	Yes

dukshme që gjatë futjes së të dhënave.

Figura 7 – Validimi me shenjë të detyrueshme

Yjet e vendosur para dhe pas shenjës që kërkohet tregojnë se para dhe pas asaj shenje duhet patjetër të ketë tekst.

Analiza e shembujve

Nga shembujt e mëparshëm mund të vërehet se validimi nuk shërben për zbulimin e të dhënave, por për mbrojtjen e bazës së të dhënave nga informacionet e pasakta. Sa më i madh të jetë numri i përdoruesve që fusin të dhëna, aq më e madhe është nevoja për kontroll automatik të vlerave.

Kombinimi i llojeve të të dhënave, vetive të fushave dhe rregullave të validimit krijon një bazë të sigurt për punën me informacionet. Falë këtyre mekanizmave, të dhënat bëhen më të qëndrueshme, më të sakta dhe më të lehta për t'u përpunuar.



PËRFUNDIM:

Lloji i të dhënave dhe vetitë e fushës kanë ndikim të madh në cilësinë e informacioneve në bazën e të dhënave. Zgjedhja e duhur mundëson që të dhënat të përpunohen, të kërkohen dhe të analizohen më lehtë.

Validimi dhe vetitë e fushave përfaqësojnë mekanizma të rëndësishëm për kontrollin e informacioneve të futura. Ato ndihmojnë në zvogëlimin e mundësisë për gabime dhe sigurojnë siguri e saktësi më të madhe të të dhënave.



A E DI?

A e dini se kompjuteri i përpunon në mënyra të ndryshme tekstet, numrat dhe datat? Falë llojeve të të dhënave, sistemi e di nëse një vlerë mund të përdoret për llogaritje apo duhet vetëm të paraqitet.



PYETJE PËR PËRSËRITJE

1. Çfarë përfaqëson tipi i të dhënave?
2. Cilat lloje të të dhënave përdoren më shpesh?
3. Kur përdoret tipi tekstual i të dhënave?
4. Kur përdoret tipi numerik i të dhënave?
5. Çfarë përfaqëson validimi i të dhënave?
6. Pse përdoren vetitë e fushave?
7. Çfarë është vlera e paracaktuar?

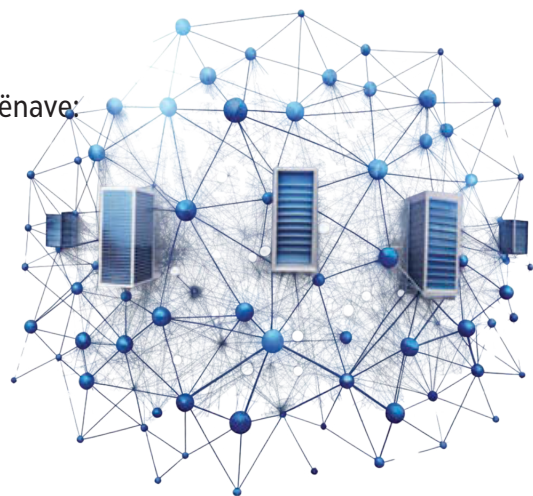


DETYRA PRAKTIKE

Detyra 1

Për secilën fushë përcakto llojin e përshtatshëm të të dhënave:

Fusha
Emri
Dita e lindjes
Numri i mungesave
Nxënës aktiv
Vërejtje



Detyra 2

Përmend tri fusha për të cilat do të vendosësh që futja e të dhënave të jetë e detyrueshme.

Detyra 3

Jep një shembull të një fushe që do të kishte një vlerë të paracaktuar.

Detyra 4

Shpjego pse mosha duhet të jetë e dhënë numerike dhe jo tekstuale.



DETYRË HULUMTUESE

Hulumto se cilat lloje të të dhënave përdorin Microsoft Access dhe LibreOffice Base. Bëj një krahasim dhe gjej cilat lloje të të dhënave janë të përbashkëta për të dyja programet.



MBAJ MEND ÇFARË MËSOVE

Çdo fushë ka një lloj të përcaktuar të të dhënave.

Lloji i të dhënave përcakton se çfarë vlerash mund të futen.

Vetitë e fushave mundësojnë kontrollin mbi të dhënat.

Validimi parandalon futjen e vlerave të pasakta.

Zgjedhja e duhur e llojit të të dhënave përmirëson efikasitetin e bazës së të dhënave.

PËR ATA QË DËSHIROJNË TË DINË MË SHUMË



Në sistemet profesionale ekzistojnë dhjetëra lloje të ndryshme të të dhënave. Disa prej tyre përdoren për ruajtjen e fotografive, videove, koordinatave gjeografike ose dokumenteve komplekse.



KONTROLLO NJOHURITË E TUA

Rretho përgjigjen e saktë

1. Për datën e lindjes, lloji më i përshtatshëm është:

- a) teksti
- b) numër
- c) data/koha
- ç) logjike

2. Validimi shërben për:

- a) fshirjen e regjistrimeve
- b) kontrollimin e të dhënave të futura
- c) printimin e raporteve
- ç) krijimin e tabelave

E saktë apo e pasaktë

- Në fushën e moshës mund të futet çfarëdo teksti.
- Lloji logjik zakonisht ka dy vlera të mundshme.
- Validimi përmirëson cilësinë e të dhënave.





3.4

ÇELËSI PRIMAR DHE LIDHJA E TABELAVE

TASHMË DI ...

- të krijosh tabela në një bazë të të dhënave;
- të përdorësh lloje të ndryshme të të dhënave;
- të zbatosh vetitë e fushave;
- të përdorësh rregulla për validimin e të dhënave;
- të organizosh informacionet në formë tabelare.

PYETJE PËR TË MENDUAR

- Çfarë do të ndodhë nëse dy regjistrime kanë të njëjtat të dhëna?
- Si mund të dallohen me siguri dy nxënës me të njëjtin emër dhe mbiemër?
- Pse çdo regjistrim duhet të ketë një identifikues unik?
- Si e gjen shpejt kompjuteri regjistrimin e nevojshëm?
- Si mund të përdoren të njëjtat informacione në tabela të ndryshme pa i përsëritur ato?

NË PËRMBAJTJEN QË VIJON DO TË MËSOSH:

- të shpjegosh çfarë përfaqëson çelësi primar;
- të dallosh fushat e përshtatshme për çelës primar;
- të analizosh rëndësinë e identifikimit unik;
- të shpjegosh përparësitë e përdorimit të çelësit primar;
- të zbatosh çelësin primar gjatë krijimit të tabelave;
- të analizosh organizimin e të dhënave në një bazë të dhënash relacionale.

Kur punohet me një numër të madh regjistrimesh, shpesh paraqitet nevoja që çdo regjistrim të mund të identifikohet në mënyrë të qartë dhe të padyshimtë. Le të supozojmë se në një tabelë ruhen të dhëna për nxënësit. Është e mundur që dy nxënës të kenë të njëjtin emër dhe mbiemër, prandaj vetëm nga këto të dhëna nuk mund të përcaktohet me siguri se për cilin nxënës bëhet fjalë. Për këtë arsye, në bazat e të dhënave përdoret një fushë e veçantë, e cila mundëson identifikimin unik të çdo regjistrimi.

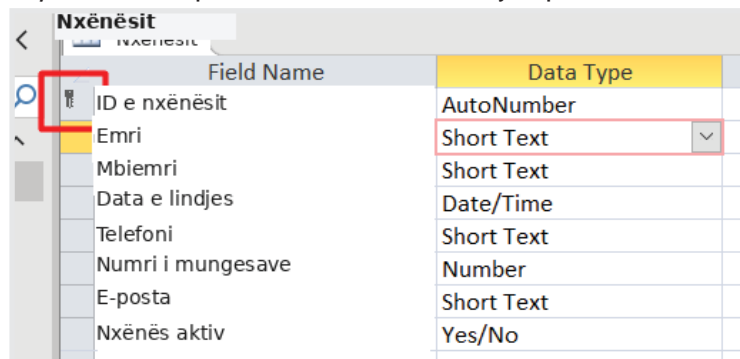
Fusha që përmban një vlerë unike për çdo regjistrim quhet çelës primar. Detyra e tij themelore është të sigurojë që çdo regjistrim në tabelë të jetë unik dhe lehtësisht i identifikueshëm. Vlera e çelësit primar nuk duhet të përsëritet te dy regjistrime të ndryshme dhe nuk mund të mbetet bosh.

Çelësi primar ka rëndësi të madhe për funksionimin e duhur të bazës së të dhënave. Përveçse siguron identifikimin unik të regjistrimeve, ai shërben edhe si bazë për lidhjen e tabelave dhe organizimin e sistemeve komplekse të informacionit. Për këtë arsye, që gjatë krijimit të tabelës, duhet të zgjidhet me kujdes se cila fushë do ta ketë këtë rol.

3.4.1 KARAKTERISTIKAT E ÇELËSIT PRIMAR

Që një fushë të mund të përdoret si çelës primar, ajo duhet të plotësojë disa kushte.

Së pari, vlera duhet të jetë unike për çdo regjistrim. Nëse dy persona kanë të njëjtin numër identifikimi, sistemi nuk do të jetë në gjendje t'i dallojë saktë. Së dyti, vlera nuk duhet të jetë bosh, sepse çdo regjistrim duhet të ketë identifikuesin e vet unik. Së treti, është e dëshirueshme që vlera të jetë e qëndrueshme dhe të ndryshohet rrallë, pasi ndryshimet e shpeshta mund të shkaktojnë probleme në organizimin e të dhënave.



Field Name	Data Type
ID e nxënësit	AutoNumber
Emri	Short Text
Mbiemri	Short Text
Data e lindjes	Date/Time
Telefoni	Short Text
Numri i mungesave	Number
E-posta	Short Text
Nxënës aktiv	Yes/No

Figura 1. Çelësi primar i vendosur

Në shumë raste, si çelës primar përdoret një fushë e veçantë që automatikisht merr një vlerë të re gjatë shtimit të çdo regjistrimi të ri. Në këtë mënyrë shmanget mundësia që përdoruesi, gabimisht, të fusë një vlerë që tashmë ekziston.

Shembull

Shqyrto tabelën e mëposhtme.

NxënësID	Emri	Mbiemri
1	Ana	Petrova
2	Ajet	Rexhepi
3	Ana	Petrova
4	John	Wayne

Nga tabela mund të vërehet se emri dhe mbiemri „Ana Petrova“ shfaqen dy herë. Nëse për identifikim do të përdorej vetëm emri dhe mbiemri, do të ishte e vështirë të dalloheshin këto dy regjistrime.

Fusha NxënësID përmban një vlerë të ndryshme për çdo nxënës dhe, për këtë arsye, mund të përdoret si çelës primar. Pavarësisht se sa nxënës kanë të njëjtin emër ose mbiemër, numri i tyre i identifikimit do të jetë i ndryshëm. Në këtë rast, si numër identifikimi mund të merret një e dhënë unike e nxënësit, për shembull, numri i tij në ditar. Në një klasë mund të ketë vetëm një nxënës me numrin 3. Në sistemin arsimor përdoret EIBU-ja (Numri Unik i Identifikimit të Nxënësit); në Ministrinë e Punëve të Brendshme përdoret numri i letërnjoftimit; në Ministrinë e Punëve të Jashtme përdoret numri i pasaportës; ndërsa në Ministrinë e Shëndetësisë përdoret numri i librezës shëndetësore. Të gjitha këto të dhëna lidhen me numrin personal (Numri Amë i Vetëm i Qytetarit). Për shkak të rregulloreve ligjore, jo çdokush mund të ketë qasje në numrin personal një personi. Prandaj, çdo institucion krijon një mënyrë unike për t'i dalluar qytetarët që kanë të njëjtat të dhëna personale.

Nëse një organizatë ose kompani u lëshon karta klientëve të saj dhe dëshiron që numrat e tyre të jenë të njëpasnjëshëm, si tip i të dhënave përdoret Autonumber (Numërim automatik). Ky është një opsion shumë i dobishëm edhe te arkat fiskale nëpër dyqane, të cilat gjatë një dite lëshojnë edhe qindra kuponë fiskalë.

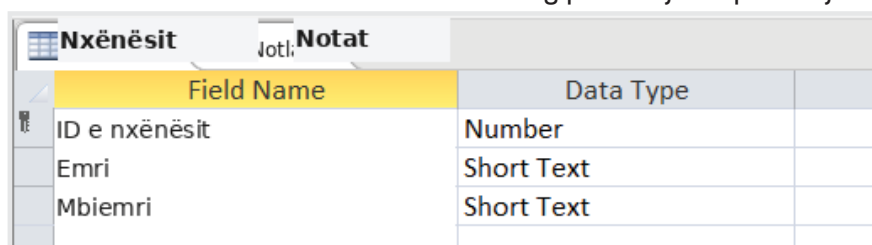
Analiza e shembullit

Nga shembulli i mëparshëm mund të vërehet se çelësi primar nuk është e nevojshme të ketë kuptim për përdoruesin. Roli i tij është të sigurojë veçantinë e regjistrimeve dhe organizimin e duhur të të dhënave. Në praktikë, përdoruesit zakonisht shohin emrat, adresat ose informacionet e tjera, ndërsa sistemi, në prapavijë, përdor çelësat primarë për menaxhimin e sigurt të të dhënave.

Përdorimi i çelësit primar e lehtëson ndjeshëm kërkimin, përditësimin dhe mirëmbajtjen e informacionit. Falë tij, çdo regjistrim mund të gjendet dhe të përpunohet shpejt, pa rrezikun që të ngatërrohet me regjistrime të tjera që përmbajnë të dhëna të ngjashme.

3.4.2 LIDHJA E TABELAVE

Në bazat e të dhënave më të mëdha, informacionet zakonisht organizohen në disa tabela. Për t'i përdorur këto tabela së bashku, është e nevojshme që ndërmjet tyre të ekzistojë një lidhje. Lidhja e tabelave mundëson organizimin më efikas të informacionit dhe shmang përsëritjen e panevojshme të të dhënave.



Field Name	Data Type
ID e nxënësit	Number
Emri	Short Text
Mbiemri	Short Text

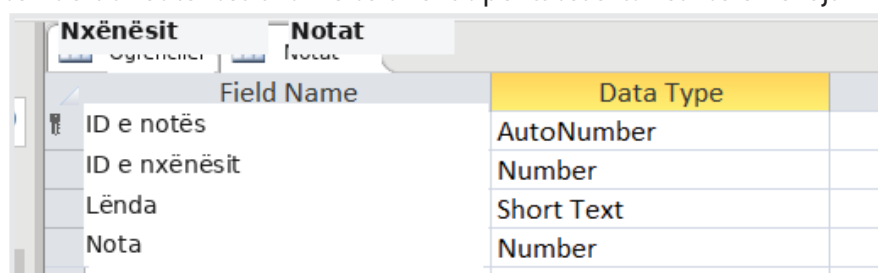
Figura 2. Modifikimi i tabelës me nxënësit



ID e nxënësit	Emri	Mbiemri	Click to Add
1	Ana	Petrova	
2	Marko	Stojanov	
3	Elena	Nikolovska	
(New)			

Figura 3. Tabela me nxënësit

Të supozojmë se ekziston një tabelë me të dhënat e nxënësve dhe një tabelë e veçantë me notat e tyre. Në vend që në çdo regjistrim të notës të përsëriten emri dhe mbiemri i nxënësit, mund të përdoret numri identifikues i nxënësit nga tabela e nxënësve. Në këtë mënyrë, informacioni ruhet vetëm një herë, ndërsa megjithatë mund të ndërlidhet lehtësisht me të dhënat përkatëse kur është e nevojshme.



Field Name	Data Type
ID e notës	AutoNumber
ID e nxënësit	Number
Lënda	Short Text
Nota	Number

Figura 4. Modifikimi i tabelës me nota

Nxënësit		Notat		
ID e notës	ID e nxënësit	Lënda	Nota	Click to Add
1	1	Teknologji informatike	4	
2	1	Histori	5	
3	2	Teknologji informatike	5	
4	3	Teknologji informatike	3	
5	1	Matematikë	4	

Figura 5. Plotësimi i tabelës me nota

Lidhja ndërmjet tabelave përbën një nga karakteristikat më të rëndësishme të bazave relacionale të të dhënave. Falë kësaj lidhjeje, të dhënat mbeten të organizuara, ndërsa sistemi mund të përpunojë në mënyrë efektive edhe një sasi të madhe informacioni.

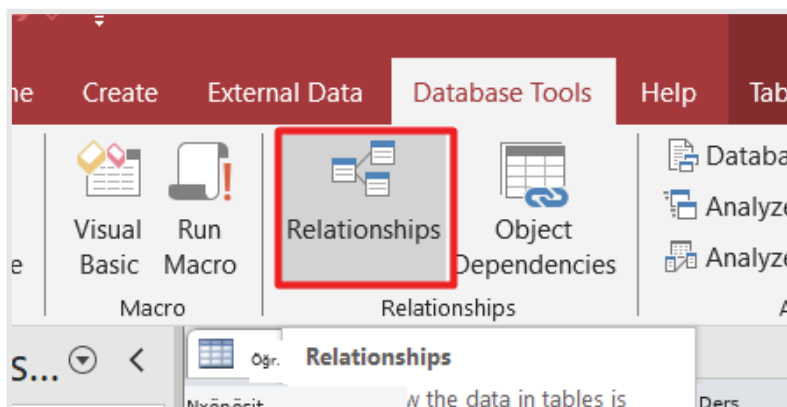


Figura 6. Vendosja e marrëdhënieve në bazën e të dhënave

Gjatë krijimit të marrëdhënieve ndërmjet tabelave, është e nevojshme që fushat që lidhen me njëra-tjetrën të jenë të të njëjtit lloj të dhënash. Në këtë rast, të dyja tabelat do të lidhen përmes fushës ID e nxënësit.

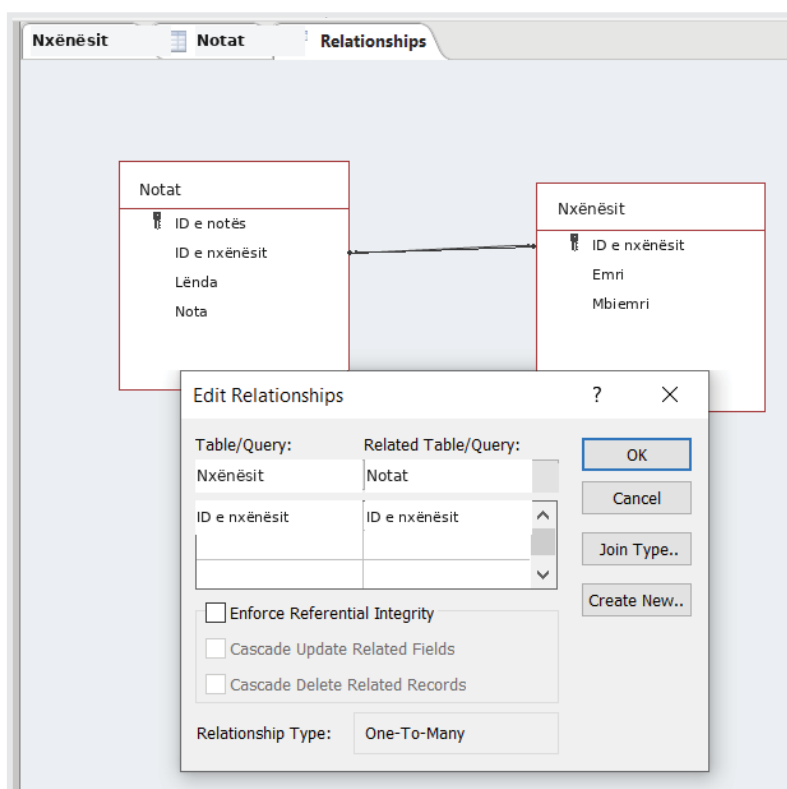


Figura 7. Krijimi i marrëdhënieve ndërmjet tabelave

Redaktimi dhe përditësimi i tabelave

Bazat e të dhënave rrallëherë mbeten të pandryshuara. Me kalimin e kohës, shfaqet nevoja për shtimin e fushave të reja, ndryshimin e vetive ekzistuese ose futjen e regjistrimeve të reja. Për këtë arsye, sistemet moderne mundësojnë redaktimin dhe përditësimin e lehtë të tabelave.

Redaktimi i tabelës mund të përfshijë shtimin e një fushe të re, ndryshimin e emrit të një fushe ekzistuese ose ndryshimin e llojit të të dhënave të saj. Këto ndryshime zakonisht bëhen kur shfaqen kërkesa të reja ose kur nevojitet një organizim më i mirë i informacionit.

Përditësimi i tabelave i referohet ndryshimit të të dhënave. Për shembull, mund të ndryshohet numri i telefonit, adresa ose ndonjë informacion tjetër që tashmë ekziston në bazën e të dhënave. Përditësimi i rregullt është i domosdoshëm që të dhënat të mbeten të sakta dhe të dobishme.

Tabelat e organizuara mirë, fushat e zgjedhura në mënyrë të përshtatshme dhe përditësimi i kujdesshëm i të dhënave përbëjnë bazën e një baze të dhënash të sigurt dhe efikase. Falë këtyre veprimeve, informacioni mbetet i saktë, i organizuar dhe lehtësisht i qasshëm për përdoruesit.



PËRFUNDIM:

Çelësi primar përbën një nga elementet më të rëndësishme të bazave relacionale të të dhënave. Detyra e tij kryesore është të sigurojë që çdo regjistrim të jetë unik dhe lehtësisht i identifikueshëm. Falë çelësit primar, të dhënat mund të organizohen dhe mirëmbahen në mënyrë të sigurt.

Përdorimi i çelësit primar përmirëson efikasitetin e kërkimit, përditësimit dhe menaxhimit të informacionit. Ai përbën bazën për krijimin e bazave të të dhënave më të mëdha dhe më komplekse.



A E DI?

A e di se mund të ketë dy persona me të njëjtin emër dhe mbiemër, por nuk mund të kenë të njëjtin numër amë? Një parim i ngjashëm përdoret edhe në bazat e të dhënave përmes çelësit primar.



PYETJE PËR PËRSËRITJE

1. Çfarë përfaqëson çelësi primar?
2. Cilat kushte duhet t'i plotësojë çelësi primar?
3. Pse vlerat e çelësit primar nuk duhet të përsëriten?
4. A mund të jetë çelësi primar bosh?
5. Cilat janë përparësitë e përdorimit të çelësit primar?
6. Si ndihmon çelësi primar në kërkimin e të dhënave?



DETYRA PRAKTIKE

Detyra 1

Përcakto se cila fushë do të ishte më e përshtatshme për çelës primar:

| Shifra | Produkt | Sasia |

Detyra 2

Jep tre shembuj të fushave që mund të përdoren si çelës primar.

Detyra 3

Shpjego pse emri dhe mbiemri nuk janë gjithmonë zgjedhje e mirë për çelës primar.

Detyra 4

Mendo se çfarë do të ndodhte nëse dy regjistrime do të kishin të njëjtën vlerë të çelësit primar.



DETYRË HULUMTUESE

Hulumtose si institucionet të ndryshme sigurojnë identifikimin unik të personave, produkteve ose dokumenteve. Gjej të paktën tre shembuj dhe shpjego si funksionojnë.



MTBAJ MEND ÇFARË MËSOVE

- Çelësi primar mundëson identifikimin unik të regjistrimeve.
- Vlerat e tij nuk duhet të përsëriten.
- Nuk duhet të përmbajë vlera boshe.
- Lehtëson kërkimin dhe përditësimin e të dhënave.
- Përfaqëson një element të rëndësishëm në organizimin e bazave relacionale të të dhënave.



PËR ATA QË DËSHIROJNË TË DINË MË SHUMË

Në bazat e mëdha të të dhënave, çelësat primarë shpesh gjenerohen automatikisht. Në këtë mënyrë, sistemi garanton që çdo regjistrim i ri të marrë një numër unik identifikues.



KONTROLLO NJOHURITË E TUA

Rretho përgjigjen e saktë

1. Çelësi primar:

- a) mund të përsëritet
- b) duhet të jetë unik
- c) nuk është i rëndësishëm
- ç) shërben për formatim

2. Çelësi primar përdoret më shpesh për:

- a) identifikimin e regjistrimeve
- b) printimin e raporteve
- c) fshirjen e tabelave
- ç) futjen e tekstit

E saktë ose e pasaktë

1. Dy regjistrime të ndryshme mund të kenë të njëjtin çelës primar.
2. Çelësi primar mund të jetë bosh.
3. Çelësi primar ndihmon në organizimin e të dhënave.





3.5

FORMA PËR FUTJEN DHE SHQYRTIMIN E TË DHËNAVE

TASHMË DI ...

- të krijosh tabela në bazën e të dhënave;
- të përdorësh lloje të ndryshme të të dhënave;
- të përcaktosh vetitë e fushave;
- të përdorësh çelësin primar;
- të futësh dhe të përditësosh të dhënat në tabela.

PYETJE PËR TË MENDUAR

- A është praktike që përdoruesi t'i fusë gjithmonë të dhënat drejtpërdrejt në tabela?
- Si mund të thjeshtohet futja e informacionit?
- A duhet që të gjithë përdoruesit ta shohin strukturën e tabelës?
- Si mund të paraqiten të dhënat në mënyrë më të qartë?
- A është e mundur që futja e të dhënave të duket si plotësimi i një formulari elektronik?

NË PËRMBAJTJET QË VIJOJNË DO TË MËSOSH:

- të shpjegosh se çfarë është një formular;
- të përmendësh përparësitë e përdorimit të formularëve;
- të krijosh një formular të thjeshtë;
- të futësh të dhëna përmes formularit;
- të redaktosh regjistrimet ekzistuese;
- të analizosh zbatimin e formularëve në sistemet informatike.

Në mësimet e mëparshme, të dhënat futeshin drejtpërdrejt në tabela. Edhe pse kjo mënyrë është e thjeshtë, ajo nuk është gjithmonë më praktike. Në bazat e të dhënave më të mëdha, përdoruesit shpesh nuk kanë nevojë ta shohin të gjithë tabelën dhe as të punojnë me të gjitha fushat njëkohësisht. Në vend të kësaj, është shumë më praktike që të dhënat të futen përmes një ekrani të përgatitur posaçërisht, i cili shfaq vetëm informacionet e nevojshme.

Pikërisht për këtë arsye, në bazat e të dhënave përdoren formularët. Formularët mundësojnë që të dhënat të futen, të shikohen dhe të përditësohen në mënyrë më të qartë dhe më të kontrolluar. Në vend që përdoruesi të punojë me rreshta dhe kolona, ai punon me fusha, butona, lista dhe elemente të tjera që ngjajnë me një formular elektronik.

Në jetën e përditshme hasen shpesh forma të ndryshme. Plotësimi i një aplikimi online, regjistrimi i një llogarie përdoruesi, futja e të dhënave për një porosi ose regjistrimi elektronik për një provim janë shembuj të formave që, në prapavijë, komunikojnë me një bazë të të dhënave.

Nga tabela në formular

Të shqyrtohet tabela e nxënësve.

NxënësID	Emri	Mbiemri	Telefon	E-mail
1	Ana	Petrova	071123456	ana@email.com
2	Marko	Stojanov	075654321	marko@email.com

Në pamje të parë, tabela përmban të gjitha informacionet e nevojshme. Megjithatë, nëse përdoruesi duhet të fusë çdo ditë të dhëna për nxënës të rinj, kjo pamje nuk është gjithmonë më praktike. Veçanërisht kur tabela përmban dhjetëra ose njëzet fusha.

Për këtë arsye, mund të krijohet një formular që i paraqet të dhënat në mënyrë më të thjeshtë dhe më të qartë.

Përparësitë e përdorimit të formularëve

Formularët përmirësojnë ndjeshëm komunikimin ndërmjet përdoruesit dhe bazës së të dhënave. Ata mundësojnë futjen më të thjeshtë të informacionit dhe zvogëlojnë mundësinë e gabimeve. Në vend që përdoruesi të lëvizë nëpër një tabelë të madhe, vëmendja përqendrohet vetëm te të dhënat që duhet të futen.

Një përparësi tjetër është mundësia për organizim më të mirë të elementeve. Informacionet e ndërlikura mund të grupohen, ndërsa mund të përdoren butona për përzgjedhje, lista rënësë ose fusha për shënim. Në këtë mënyrë, formulari bëhet më i qartë dhe më i lehtë për t'u përdorur.

Formularët mundësojnë gjithashtu kufizimin e qasjes në informacione të caktuara. Përdoruesi mund të punojë me formularin pa e hapur drejtpërdrejt tabelën. Kjo kontribuon në siguri më të mirë dhe në kontroll më të madh mbi të dhënat.

Krijimi i formularit

Në programet Microsoft Access dhe LibreOffice Base ekzistojnë mjete që mundësojnë krijimin automatik të formularëve në bazë të një table ekzistuese. Fillimisht, përdoruesi zgjedh tabelën që do të shërbejë si burim i të dhënave dhe më pas programi i vendos automatikisht fushat në formular.

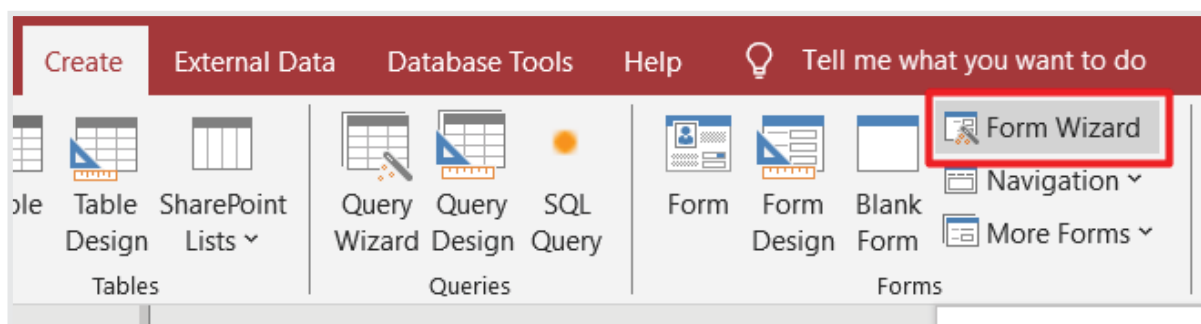


Figura 1. Hapja e formularit

Pasi të krijohet formulari, është e mundur të rregullohet më tej pamja e tij. Mund të ndryshohen titujt, madhësia e elementeve, pozicioni i tyre dhe mënyra e paraqitjes së të dhënave. Në këtë mënyrë, formulari mund të përshtatet me nevojat konkrete të përdoruesve.

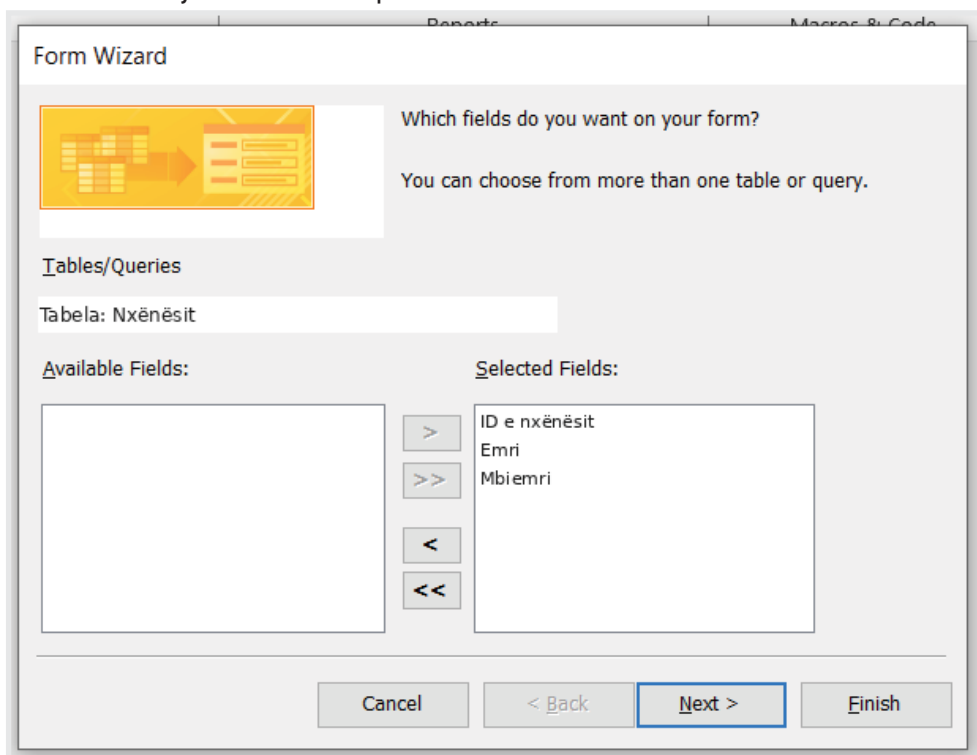


Figura 2. Zgjedhja e tabelës dhe e fushave nga tabela

Të gjitha fushat e nevojshme përzgjidhen nga tabela përkatëse dhe vendosen në formular. Nëse baza e të dhënave përmban disa tabela, fillimisht duhet të zgjidhet tabela, e më pas fushat që duhet të shfaqen në formular.

Futja e të dhënave përmes formularit

Pas krijimit të formularit, ai mund të përdoret për futjen e të dhënave të reja në bazën e të dhënave. Përdoruesi nuk ka më nevojë ta hapë drejtpërdrejt tabelën dhe të kërkojë rreshtin përkatës për futjen e informacionit. Në vend të kësaj, mjafton të plotësojë fushat që shfaqen në formular.

Gjatë futjes së një regjistrimi të ri, çdo informacion vendoset në fushën përkatëse. Nëse për një fushë është përcaktuar ndonjë rregull vërtetimi ose kufizim, programi kontrollon automatikisht nëse vlera e futur i plotëson kushtet e përcaktuara. Në këtë mënyrë zvogëlohet mundësia e futjes së të dhënave të pasakta ose jo të plota.

Formularët mundësojnë që procesi i futjes së informacionit të jetë më i ngjashëm me plotësimin e një formulari elektronik sesa me punën me një tabelë klasike. Kjo është veçanërisht e dobishme kur baza e të dhënave përdoret nga persona që nuk kanë njohuri të avancuara në teknologjinë e informacionit.

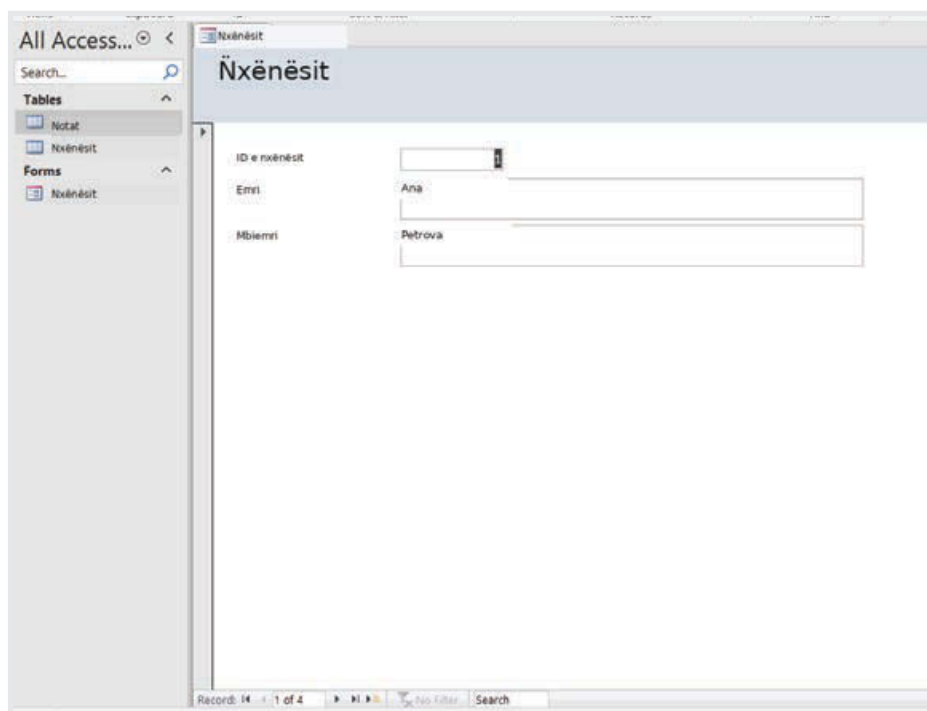


Figura 3. Formulari në MS Access

Pas krijimit të formularit përmes Magjistarit për formularë, shfaqen të gjitha regjistrimet, nëse ato ekzistojnë.

Nga shembulli i mëparshëm mund të vërehet se përdoruesi nuk sheh më rreshta dhe kolona. Në vend të kësaj, çdo e dhënë futet në një fushë të veçantë. Kjo mënyrë e punës është më e ngjashme me formularët elektronikë që përdorim çdo ditë në internet.

Redaktimi i regjistrimeve ekzistuese

Të dhënat e futura në bazën e të dhënave nuk mbeten gjithmonë të pandryshuara. Numrat e telefonit ndryshojnë, përdoruesit marrin adresa të reja të postës elektronike, ndërsa informacione të caktuara duhet të përditësohen rregullisht. Formularët mundësojnë që këto ndryshime të kryhen thjesht dhe shpejt.

Kur përdoruesi hap një regjistrim ekzistues përmes formularit, të gjitha informacionet shfaqen në fusha që mund të ndryshohen drejtpërdrejt. Pas kryerjes së ndryshimit, të dhënat e reja ruhen automatikisht në bazën e të dhënave. Në këtë mënyrë, përdoruesi nuk ka nevojë ta kërkojë informacionin përkatës në tabelë.

Figura 4. Ndryshimi i një regjistrimi ekzistues

Redaktimi përmes formularit mundëson qartësi më të madhe dhe kontroll më të mirë të informacionit. Përdoruesi përqendrohet vetëm te regjistrimi që po përpunon, pa u ngarkuar me të dhënat e tjera që ndodhen në tabelë.

Navigimi nëpër regjistrime

Një formular zakonisht nuk shfaq vetëm një regjistrim. Ai mundëson lëvizjen nëpër të gjitha regjistrimet që ndodhen në tabelë. Për këtë qëllim përdoren butonat e navigimit, përmes të cilëve përdoruesi mund të kalojë te regjistrimi i mëparshëm, i ardhshëm, i parë ose i fundit.

Kjo mundësi është veçanërisht e dobishme kur tabela përmban një numër të madh regjistrimesh. Në vend që përdoruesi t'i kërkojë të dhënat manualisht, mjafton të përdorë elementet e navigimit të formularit. Në këtë mënyrë, shqyrtimi i informacionit bëhet më i shpejtë dhe më i thjeshtë.

Në shumë sisteme moderne informatike, formularët përmbajnë edhe mundësi shtesë për kërkimin dhe filtrimin e të dhënave. Edhe pse këto funksione do të trajtohen më hollësisht në mësimet e ardhshme, është e rëndësishme të kuptohet se formularët nuk shërbejnë vetëm për futjen e informacionit, por edhe për përdorimin e tij në mënyrë efikase.

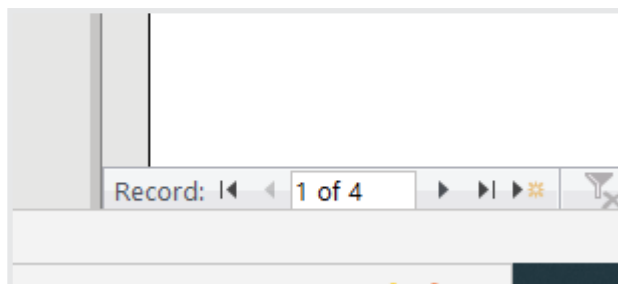


Figura 5. Navigimi nëpër regjistrime

Shembull praktik: Evidentimi i nxënësve

Le të supozojmë se nevojitet të përgatitet një sistem i thjeshtë për evidentimin e nxënësve në një shkollë. Në bazën e të dhënave tashmë ekziston një tabelë që përmban informacionet e nevojshme për ta. Në vend që mësuesi ose punonjësi administrativ të punojë drejtpërdrejt me tabelën, mund të krijohet një formular që mundëson futjen dhe shqyrtimin më të lehtë të të dhënave.

Formulari mund të përmbajë fusha për: emrin, mbiemrin, numrin e telefonit, adresën e postës elektronike dhe klasën. Gjatë futjes së një nxënësi të ri, përdoruesi plotëson fushat, ndërsa sistemi e ruan automatikisht regjistrimin në bazën e të dhënave. Nëse më vonë duhet të ndryshohet ndonjë e dhënë, përdoruesi thjesht hap regjistrimin përkatës përmes formularit dhe bën ndryshimin e nevojshëm.

Kjo mënyrë e punës e përmirëson ndjeshëm efikasitetin. Në vend të një tablele të ndërlikuar me shumë rreshta dhe kolona, përdoruesi punon me një ekran të thjeshtë dhe të qartë, i cili shfaq vetëm informacionet që i nevojiten. Për këtë arsye, formularët përfaqësojnë një nga elementet më të përdorura në bazat moderne të të dhënave.

Analiza e shembullit

Nga shembulli i mëparshëm mund të vërehet se formularët përfaqësojnë një ndërmjetës midis përdoruesit dhe bazës së të dhënave. Ata nuk i zëvendësojnë tabelat, por mundësojnë punë më të lehtë me to. Të gjitha të dhënat vazhdojnë të ruhen në tabela, por përdoruesi më së shpeshti komunikon përmes formularëve.

Me përdorimin e formularëve përmirësohet qartësia e të dhënave, zvogëlohet mundësia për gabime dhe lehtësohet futja dhe përditësimi i të dhënave. Për shkak të këtyre përparësive, formularët kanë një rol të rëndësishëm në zhvillimin e sistemeve informatike të bazuara në baza të të dhënave.



PËRFUNDIM:

Formularët përfaqësojnë një element të rëndësishëm të bazave të të dhënave dhe mundësojnë komunikim më të thjeshtë ndërmjet përdoruesit dhe të dhënave. Në vend që të punohet drejtpërdrejt me tabelat, përdoruesi përdor një ndërfaqe të qartë, përmes së cilës mund të fusë, të shqyrtojë dhe të përditësojë informacionet.

Me përdorimin e formularëve zvogëlohet mundësia e gabimeve gjatë futjes së të dhënave, ndërsa njëkohësisht përmirësohen qartësia dhe efikasiteti i punës. Për këto arsye, formularët përfaqësojnë një nga elementet më të përdorura në bazat moderne të të dhënave.



A E DI?

A e di se çdo faqe interneti për regjistrim, hyrje ose porosi online, në fakt, përdor një formular? Edhe pse përdoruesi sheh vetëm fushat për futjen e të dhënave, në prapavijë këto informacione ruhen në një bazë të të dhënave.



PYETJE PËR PËRSËRITJE

1. Çfarë përfaqëson formulari në një bazë të të dhënave?
2. Cilat janë përparësitë e përdorimit të formularëve në krahasim me punën e drejtpërdrejtë me tabela?
3. Për çfarë përdoren më së shpeshti formularët?
4. Si futen të dhënat përmes formularit?
5. Si kryhet përditësimi i regjistrimeve ekzistuese?
6. Çfarë përfaqëson navigimi nëpër regjistrime?
7. A i zëvendësojnë formularët tabelat?



DETYRA PRAKTIKE

Detyra 1

Hape tabelën ekzistuese dhe krijo një formular që do t'i shfaqë të gjitha fushat.

Detyra 2

Përmes formularit, fut të paktën pesë regjistrime të reja.

Detyra 3

Ndrysho dy regjistrime ekzistuese duke përdorur formularin.

Detyra 4

Shqyrto të gjitha regjistrimet duke përdorur butonat e navigimit.

Detyra 5

Propozo tre përmirësime të formularit që do ta bënë atë më të qartë për përdoruesit.



DETYRË HULUMTUESE

Hulumto shembuj të formularëve elektronikë që përdoren çdo ditë në internet.

Gjej të paktën tre shembuj dhe analizoji:

- Cili është qëllimi i tyre;
- Cilat të dhëna futen;
- Si janë të organizuara fushat;
- Cilat rregulla për futjen e të dhënave zbatohen.



MBAJ MEND ÇFARË MËSOVE

- Formulari paraqet një ndërfaqe përdoruesi për punë me të dhënat.
- Formularët mundësojnë futjen, shqyrtimin dhe përditësimin e regjistrimeve.
- Të dhënat vazhdojnë të ruhen në tabela.
- Formularët përmirësojnë qartësinë dhe e lehtësojnë punën.
- Elementet e navigimit mundësojnë lëvizjen nëpër regjistrime.
- Formularët përdoren gjerësisht në sistemet moderne informatike.

PËR ATA QË DËSHIROJNË TË DINË MË SHUMË



Përveç formularëve të thjeshtë, sistemet moderne mundësojnë krijimin e formularëve të ndërlikuar që përmbajnë butona, lista për zgjedhje, imazhe, llogaritje automatike dhe të dhëna të lidhura nga disa tabela. Në aplikacionet profesionale, formularët shpesh janë mënyra kryesore e komunikimit ndërmjet përdoruesit dhe bazës së të dhënave.



KONTROLLO NJOHURITË E TUA

Rretho përgjigjen e saktë

1. Formulari shërben për:

- a) fshirjen e bazës së të dhënave
- b) futjen dhe shqyrtimin e të dhënave
- c) formatimin e sistemit operativ
- ç) kompresimin e skedarëve

2. Të dhënat e futura përmes formularit ruhen në:

- a) memoria operative
- b) formulari
- c) tabela
- ç) raporti

E saktë apo e pasaktë

- Formularët mundësojnë punë më të lehtë me të dhënat.
- Të dhënat ruhen vetë në formular.
- Formularët mund të përdoren për përditësimin e regjistrimeve.
- Butonat e navigimit mundësojnë lëvizjen nëpër regjistrime.



ANALIZË

Në një bazë të të dhënave të një shkolle ekziston një tabelë me 500 nxënës.

Shpjego:

- pse përdorimi i formularit është më praktik sesa puna e drejtpërdrejtë me tabelën;
- cilat përparësi i ka përdoruesi;
- si mund të ndihmojë formulari gjatë futjes së një nxënësi të ri.

3.6



PYETËSORËT (QUERIES) – KËRKIMI DHE PËRZGJEDHJA E TË DHËNAVE

TASHMË DI

- të krijosh tabela në një bazë të të dhënave;
- të përdorësh lloje të ndryshme të të dhënave;
- të aplikosh çelësin primar;
- të futësh të dhëna përmes formularëve;
- të shqyrtosh dhe të përditësosh regjistrimet.

PYETJE PËR TË MENDUAR

1. Çfarë ndodh nëse baza ka mijëra regjistrime, ndërsa ty të duhet vetëm një?
2. Si mund të gjenden të gjithë nxënësit e një paraleleje?
3. Si mund të shfaqen vetëm nxënësit me sukses të shkëlqyeshëm?
4. A është e nevojshme të shqyrtohen gjithmonë të gjitha regjistrimet?
5. Si mund t'i gjejë kompjuteri shpejt informacionet e nevojshme?

NË PËRMBAJTJET QË VIJOJNË DO TË MËSOSH:

- shpjegosh çfarë paraqet një pyetësor;
- përdorësh pyetësorë për kërkimin e të dhënave;
- zbatosh kushte për përzgjedhjen e regjistrimeve;
- renditësh informacionet sipas kriterëve të përcaktuara;
- analizosh rezultatet e marra nga pyetësi;

Hyrje

Në bazat reale të të dhënave shpesh ruhen qindra, mijëra ose madje edhe miliona regjistrime. Edhe pse të gjitha të dhënat janë të rëndësishme, në praktikë përdoruesit më së shpeshti kanë nevojë vetëm për një pjesë të caktuar të informacionit. Për shembull, mësuesi mund të dëshirojë të shohë vetëm nxënësit e një paraleleje të caktuar, bibliotekari vetëm librat që aktualisht janë të huazuar, ndërsa administrata vetëm nxënësit që nuk e kanë paguar ekskursionin.

Shqyrtimi i të gjithë tabelës sa herë që nevojitet një informacion i caktuar do të ishte i ngadalshëm dhe joefikas. Për këtë arsye, sistemet për menaxhimin e bazave të të dhënave mundësojnë përdorimin e mekanizmave të veçantë për kërkimin dhe përzgjedhjen e informacionit. Këta mekanizma quhen pyetësorë.

Pyetësorët përfaqësojnë një mjet që mundëson që nga baza e të dhënave të nxirren vetëm informacionet që plotësojnë kushtet e përcaktuara. Në vend që përdoruesi t'i kërkojë manualisht të gjitha regjistrimet, sistemi i gjen automatikisht dhe shfaq vetëm rezultatet e nevojshme.

ÇFARË PARAQET NJË PYETËSOR?

Pyetësi paraqet një kërkesë që i dërgohet sistemit për menaxhimin e bazës së të dhënave. Me ndihmën e pyetësit, përdoruesi përcakton se cilat të dhëna dëshiron të marrë dhe sipas cilave kushte duhet të përzgjidhen.

Ndryshe nga tabela, e cila përmban të gjitha të dhënat, pyetësi zakonisht shfaq vetëm një pjesë të tyre. Për shembull, mund të shfaqen vetëm nxënësit e një paraleleje të caktuar, vetëm nxënësit me mesatare më të madhe se 4,50 ose vetëm nxënësit që marrin pjesë në gara.

Është e rëndësishme të theksohet se pyetësi nuk i ndryshon të dhënat në tabelë. Ai vetëm shfaq informacionet që plotësojnë kushtet e përcaktuara. Pasi të mbyllet pyetësi, të dhënat në tabelë mbeten të pandryshuara.

Shembull praktik

TË SHQYRTOJMË TABELËN E MËPOSHTME.

Emri	Mbiemri	Klasa	Mesatarja
Ana	Petrova	I-1	5.00
Marko	Stojanov	I-2	4.20
Elena	Nikolovska	I-1	4.80
Merita	Rexhepi	I-3	3.90

Të supozojmë se është e nevojshme të shfaqen vetëm nxënësit nga paralelja I-1.

Në vend që të kërkohet manualisht në tabelë, mund të krijohet një pyetësor që do ta vendosë kushtin.

Klasa = I-1

Si rezultat do të përftohet:

Emri	Mbiemri	Klasa	Mesatarja
Ana	Petrova	I-1	5.00
Elena	Nikolovska	I-1	4.80

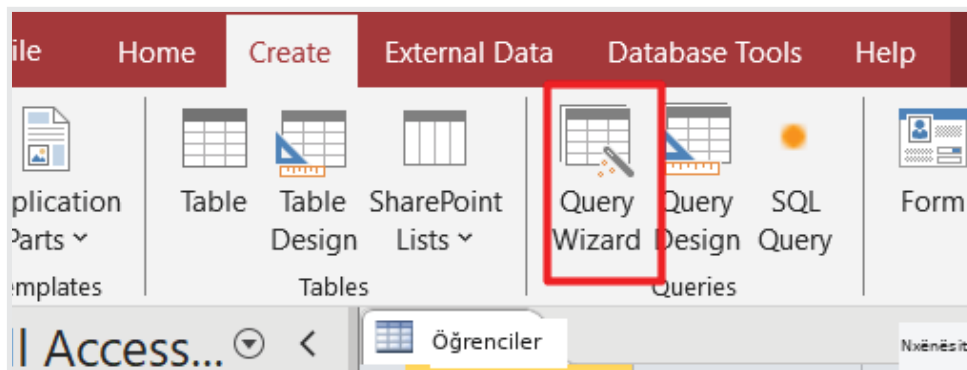


Figura 1. Pyetësor në MS Access

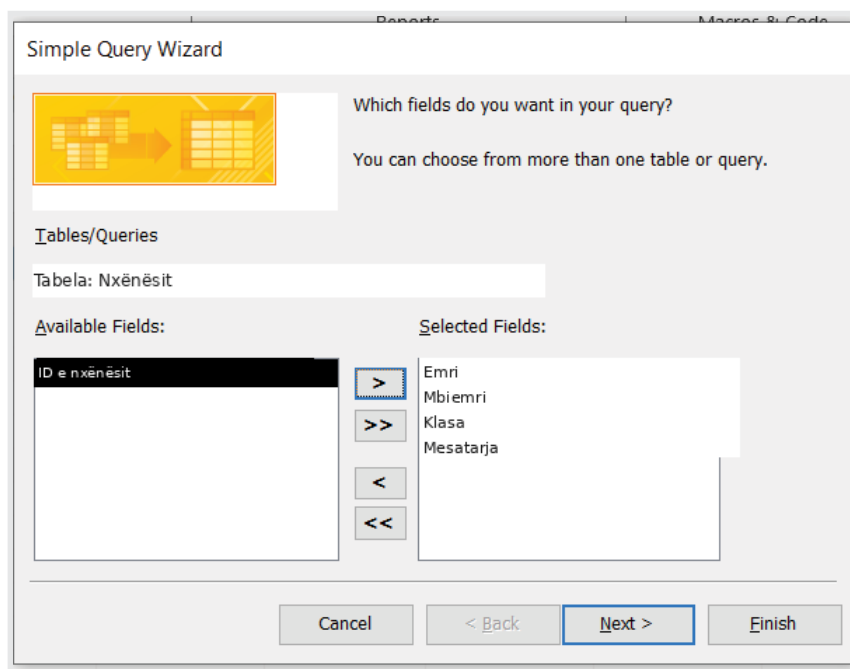


Figura 2. Përcaktimi i parametrave në pyetësor

Në raportin e krijuar më parë, i cili përmban të gjithë elementët, përmes pamjes së dizajnit mund të përcaktohen kriteret e nevojshme.

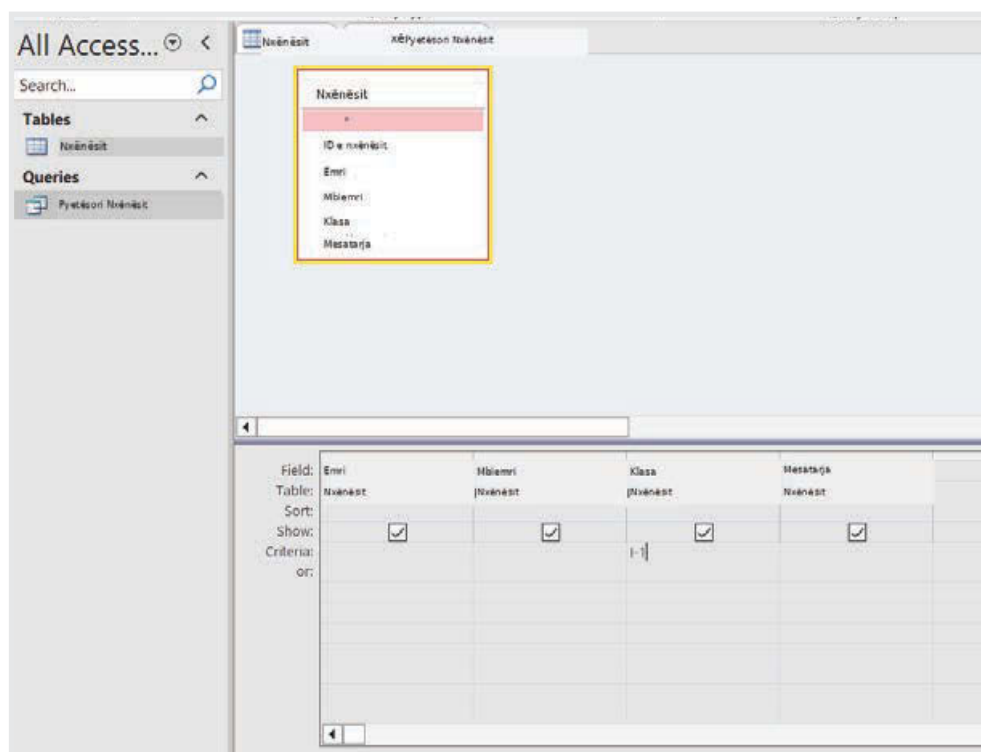


Figura 3. Përcaktimi i kriterit në pyetësor

Në këtë rast, bëhet fjalë për nxënësit e I-1 dhe kjo shënohet në fushën Kriteri, të „Klasa“. Pasi të ekzekutohet pyetësori duke zgjedhur komandën „Run“, shfaqen vetëm nxënësit e paraleles së përcaktuar, në këtë rast janë nxënësit e I-1 dhe kjo shënohet në fushën kriteri, të „klasa“. pasi të ekzekutohet pyetësori me zgjedhjen e komandës „Run“, do të shfaqen vetëm nxënësit e paraleles së përcaktuar.

Emri	Mbiemri	Klasa	Mesatarja
Ana	Petrova	I-1	5
Elena	Nikolovska	I-1	4.8
			0

Figura 4. Efekti i kriterit të vendosur në pyetësor

Ky pyetësor, i përgatitur në këtë mënyrë, i paraqet të dhënat që kërkohen nga baza. Në një bazë të të dhënave mund të krijohen një numër i pakufizuar pyetësorësh për veçori të ndryshme të bazës

PËRSE PËRDOREN PYETËSORËT?

Pyetësorët e lehtësojnë ndjeshëm përdorimin e bazave të të dhënave. Në vend që përdoruesi të punojë me një numër të madh regjistrimesh, sistemi i përzgjedh automatikisht informacionet e nevojshme. Kjo jo vetëm që kursen kohë, por edhe zvogëlon mundësinë e gabimeve.

Përveç kësaj, pyetësorët mundësojnë që të dhënat të analizohen në mënyra të ndryshme. E njëjta tabelë mund të përdoret për marrjen e informacioneve të ndryshme, në varësi të kushteve të përcaktuara. Kështu, një herë mund të kërkohen nxënësit sipas paraleles, herën tjetër sipas suksesit, ndërsa herën e tretë sipas numrit të mungesave.

Për shkak të këtyre mundësive, pyetësorët janë një nga mjetet më të dobishme gjatë punës me bazat e të dhënave dhe kanë zbatim të gjerë në sisteme të ndryshme informatike.

KUSHTET NË PYETËSORËT

Një nga përparësitë më të mëdha të pyetësorëve është mundësia e vendosjes së kushteve, sipas të cilave do të përzgjidhen të dhënat. Në vend që të shfaqen të gjitha regjistrimet nga tabela, përdoruesi mund të përcaktojë se cilat informacione dëshiron të marrë. Kjo mundëson gjetjen e shpejtë të të dhënave të nevojshme, edhe kur baza përmban një numër të madh regjistrimesh.

Kushti paraqet një rregull që regjistrimi duhet ta plotësojë në mënyrë që të shfaqet në rezultate. Për shembull, mund të kërkohen të gjithë nxënësit e një paraleleje të caktuar, të gjithë librat e një autori të caktuar ose të gjitha produktet, sasia e të cilave është më e vogël se një vlerë e përcaktuar. Sistemi i kontrollon automatikisht regjistrimet dhe shfaq vetëm ato që e plotësojnë kushtin.

Në praktikë mund të përdoren lloje të ndryshme kushtesh. Disa kushte kontrollojnë nëse vlera është e barabartë me një të dhënë të caktuar, ndërsa të tjera kontrollojnë nëse vlera është më e madhe, më e vogël ose ndodhet brenda një intervali të përcaktuar. Falë këtij fleksibiliteti, pyetësorët mund të përshtaten me nevojat e ndryshme të përdoruesve.

Shembull i përdorimit të kushtit

Të shqyrtojmë tabelën e më poshtme.

Emri	Mbiemri	Mesatarja
Ana	Petrova	5.00
Marko	Stojanov	4.20
Elena	Nikolovska	4.80
Merita	Rexhepi	3.90

Të supozojmë se është e nevojshme të shfaqen vetëm nxënësit me mesatare më të madhe se 4,50. Kushti do të jetë:

Mesatarja >4.50

Rezultati do t'i përmbajë regjistrimet e mëposhtme:

Emri	Mbiemri	Mesatarja
Ana	Petrova	5.00
Elena	Nikolovska	4.80

Nga shembulli mund të vërehet se sistemi i ka përzgjedhur automatikisht vetëm regjistrimet që e plotësojnë kushtin e vendosur.

Field:	Emri	Mbiemri	Klasa	Mesatarja	
Table:	Nxënësit	Nxënësit	Nxënësit	Nxënësit	
Sort:					
Show:	☒	☒	☒	☒	☐
Criteria:				>4.5	
or:					

Figura 5. Vendosja e kushtit për mesataren

Renditja e rezultateve

Përveç kërkimit të të dhënave, pyetësorët mundësojnë edhe renditjen e tyre. Renditja paraqet organizimin e rezultateve sipas një radhe të caktuar. Zakonisht përdoret për një shqyrtim dhe analizë më të lehtë të informacionit.

Të dhënat mund të renditen sipas rendit alfabetik, sipas vlerave numerike ose sipas datave. Për shembull, lista e nxënësve mund të renditet sipas mbiemrit, lista e produkteve sipas çmimit, ndërsa lista e librave sipas vitit të botimit.

Renditja nuk i ndryshon të dhënat në tabelë. Ajo ndikon vetëm në mënyrën se si shfaqen rezultatet në pyetësor. Në këtë mënyrë, përdoruesi mund të ketë shpejt një pasqyrë më të qartë të informacionit.

Shembulli renditjes

Të shqyrtojmë rezultatin e më poshtëm.

Emri	Mesatarja
Marko	4.20
Ana	5.00
Merita	3.90
Elena	4.80

Nëse zbatohet renditja sipas mesatares në rend zbritës, rezultati do të jetë:

Emri	Mesatarja
Ana	5.00
Elena	4.80
Marko	4.20
Merita	3.90

Në këtë mënyrë, vlerat më të larta shfaqen në fillim të listës.

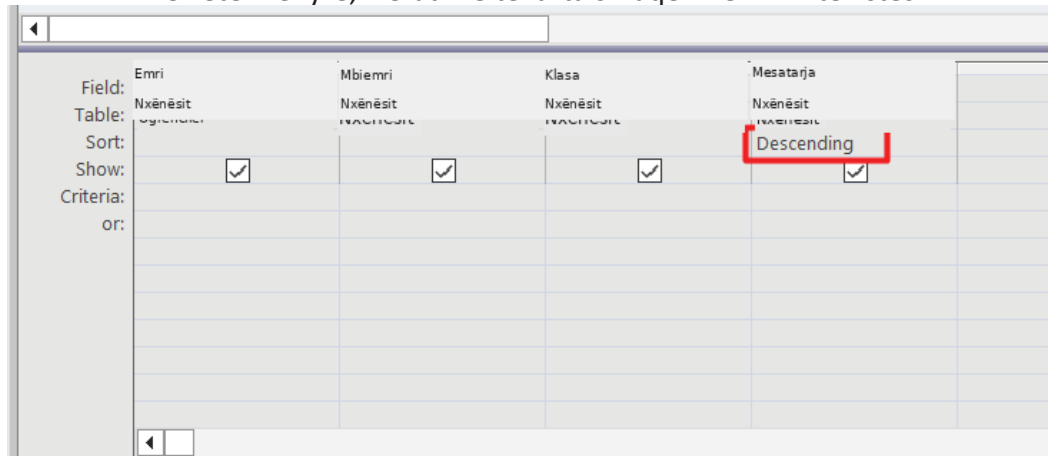


Figura 6 Renditja e rezultateve

Pas çdo ndryshimi që bëhet në pamjen e dizajnit të pyetësorit, është e nevojshme të aktivizohet komanda „Run“ për të marrë gjendjen e re të pyetësorit.

Kombinimi i më shumë kushteve

Në shumë situata, një kusht i vetëm nuk mjafton për të marrë informacionet e nevojshme. Për këtë arsye, pyetësorët mundësojnë përdorimin e disa kushteve njëkohësisht. Për shembull, mund të kërkohen nxënësit që i përkasin një paraleleje të caktuar dhe njëkohësisht kanë mesatare mbi një vlerë të caktuar.

Të supozojmë se duhet të gjenden nxënësit e paraleles I-1 me mesatare më të madhe se 4,50.

Nevojiten dy kushte:

- Paralelja = I-1
- Mesatarja > 4,50

Në rezultate do të shfaqen vetëm regjistrimet që i plotësojnë të dyja kushtet.

Kombinimi i kushteve mundëson kërkim shumë më të saktë të informacionit. Sa më saktë të përcaktohen kushtet, aq më relevante do të jenë rezultatet e marra.

Shmëbull praktik: Baza e të dhënave të shkollës

Të supozojmë se shkolla mban evidencë për të gjithë nxënësit e klasës së parë. Drejtori dëshiron të marrë një listë të nxënësve që kanë mesatare më të madhe se 4,50 dhe njëkohësisht nuk kanë më shumë se tre mungesa të pajustificuara.

Në vend që të shqyrtohen manualisht qindra regjistrime, mund të krijohet një pyetësor që automatikisht do të përzgjedhë vetëm nxënësit që i plotësojnë të dyja kushtet. Rezultati do të merret brenda pak sekondash, pa pasur nevojë për përpunim shtesë.

Situata të tilla hasen çdo ditë në arsim, shëndetësi, bankë dhe administratë. Falë pyetësorëve, përdoruesit mund të marrin shpejt informacione të sakta nga sasi të mëdha të të dhënave. Për këtë arsye, pyetësorët janë një nga mekanizmat më të fuqishëm për punën me bazat e të dhënave

Analiza e shembullit

Nga shembujt e mëparshëm mund të përfundohet se pyetësorët nuk shërbejnë vetëm për gjetjen e të dhënave, por edhe për analizimin e tyre. Ata mundësojnë përzgjedhjen e informacioneve të nevojshme, renditjen dhe kombinimin e tyre sipas kriterëve të ndryshme. Në këtë mënyrë, përdoruesi merr vetëm të dhënat relevante, në vend që të punojë me të gjithë përmbajtjen e tabelës.

Falë mundësisë për vendosjen e kushteve dhe renditjen e rezultateve, pyetësorët e përmirësojnë ndjeshëm efikasitetin e punës me bazat e të dhënave. Ata përfaqësojnë një mjet të pazëvendësueshëm në sistemet moderne informatike dhe përdoren çdo ditë për kërkimin dhe analizimin e informacionit.

Screenshot 5 –PYETËSOR ME MË SHUMË KUSHTE

Screenshot 6 – REZULTATET E PYETËSORIT



PËRFUNDIM:

Pyetësorët përfaqësojnë një mjet të rëndësishëm për punën me bazat e të dhënave, sepse mundësojnë gjetjen, përzgjedhjen dhe analizimin e shpejtë të informacionit. Në vend që përdoruesi t'i kërkojë manualisht të gjitha regjistrimet në tabelë, sistemi i përzgjedh automatikisht të dhënat që plotësojnë kushtet e përcaktuara.

Me përdorimin e pyetësorëve mund të zbatohen kriterë të ndryshme për kërkimin, të kombinohen disa kushte dhe të renditen rezultatet sipas nevojave të përdoruesit. Në këtë mënyrë përmirësohet efikasiteti i punës dhe lehtësohet marrja e vendimeve bazuar në informacione të sakta.

Për shkak të zbatimit të tyre praktik, pyetësorët përfaqësojnë një nga mekanizmat më të rëndësishëm në bazat moderne të të dhënave dhe përdoren pothuajse në të gjitha sistemet informative.



A E DI?

A e di se motorët e kërkimit në internet përdorin një parim të ngjashëm me pyetësorët? kur fut një term për kërkim, sistemi kërkon automatikisht nëpër një numër të madh të të dhënave dhe shfaq vetëm rezultatet që përputhen me kërkesën e vendosur.



PYETJE PËR PËRSËRITJE

1. Çfarë paraqet një pyetësor në një bazë të të dhënave?
2. Cili është ndryshimi ndërmjet tabelës dhe pyetësorit?
3. Çfarë paraqet një kusht në pyetësor?
4. Përse përdoret renditja e rezultateve?
5. A i ndryshon pyetësori të dhënat në tabelë?
6. Pse është i dobishëm kombinimi i disa kushteve?
7. Në cilat situata përdoren më shpesh pyetësorët?



DETYRA PRAKTIKE

Detyra 1

Krijë një pyetësor që do t'i shfaqë të gjithë nxënësit e një paraleleje të zgjedhur.

Detyra 2

Krijë një pyetësor që do t'i shfaqë nxënësit me mesatare më të madhe se 4,50.

Detyra 3

Rendit rezultatet e pyetësorit të mëparshëm sipas mesatares në rend zbritës.

Detyra 4

Krijë një pyetësor që do t'i shfaqë nxënësit e paralelës I-1 me mesatare më të madhe se 4,50.

Detyra 5



DETYRË HULUMTUESE

Krijë një pyetësor që do t'i shfaqë vetëm nxënësit që kanë më shumë se pesë mungesa.

Në shkollë ekzistojnë të dhëna për nxënësit, mësuesit, lëndët dhe notat.

Propozo të paktën pesë pyetje të cilave drejtori mund t'u marrë përgjigje duke përdorur pyetësorët.

Për çdo pyetje, jep:

- të dhënat që nevojiten;
- kushtet që duhet të vendosen;
- çfarë rezultati pritet.



MBAJ MEND ÇFARË MËSOVE

- Pyetësori paraqet një mjet për kërkimin dhe përzgjedhjen e të dhënave.
- Pyetësorët mundësojnë shfaqjen vetëm të informacioneve të nevojshme.
- Kushtet përcaktojnë se cilat regjistrime do të shfaqen.
- Është i mundur përdorimi i disa kushteve njëkohësisht.
- Rezultatet mund të renditen sipas kriterëve të ndryshme.
- Pyetësorët nuk i ndryshojnë të dhënat në tabelë.
- Pyetësorët e përmirësojnë ndjeshëm efikasitetin e punës me bazat e të dhënave.



PËR ATA QË DËSHIROJNË TË DINË MË SHUMË

Në sistemet moderne për menaxhimin e bazave të të dhënave ekzistojnë pyetësorë të avancuar, të cilët mund të kryejnë llogaritje statistikore, të grupojnë të dhënat dhe të analizojnë informacionet nga disa tabela të ndërlidhura. Në sistemet e mëdha të informacionit, këta pyetësorë përdoren shpesh për përgatitjen e raporteve, analizën e shitjeve, monitorimin e të dhënave financiare dhe mbështetjen e marrjes së vendimeve afariste.





KONTROLLO NJOHURITË E TUA

Rretho përgjigjen e saktë

1. Pyetësi shërben për:

- a) Formatimin e tabelave
- b) Kërkimin dhe përzgjedhjen e të dhënave
- c) Instalimin e programeve
- d) Krijimin e një sistemi operativ

2. Kushti në pyetësor:

- a) i fshin të dhënat
- b) përcakton se cilët regjistra do të shfaqen
- c) e ndryshon strukturën e tabelës
- d) e mbyll bazën e të dhënave

3. Renditja përdoret për:

- a) renditjen e rezultateve
- b) fshirjen e rezultateve
- c) krijimin e formularëve
- d) shtimin e fushave të reja

E saktë apo e pasaktë

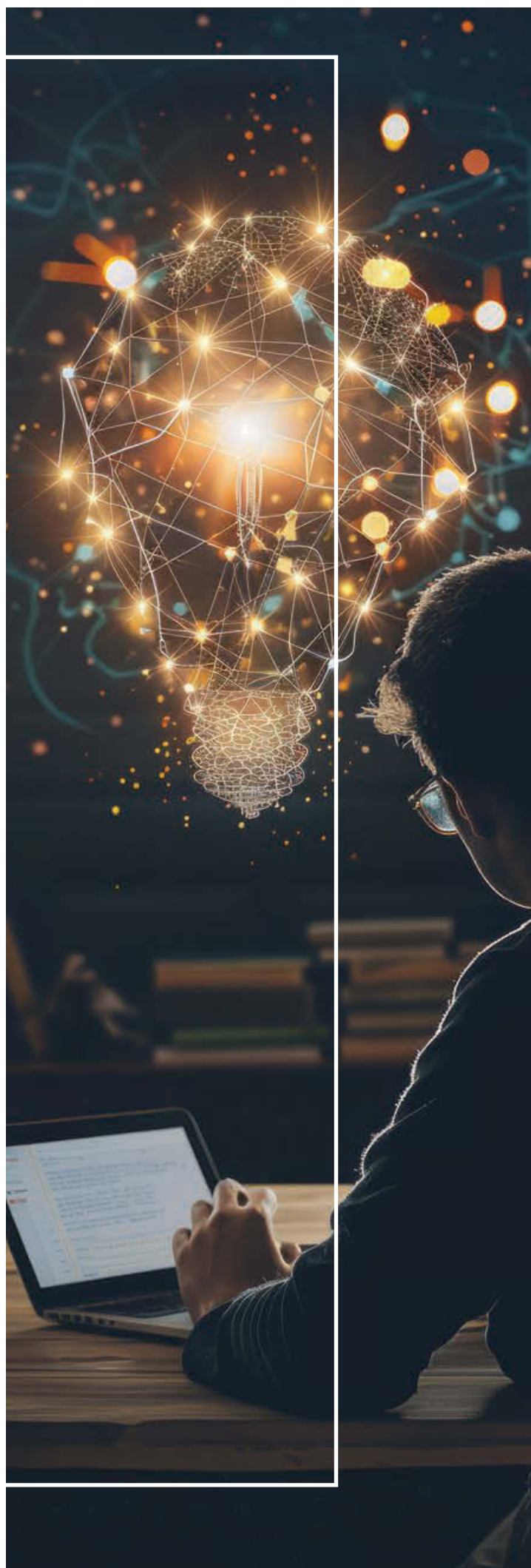
- Pyetësorët mund të përdorin disa kushte njëkohësisht.
- Rezultatet e pyetësorit mund të renditen.
- Pyetësi gjithmonë i shfaq të gjitha regjistrimet e tabelës.
- Pyetësi i ndryshon të dhënat në tabelë.

ANALIZË

Në një bazë të të dhënave për një bibliotekë ruhen informacione për librat, autorët dhe përdoruesit.

Shpjego:

- çfarë pyetësi do të krijojë për të shfaqur të gjithë librat e një autori të caktuar;
- çfarë kushti do të vendosje;
- si do t'i renditje rezultatet që të ishin më të qarta dhe më të lehta për t'u parë.





3.7

RAPORTET (REPORTS) – SHFAQJA DHE PRINTIMI I TË DHËNAVE

TASHMË DI ...

- të krijosh tabela në një bazë të të dhënave;
- të përdorësh lloje të ndryshme të të dhënave;
- të vendosësh të dhëna përmes formularëve;
- të krijosh pyetësorë për kërkimin e informacioneve;
- të përdorësh kushte dhe renditjen e rezultateve;
- të analizosh të dhënat e marra nga baza e të dhënave.

PYETJE PËR TË MENDUAR

- Si mund të përgatiten për printim të dhënat nga baza e të dhënave?
- A është tabela gjithmonë mënyra më e mirë për paraqitjen e informacioneve?
- Si mund të krijohet një listë e nxënësve e përgatitur për printim?
- Si i përgatisin institucionet raportet për punën e tyre?
- Në ç'mënyrë mund të paraqiten më qartë rezultatet e pyetësorëve?

NË PËRMBAJTJEN QË VIJON DO TË MËSOSH:

- të shpjegosh se çfarë përfaqëson një raport;
- të renditësh përparësitë e përdorimit të raporteve;
- të krijosh një raport të thjeshtë;
- të përdorësh të dhëna nga tabelat dhe pyetësorët në raporte;
- të rregullosh pamjen e një raporti;
- të analizosh zbatimin e raporteve në sistemet informatike.

Gjatë punës me të dhëna, të dhënat organizohen në tabela; ato mund të vendosen përmes formularëve dhe mund të kërkohen me ndihmën e pyetësorëve. Megjithatë, në shumë situata është e nevojshme që informacionet e marra të paraqiten në një mënyrë më të qartë ose të përgatiten për printim. Për shembull, mësuesi mund të dëshirojë të printojë një listë të nxënësve, bibliotekari një raport për librat e huazuar, ndërsa drejtori një pasqyrë të suksesit sipas paraleleve.

Edhe pse tabelat dhe pyetësorët përmbajnë të dhënat e nevojshme, pamja e tyre nuk është gjithmonë e përshtatshme për prezantim ose printim. Për këtë arsye, sistemet për menaxhimin e bazave të të dhënave mundësojnë përdorimin e raporteve. Raportet shërbejnë për paraqitjen e organizuar dhe profesionale të informacioneve.

Raporti paraqet një objekt të veçantë në bazën e të dhënave, i cili i shfaq të dhënat në një formë të përgatitur për shqyrtim, printim ose shpërndarje. Ai mund të përmbajë tituj, nëntituj, informacione të grupuara, numra të faqeve dhe elemente të tjera që përmirësojnë qartësinë e paraqitjes.

Çfarë përfaqëson një raport?

Raporti është një paraqitje e organizuar e të dhënave që gjenden në bazën e të dhënave. Ndryshe nga formularët, të cilët më së shpeshti përdoren për futjen dhe përpunimin e informacioneve, raportet janë të destinuara për shfaqjen dhe prezantimin e të dhënave.

Raportet mund të përdorin informacione drejtpërdrejt nga tabelat ose nga pyetësorët e krijuar më parë. Në këtë mënyrë, përdoruesi mund të shfaqë vetëm një pjesë të caktuar të të dhënave që nevojiten për një analizë ose prezantim konkret.

Në praktikë, raportet përdoren shpesh për përgatitjen e listave, pasqyrave statistikore, rezultateve të analizave, raporteve financiare dhe dokumenteve të tjera që duhet të jenë të organizuara qartë dhe të lehta për t'u lexuar.

Shembull praktik

Le të supozojmë se shkolla dëshiron të përgatisë një listë të nxënësve të paraleles I-1.

Të dhënat fillimisht ruhen në një tabelë, më pas, përmes një pyetësi, përzgjidhen vetëm nxënësit e paraleles së zgjedhur.

Rezultatet e përfituara mund të duken kështu:

Emri	Mbiemri	Klasa
Ana	Petrova	I-1
Elena	Nikolovska	I-1
Merita	Rexhepi	I-1

Edhe pse informacionet janë të sakta, një paraqitje e tillë nuk është më e përshtatshme për printim dhe arkivim. Për këtë arsye, krijohet një raport që do të përmbajë titullin, datën e përgatitjes dhe të dhënat e renditura në mënyrë të rregullt.

Nxënësit					
ID e nxënësit	Emri	Mbiemri	Klasa	Mesatarja	Click to Add
1	Ana	Petrova	I-1	5	
2	Marko	Stojanov	I-2	4.2	
3	Elena	Nikolovska	I-1	4.8	
4	Merita	Rexhepi	I-1	5	

Figura 1. Të dhënat burimore për raportin

Përparësitë e raporteve

Raportet mundësojnë që informacionet të paraqiten në mënyrë më të qartë dhe profesionale. Ato mund të përmbajnë elemente të ndryshme që përmirësojnë lexueshmërinë, si tituj, logo, numra faqesh dhe të dhëna të grupuara.

Një përparësi tjetër është mundësia e printimit të informacioneve. Në vend që përdoruesi të printojë tabela ose rezultate nga pyetësorët, ai mund të përgatisë një raport që është krijuar posaçërisht për këtë qëllim.

Raportet mundësojnë gjithashtu një paraqitje më të mirë të informacioneve. Të dhënat mund të organizohen sipas kriterëve të ndryshme, ndërsa përdoruesi mund t'i gjejë lehtësisht informacionet e nevojshme.

Krijimi i një raporti

Në Microsoft Access dhe LibreOffice Base ekzistojnë mjete që mundësojnë krijimin automatik të raporteve. Fillimisht, përdoruesi zgjedh një tabelë ose pyetësor si burim të të dhënave, dhe më pas programi gjeneron një raport bazë.

Pas krijimit, raporti mund të përpunohet më tej. Është e mundur të ndryshohet renditja e të dhënave, të shtohen tituj dhe të zgjidhet mënyra e paraqitjes së informacioneve. Në këtë mënyrë, raporti përshtatet me nevojat konkrete të përdoruesit.

Në institucionet arsimore, raportet përdoren shpesh për përgatitjen e listave të nxënësve dhe raporteve për suksesin, statistikave për mungesat dhe të dhënave të tjera që përdoren rregullisht në punë.

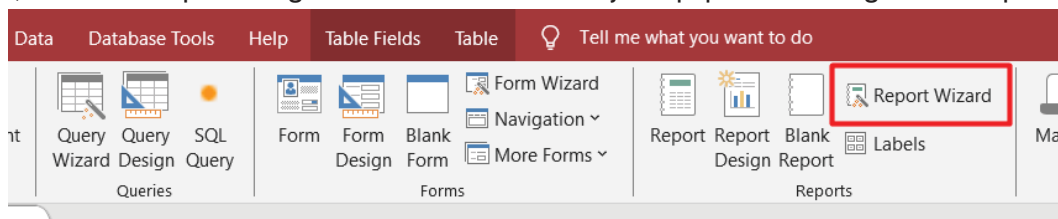


Figura 2. Krijimi i një raporti me Report Wizard

Rregullimi i pamjes së raportit

Pas krijimit të raportit, pamja e tij mund të përmirësohet më tej. Përdoruesi mund të ndryshojë renditjen e fushave, të shtojë tituj dhe nëntituj, të vendosë numrat e faqeve dhe të zgjedhë një paraqitje të përshtatshme për printim.

Një raport i organizuar mirë mundëson që të dhënat të lexohen dhe të analizohen më lehtë. Veçanërisht te raportet që përmbajnë një numër të madh regjistrimesh, rregullimi i duhur ka rëndësi të madhe për përdorshmërinë e tyre.

Në shumë raste, raportet përmbajnë edhe informacione shtesë, si: datën e përgatitjes, emrin e institucionit ose një përshkrim të shkurtër të përmbajtjes. Këto elemente kontribuojnë në një organizim më të mirë dhe në një pamje më profesionale të dokumentit.

Shembull praktik: Raport për suksesin e nxënësve

Le të supozojmë se duhet të përgatitet një raport për nxënësit që kanë mesatare më të lartë se 4,50.

Fillimisht krijohet një pyetësor që përzgjedh nxënësit të cilët e plotësojnë këtë kusht. Më pas, rezultatet e marra përdoren si bazë për raportin. Fillimisht krijohet një pyetësor në të cilin kriteri është që mesatarja të jetë 4,50. Ky pyetësor më pas përzgjidhet gjatë krijimit të raportit.

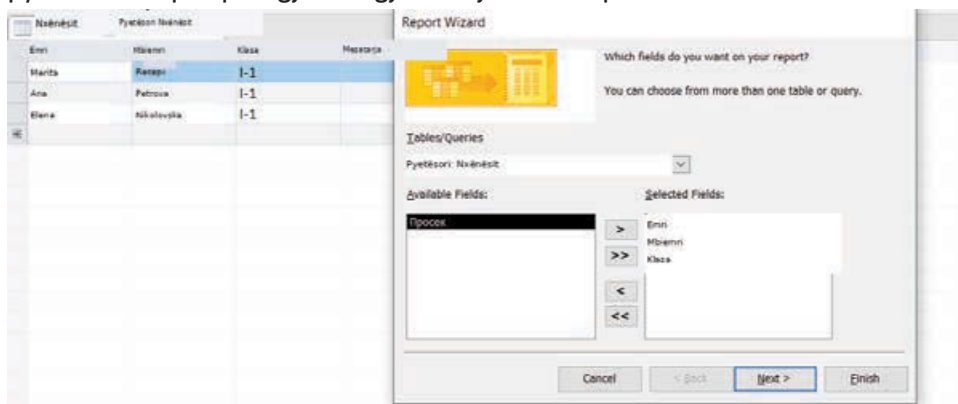


Figura 2. Përgatitja e raportit

Në raport mund të paraqiten emri, mbiemri, paralelja dhe suktesi mesatar i secilit nxënës. Gjithashtu, mund të shtohet titulli „Nxënës me sukses të shkëlqyer“ dhe data e përgatitjes së dokumentit.

Në këtë mënyrë, informacionet e marra bëhen më të qarta dhe përgatiten për printim ose prezantim para këshillit të mësuesve, kujdestarëve të klasave ose drejtimit të shkollës.

Mbiemri	Klasa	Nota	Mesatarja
Reçepi	I-1		5
Petrova	I-1		5
Nikolovska	I-1		4.8

Figura 3. Raport i përgatitur për printim

Analiza e shembullit

Nga shembulli i mëparshëm mund të vërehet se raporti nuk paraqet një burim të ri të të dhënave, por një paraqitje të organizuar të informacioneve ekzistuese. Të dhënat mbeten në tabela dhe në pyetësorë, ndërsa raporti shërben për prezantimin e tyre.

Me përdorimin e raporteve përmirësohet lexueshmëria e informacioneve dhe lehtësohet përdorimi i tyre në punën e përditshme. Në vend që përdoruesi të punojë me tabela dhe pyetësorë të ndërlikuar, ai merr një dokument të përgatitur për shqyrtim dhe printim.

Për këto arsye, raportet janë një pjesë e rëndësishme e bazave moderne të të dhënave dhe kanë zbatim të gjerë në arsim, administratë, shëndetësi, institucionet financiare dhe në shumë fusha të tjera.



PËRFUNDIM:

Raportet përfaqësojnë një element të rëndësishëm të bazave të të dhënave dhe mundësojnë që informacionet të paraqiten në mënyrë të organizuar dhe të qartë. Ndryshe nga tabelat dhe pyetësorët, të cilët përdoren për ruajtjen dhe kërkimin e të dhënave, raportet janë të destinuara për prezantimin dhe printimin e tyre.

Me përdorimin e raporteve mund të përgatiten lista, analiza dhe pasqyra të ndryshme, të cilat përdoren në punën e përditshme të institucioneve dhe organizatave. Falë mundësisë për rregullimin e pamjes, raportet mundësojnë që informacionet të paraqiten qartë dhe në mënyrë profesionale.

Për shkak të zbatimit të tyre praktik, raportet janë një pjesë e rëndësishme e çdo baze të të dhënave dhe përdoren shpesh për përgatitjen e dokumenteve që duhet të shqyrtohen, të arkivohen ose të printohen.



A E DI?

A e di se pjesa më e madhe e dokumenteve që marrin drejtorët, menaxherët dhe drejtuesit në institucione të ndryshme gjenerohen automatikisht nga bazat e të dhënave? Në vend që të përgatiten manualisht, raportet krijohen për disa sekonda, bazuar në të dhënat që tashmë gjenden në sistem.



PYETJE PËR PËRSËRITJE

Çfarë përfaqëson një raport në një bazë të të dhënave?

Cili është ndryshimi midis formularit dhe raportit?

Nga cilat burime mund të merren të dhënat për një raport?

Për çfarë përdoren raportet?

Cilat janë përparësitë e raporteve gjatë printimit të të dhënave?

Si mund të përmirësohet pamja e raportit?

Në cilat fusha përdoren më shpesh raportet?



DETYRA PRAKTIKE

Detyra 1

Krijoni një raport në bazë të tabelës me nxënësit.

Në raport paraqit:

- emri;
- mbiemri;
- klasa.

Detyra 2

Krijoni një pyetësor që do të shfaqë nxënësit me mesatare më të lartë se 4,50 dhe, në bazë të tij, përgatit një raport.

Detyra 3

Shtoni një titull në raport:

„Lista e nxënësve me sukses të shkëlqyer“

Detyra 4

Shtoni datën e përgatitjes dhe numrat e faqeve në raport.

Detyra 5

Shtype ose eksporto raportin në formatin PDF.



DETYRË HULUMTUESE

Hulumto se në cilat situata shkollat përdorin raporte.

Jep të paktën pesë shembuj dhe shpjegoji:

- cili është qëllimi i tyre;
- cilat të dhëna përdoren;
- kush do t'i përdorte këto raporte.



MBAJ MEND ÇFARË MËSOVE

- Raporti shërben për paraqitjen e organizuar të të dhënave.
- Raportet mund të përdorin të dhëna nga tabelat dhe pyetësorët.
- Ato janë të destinuara për shqyrtimin, shtypjen dhe prezantimin e informacioneve.
- Pamja e raportit mund të rregullohet dhe formatohet më tej.
- Raportet nuk përmbajnë të dhëna të reja, por paraqesin informacionet ekzistuese.
- Raportet përdoren gjerësisht në sistemet moderne të informacionit.

PËR ATA QË DËSHIROJNË TË DINË MË SHUMË



Në sistemet profesionale të informacionit, raportet shpesh përmbajnë grafikë, analiza statistikore dhe llogaritje automatike. Për shembull, një raport financiar mund të llogarisë automatikisht të ardhurat dhe shpenzimet, ndërsa një raport i shitjeve mund të paraqesë tendencat dhe krahasimet ndërmjet periudhave të ndryshme.



KONTROLLO NJOHURITË E TUA

Rretho përgjigjen e saktë

1. Raporti zakonisht përdoret për:

- a) Futjen e të dhënave
- b) Paraqitjen dhe shtypjen e të dhënave
- c) Krijimin e një sistemi operativ
- ç) Fshirjen e tabelave

2. Raporti mund të përdorë të dhëna nga:

- a) Tabelë
- b) Pyetësor
- c) Tabelë dhe pyetësor
- ç) Asnjëra nga të lartpërmendurat

3. Raportet përdoren kur:

- a) Duhet të organizohen të dhënat për prezantim
- b) Duhet të instalohet një program
- c) Duhet të krijohet një sistem operativ
- ç) Duhet të formatohet disku

E saktë ose e pasaktë

- Raportet mund të shtypen.
- Raportet shërbejnë për futjen e të dhënave të reja.
- Raportet mund të përmbajnë tituj dhe numra të faqeve.
- Raportet mund të përdorin të dhëna nga pyetësorët.



ANALIZË

Në një bazë të dhënash shkollore është e nevojshme të përgatitet një listë e nxënësve me sukses të shkëlqyer.

Shpjego:

- cilat të dhëna do të jenë të nevojshme;
- nëse fillimisht duhet të krijohet një pyetësor;
- pse raporti është një zgjidhje më e mirë se një tabelë e zakonshme për shtypjen e listës.



3.8

KRIJIMI I LETRAVE QARKORE ME BAZË TË TË DHËNAVE

TASHMË DI ...

- të krijosh tabela në bazën e të dhënave;
- të futësh dhe përditësosh të dhëna;
- të përdorësh formularë për punën me të dhënat;
- të krijosh pyetësorë për kërkimin e informacionit;
- të përgatitësh raporte;
- të organizosh të dhënat në bazën e të dhënave.

PYETJE PËR TË MENDUAR

1. Si mund t'u dërgohet i njëjti dokument një numri të madh personash të ndryshëm?
2. A është praktike që emri dhe mbiemri të futen manualisht në çdo ftesë?
3. Si mund të shtypë shkolla qindra vërtetime për nxënësit?
4. Si u dërgojnë kompanitë letra të personalizuar klientëve të tyre?
5. A është e mundur që një dokument të përshtatet automatikisht për çdo marrës?

NË PËRMBAJTJET QË VIJOJNË DO TË MËSOSH:

1. të shpjegosh se çfarë është një letër qarkore;
2. të përdorësh të dhëna nga baza e të dhënave në një dokument;
3. të lidhësh dokumentin me burimin e të dhënave;
4. të vendosësh fushat për bashkim;
5. të gjenerosh disa dokumente nga një shabllon;
6. të analizosh përdorimin e letrave qarkore në praktikë.

Në shumë situata është e nevojshme që i njëjti dokument t'u dërgohet një numri të madh marrësish. Për shembull, shkolla mund të dërgojë ftesa për mbledhje me prindër, vërtetime për nxënësit ose mirënjohje për pjesëmarrësit në gara. Edhe pse përmbajtja e dokumentit është pothuajse e njëjtë, disa informacione, si emri, mbiemri ose adresa, janë të ndryshme për çdo marrës.

Kur çdo dokument do të përgatitej veçmas, kjo do të kërkonte shumë kohë dhe do të rriste mundësinë për shfaqjen e gabimeve. Për këtë arsye, programet moderne mundësojnë kombinimin automatik të një dokumenti me të dhëna nga baza e të dhënave ose nga një tabelë. Në këtë mënyrë, një shabllon mund të përdoret për krijimin e një numri të madh dokumentesh të personalizuar.

Ky proces quhet letër qarkore. Ai mundëson futjen automatike të të dhënave nga baza e të dhënave në një dokument të përgatitur paraprakisht. Si rezultat, krijohen disa dokumente që kanë të njëjtën përmbajtje, por përmbajnë të dhëna personale të ndryshme.

Çfarë është një letër qarkore?

Letra qarkore është një dokument që lidhet me një burim të dhënash. Burimi i të dhënave zakonisht është një bazë të dhënash, tabelë ose listë me informacione për marrësit. Në dokument vendosen fusha të veçanta, të cilat më pas zëvendësohen automatikisht me të dhënat reale.

Për shembull, në vend që në dokument të shkruhet një emër i caktuar, mund të vendoset fusha „Emri“. Gjatë gjenerimit të dokumenteve, programi automatikisht do ta zëvendësojë këtë fushë me emrin e secilit marrës.

Falë kësaj mënyre pune, një dokument mund të përdoret për krijimin e dhjetëra, qindra apo edhe mijëra letrave të personalizuar. Kjo e përmirëson ndjeshëm efikasitetin e punës dhe mundëson përgatitjen e shpejtë të një numri të madh dokumentesh.

Shembull praktik

Ta zëmë se shkolla organizon një garë në informatikë dhe duhet t’u dërgojë ftesa nxënësve. Në bazën e të dhënave ekziston tabela:

Emri	Mbiemri	Klasa
Ana	Petrova	II-1
Marko	Stojanov	II-2
Elena	Nikolovska	II-1

Në programin për përpunimin e tekstit përgatitet shablloni i mëposhtëm:

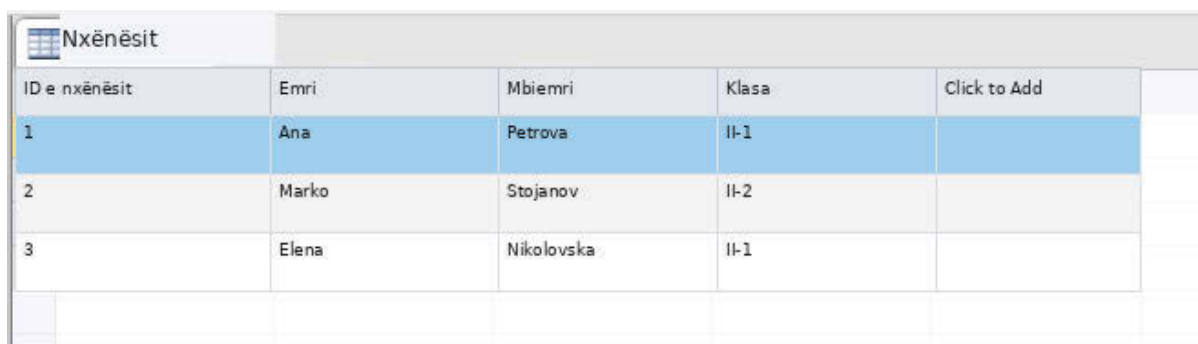
I nderuar/e <<Emri>> <<Mbiemri>>,

Ju ftojme të merrni pjesë në garën shkollore të informatikës.

Me respekt,

Këshilli organizativ

Pas bashkimit të dokumentit me bazën e të dhënave, automatikisht do të krijohen ftesa të veçanta për çdo nxënës.



Nxënësit				
ID e nxënësit	Emri	Mbiemri	Klasa	Click to Add
1	Ana	Petrova	II-1	
2	Marko	Stojanov	II-2	
3	Elena	Nikolovska	II-1	

Figura 1. Tabelë me të dhëna për marrësit

Lidhja e dokumentit me bazën e të dhënave

Për të krijuar një letër qarkore, fillimisht dokumenti duhet të lidhet me burimin e të dhënave. Burimi mund të jetë një bazë të dhënash, tabelë ose strukturë tjetër që përmban informacione për marrësit.

Pasi të vendoset lidhja, programi merr qasje në të gjitha regjistrimet që ndodhen në bazën e të dhënave. Në këtë mënyrë, të dhënat mund të përdoren automatikisht gjatë krijimit të dokumenteve.

Në programet moderne, kjo procedurë zakonisht kryhet me ndihmën e një mjeti të veçantë, të ashtuquajtur magjistari (Wizard), i cili e udhëzon përdoruesin gjatë procesit të lidhjes dhe përzgjedhjes së fushave të nevojshme. Para fillimit të procedurës, tabelat duhet të jenë të mbyllura. Në fakt, kjo mënyrë e veçantë e bazës lejon krijimin e komponentëve të tjera të dokumenteve „pasive“. Nuk mund të krijohen letra nga bazat e të dhënave në të cilat po punohet në atë moment. Nëse tabela është aktive (e hapur), magjistari nuk do të mund të krijojë letrën. Tabela duhet vetëm të jetë e përzgjedhur (e klikuar), por jo e hapur. Në disa versione të softuerit, madje është e nevojshme fillimisht të mbyllet e gjithë baza e të dhënave, në mënyrë që të mund të krijohet bashkimi me përpunuesin e tekstit.



Figura 2. Lidhja me dokumentin tekstual

Fushat për bashkim

Pasi dokumenti të jetë lidhur me bazën e të dhënave, është e nevojshme të përcaktohen vendet ku do të shfaqen të dhënat nga regjistrimet. Për këtë qëllim përdoren fushat për bashkim.

Fushat për bashkim janë shenja që tregojnë vendin ku programi do të vendosë të dhënat nga baza e të dhënave. Çdo fushë është e lidhur me një kolonë të caktuar të tabelës.

Për shembull, fusha <<Emri>> është e lidhur me kolonën Emri, ndërsa fusha <<Mbiemri>> me kolonën Mbiemri. Gjatë gjenerimit të dokumenteve, programi i zëvendëson automatikisht këto shenja me vlerat përkatëse nga çdo regjistrim.

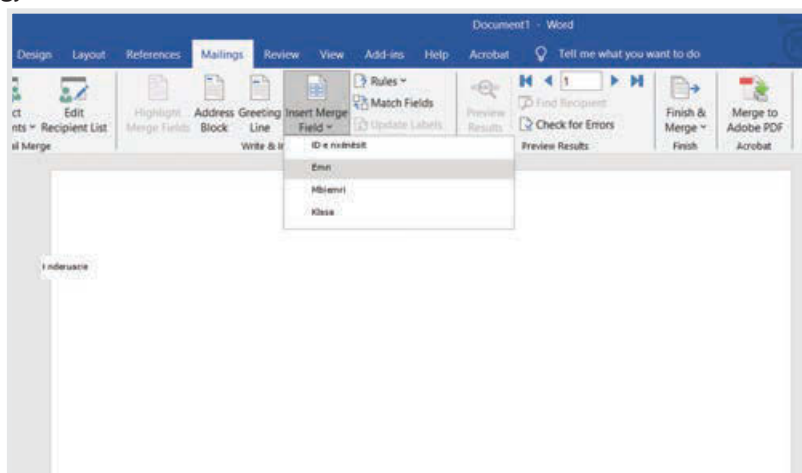


Figura 3. Futja e fushave për bashkim (Merge Fields)

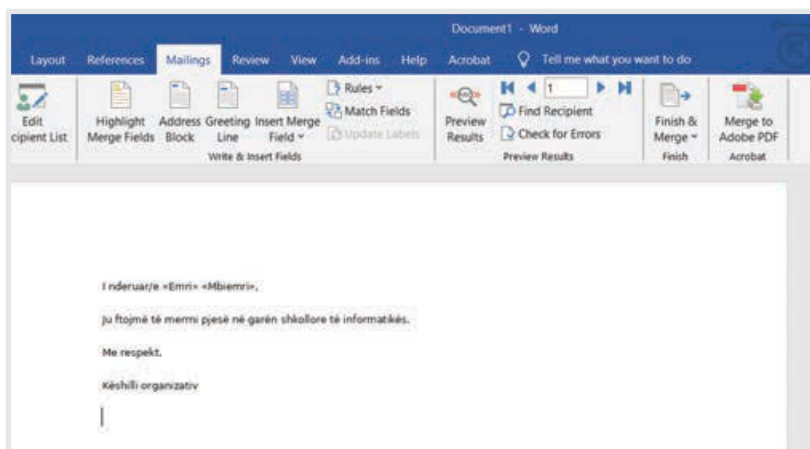


Figura 4. Letra e kompletuar

Gjenerimi i letrave qarkore

Pasi të jenë vendosur të gjitha fushat për bashkim, programi mund t'i gjenerojë automatikisht dokumentet. Për çdo regjistrim nga baza e të dhënave krijohet një version i veçantë i dokumentit.

Nëse baza e të dhënave përmban dhjetë regjistrime, do të krijohen dhjetë dokumente. Nëse përmban njëqind regjistrime, do të krijohen njëqind dokumente. Përdoruesi nuk ka nevojë t'i ndryshojë manualisht emrat dhe të dhënat e tjera.

Kjo mundësi e përshpejton ndjeshëm punën dhe mundëson përgatitjen e shpejtë të ftesave, vërtetimeve, certifikatave, njoftimeve dhe dokumenteve të tjera të personalizuar, të cilat përmbajnë të dhëna për persona të shumtë.

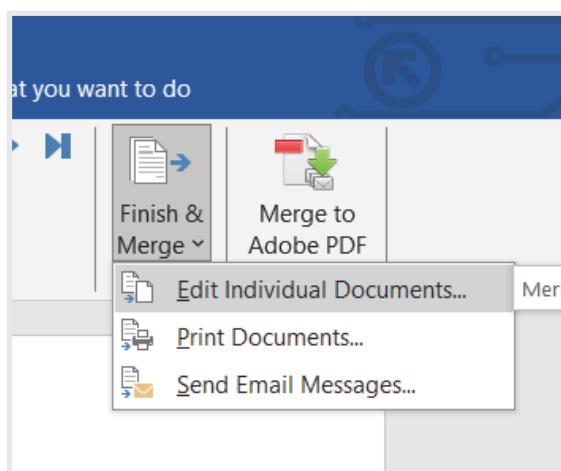


Figura 5. Gjenerimi i dokumenteve

Për të përfunduar të gjithë procedurën, është e nevojshme të bëhet bashkimi me ikonën e paraqitur në Figurën 5 dhe kjo të zbatohet për të gjitha regjistrimet që na nevojiten.

Shembull praktik: Certifikata për garën

Të supozojmë se shkolla organizon një garë në programim dhe se është e nevojshme të përgatiten certifikata për të gjithë pjesëmarrësit.

Në bazën e të dhënave ruhen emrat dhe mbiemrat e nxënësve. Në vend që çdo certifikatë të përpunohet veçmas, mund të përgatitet një shabllon në të cilin do të vendosen fushat për bashkim.

Gjatë gjenerimit të letrës qarkore, programi do të krijojë automatikisht një certifikatë për çdo nxënës. Në këtë mënyrë, për disa sekonda mund të përgatiten një numër i madh dokumentesh me pamje profesionale.

Analiza e shembullit

Nga shembulli i mëparshëm mund të vërehet se letrat qarkore përfaqësojnë një lidhje ndërmjet bazave të të dhënave dhe programeve për përpunimin e tekstit. Të dhënat mbeten në bazën e të dhënave, ndërsa dokumenti shërben si shabllon që plotësohet automatikisht me informacionet e nevojshme.

Me përdorimin e letrave qarkore zvogëlohet koha e nevojshme për përgatitjen e dokumenteve, zvogëlohet mundësia për gabime dhe sigurohet një pamje e njëjtë e të gjitha dokumenteve. Për këto arsye, kjo teknikë përdoret gjerësisht në arsim, administratë dhe në veprimtarinë afariste.

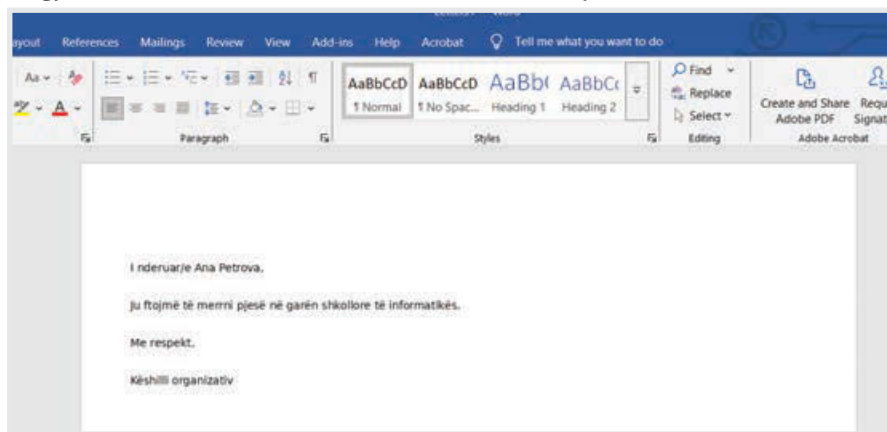


Figura 6. Dokumenti përfundimtar i personalizuar



PËRFUNDIM:

Letrat qarkore përfaqësojnë një mënyrë efikase për krijimin automatik të një numri të madh dokumentesh që kanë të njëjtën përmbajtje, por përmbajnë të dhëna të ndryshme për marrësit. Në vend që çdo dokument të përpunohet veçmas, përdoruesi përgatit një shabllon që lidhet me bazën e të dhënave ose me një tabelë.

Duke përdorur fushat për bashkim, programi i vendos automatikisht informacionet e nevojshme në dokument. Në këtë mënyrë kursehet kohë, zvogëlohet mundësia për gabime dhe sigurohet një pamje e njëjtë për të gjitha dokumentet.

Letrat qarkore përdoren gjerësisht në arsim, administratë dhe në veprimtarinë afariste, ku shpesh është e nevojshme të krijohen një numër i madh dokumentesh të personalizuar.



A E DI?

A e di se një pjesë e madhe e certifikatave, diplomave, ftesave dhe njoftimeve që marrin nxënësit dhe të punësuarit sot krijohen automatikisht me ndihmën e letrave qarkore? Në vend që çdo dokument të plotësohet manualisht, të dhënat merren nga baza e të dhënave dhe futen në një shabllon të përgatitur.



PYETJE PËR PËRSËRITJE

1. Çfarë është një letër qarkore?
2. Cilat janë përparësitë e letrave qarkore?
3. Çfarë përfaqëson burimi i të dhënave?
4. Si lidhet dokumenti me bazën e të dhënave?
5. Çfarë janë fushat për bashkim?
6. Për çfarë përdoren fushat për bashkim?
7. Në cilat situata përdoren më shpesh letrat qarkore?



DETYRA PRAKTIKE

Detyra 1

Krijoni një tabelë me fushat e mëposhtme:

- Emri
- Mbiemri
- Paralelja

Fut të dhëna për të paktën pesë nxënës.

Detyra 2

Përgatit një shabllon për ftesë për një garë shkollore.

Në shabllon parashiko vend për:

- emri;
- mbiemri;
- klasa.

Detyra 3

Lidhe dokumentin me tabelën dhe vendos fushat përkatëse për bashkim.

Detyra 4

Gjenero letra qarkore për të gjithë nxënësit nga tabela.

Detyra 5

Përgatit një shabllon për certifikatë që do t'i përdorë automatikisht të dhënat nga baza e të dhënave.



DETYRË HULUMTUESE

Hulumto se në cilat fusha përdoren më shpesh letrat qarkore.

Për secilën fushë përmend:

- llojin e dokumentit;
- burimi i të dhënave;
- përfitimet nga përdorimi i shkresave qarkore.



MBAJ MEND ATË QË MËSOVE

- Shkresa qarkore përbën një dokument të lidhur me burimin e të dhënave.
- Burimi i të dhënave mund të jetë një bazë të dhënash ose një tabelë.
- Fushat e bashkimit përdoren për futjen automatike të të dhënave.
- Një shabllon mund të përdoret për të krijuar një numër të madh dokumentesh të ndryshme.
- Letrat qarkulluese kursejnë kohë dhe zvogëlojnë mundësinë e gabimeve.
- Kjo teknikë përdoret për ftesa, certifikata, vërtetime dhe njoftime.





PËR ATA QË DËSHIROJNË TË DINË MË SHUMË

Sistemet bashkëkohore mundësojnë që letrat qarkulluese të përdoren jo vetëm për shtypjen e dokumenteve, por edhe për dërgimin automatik të postës elektronike. Në këtë mënyrë, një shabllon mund t'u dërgohet automatikisht një numri të madh përdoruesish, ndërsa secili marrës merr një mesazh të personalizuar me të dhënat e tij personale.



KONTROLLO NJOHURITË E TUA

Rretho përgjigjen e saktë

1. Letra qarkulluese shërben për:

- a) instalimin e programeve
- b) krijimin e dokumenteve të shumta të personalizuara
- c) formatimin e diskut
- ç) krijimin e bazës së të dhënave

2. Fushat e bashkimit:

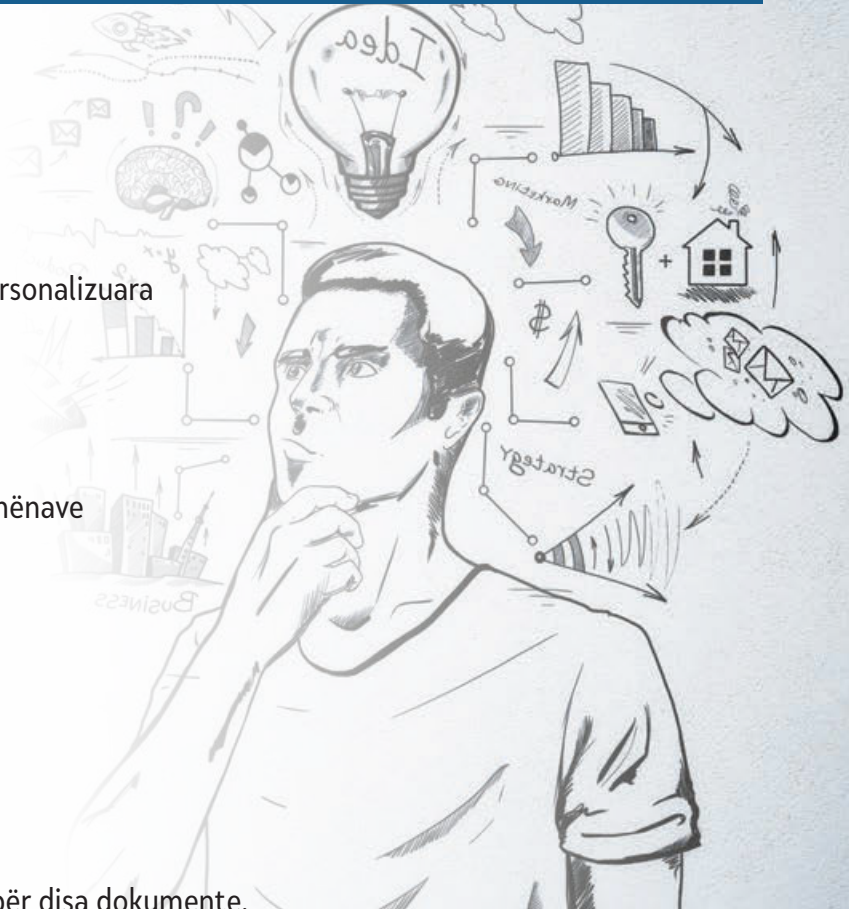
- a) i fshijnë të dhënat
- b) shërbejnë për futjen automatike të të dhënave
- c) krijojnë një tabelë të re
- ç) shtypin raporte

3. Burim i të dhënave mund të jetë:

- a) bazë të dhënash
- b) tabelë
- c) bazë të dhënash ose tabelë
- ç) sistem operativ

E saktë ose e pasaktë

- Letra qarkulluese përdor një shabllon për disa dokumente.
- Çdo dokument duhet të plotësohet manualisht.
- Të dhënat mund të merren automatikisht nga baza e të dhënave.
- Fushat e bashkimit zëvendësohen me të dhënat reale gjatë krijimit të dokumenteve.



ANALIZË

Shkolla duhet të përgatisë 300 certifikata për pjesëmarrësit në konkurs.

Shpjego:

- pse letra qarkulluese është një zgjidhje më e mirë se plotësimi manual;
- cilat të dhëna do të nevojiteshin në bazën e të dhënave;
- cilat fusha bashkimi do të përdorehin në certifikatë.

3.9



SQL – GJUHË THEMELORE PËR PUNË ME BAZA TË TË DHËNAVE

TASHMË DI ...

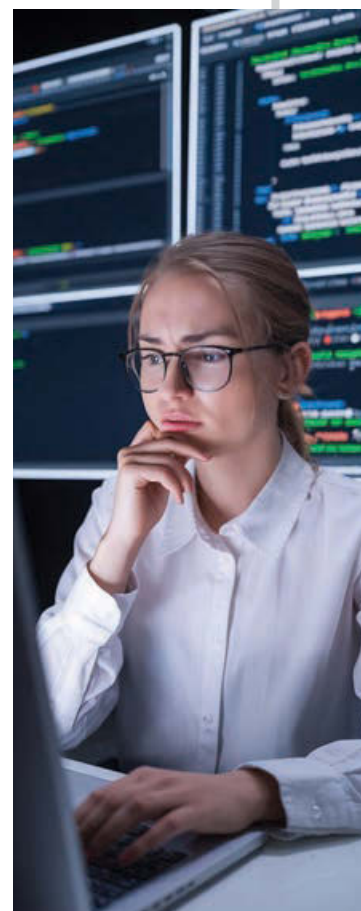
- të krijosh tabela në bazën e të dhënave;
- të përcaktosh fushat dhe llojet e të dhënave;
- të përdorësh çelësin primar;
- të futësh të dhëna përmes formularëve;
- të kërkes informacione me anë të pyetësorëve;
- të përgatitësh raporte;
- të përdorësh të dhënat nga baza e të dhënave gjatë përgatitjes së letrave qarkulluese.

PYETJE PËR TË MENDUAR

- Si komunikon kompjuteri me bazën e të dhënave?
- Si mund të kërkes informacione të caktuara nga tabela?
- Si mund të renditen rezultatet e kërkimit?
- A ekziston një gjuhë standarde që përdoret në baza të ndryshme të të dhënave?
- Si kryhen kërkimet në baza të mëdha të të dhënave?

NË PËRMBAJTJEN QË VIJON DO TË MËSOSH:

- të shpjegosh se çfarë përfaqëson SQL;
- të përdorësh komandat themelore të SQL-së;
- të kryesh kërkime të thjeshta;
- të vendosësh kushte gjatë kërkimit;
- të renditësh rezultatet;
- të analizosh rezultatet e marra me SQL.



Në mësimet e mëparshme u tregua se si mund të futen të dhënat përmes formularëve, të kërkes me anë të pyetësorëve dhe të paraqiten përmes raporteve. Në shumicën e programeve, këto veprime mund të kryhen me mjete dhe menu grafike. Megjithatë, në sfond ekziston një gjuhë standarde që mundëson komunikimin e drejtpërdrejtë me bazën e të dhënave.

Kjo gjuhë quhet SQL (Structured Query Language). Ajo përfaqëson një standard ndërkombëtar për punën me bazat relacionale të të dhënave dhe përdoret në një numër të madh sistemesh për menaxhimin e të dhënave. Falë SQL-së, përdoruesit dhe programuesit mund të krijojnë tabela, të fusin të dhëna, të kërkojnë informacione dhe të menaxhojnë përmbajtjen e bazës së të dhënave.

Edhe pse programet moderne ofrojnë mjete grafike për punën me bazat e të dhënave, shumë prej funksioneve të tyre në të vërtetë gjenerojnë komanda SQL. Prandaj, njohja e bazave të SQL-së mundëson një kuptim më të mirë të mënyrës se si funksionojnë bazat e të dhënave.

Çfarë është SQL?

SQL është një gjuhë e specializuar që përdoret për punën me bazat relacionale të të dhënave. Emri i saj rrjedh nga shprehja angleze Structured Query Language, që në përkthim do të thotë „gjuhë e strukturuar për kërkimin e të dhënave“ në bazën e të dhënave.

Me SQL mund të kërkohen informacione nga tabelat, të shtohen regjistrime të reja, të ndryshohen të dhënat ekzistuese dhe të fshihen informacionet që nuk nevojiten më. Për shkak të kësaj shumëllojshmërie, SQL përfaqëson një nga gjuhët më të rëndësishme në fushën e bazave të të dhënave.

Në këtë mësim do të trajtohen vetëm komandat themelore të SQL-së, të cilat mundësojnë kërkimin dhe përzgjedhjen e të dhënave. Veprimet kryhen përmes pyetësorit. Për secilën nga komandat e mëposhtme, fillimisht krijohet një tabelë dhe më pas një pyetësor, në mënyrë që të mund të përdoret SQL-ja.

Komanda SELECT

Komanda SQL më e përdorur është SELECT. Ajo shërben për shfaqjen e të dhënave nga një ose më shumë tabela.

Të shqyrtohet tabela Nxënës.

Emri	Mbiemri	Klasa
Ana	Petrova	II-1
Marko	Stojanov	II-2
Elena	Nikolovska	II-1

Për shfaqjen e të gjitha regjistrimeve përdoret:

```
SELECT * FROM Nxenesit;
```

Ylli (*) tregon se duhet të shfaqen të gjitha fushat e tabelës.

Nëse është e nevojshme të shfaqen vetëm emrat dhe mbiemrat, mund të shkruhet:

```
SELECT Emri, Mbiemri  
FROM Nxenesit;
```

Komanda WHERE

Në shumë situata është e nevojshme të shfaqen vetëm regjistrimet që plotësojnë një kusht të caktuar. Për këtë qëllim përdoret komanda WHERE.

Le të supozojmë se është e nevojshme të shfaqen vetëm nxënësit nga paralelja II-1.

```
SELECT *  
FROM Nxenesit  
WHERE Klasa = 'II-1';
```

Sistemi do të shfaqë vetëm regjistrimet që plotësojnë kushtin e përcaktuar.

Kushtet mundësojnë kërkim shumë më të shpejtë të të dhënave dhe përdoren pothuajse në të gjitha bazat reale të të dhënave.

Komanda ORDER BY

Ndonjëherë të dhënat duhet të shfaqen sipas një renditjeje të caktuar. Për këtë qëllim përdoret komanda ORDER BY.

Le të supozojmë se është e nevojshme që nxënësit të renditen sipas mbiemrit.

```
SELECT *
FROM Nxenesit
ORDER BY Mbiemri;
```

Rezultatet do të shfaqen sipas rendit alfabetik.

Është e mundur edhe renditja në rend zbritës:

```
SELECT *
FROM Nxenesit
ORDER BY Mbiemri DESC;
```

Fjala kyçe DESC tregon renditjen zbritëse.

Komanda DELETE

Ndonjëherë është e nevojshme që regjistrime të caktuara të hiqen nga tabela. Për këtë qëllim përdoret komanda DELETE.

Për shembull:

```
DELETE FROM Nxenesit
WHERE Klasa = 'IV-4';
```

Kjo komandë do t'i fshijë të gjitha regjistrimet që plotësojnë kushtin.

Për shkak të mundësisë së humbjes së të dhënave, DELETE duhet të përdoret me kujdes dhe gjithmonë me një kusht të përcaktuar qartë.

Shembull praktik

Të shqyrtohet një bazë të dhënash që përmban informacione për nxënësit.

Është e nevojshme të shfaqen të gjithë nxënësit me mesatare më të madhe se 4.50 dhe rezultatet të renditen sipas mbiemrit.

```
SELECT *
FROM Nxenesit
WHERE Mesatarja > 4.50
ORDER BY Mbiemri
```

Në një komandë të vetme kombinohen kërkitimi, vendosja e kushteve dhe renditja e rezultateve. Kjo përfaqëson një nga përparësitë më të mëdha të SQL-së.

Analiza e shembullit

Nga shembulli i mëparshëm mund të vërehet se SQL mundëson menaxhim të thjeshtë dhe të saktë të të dhënave. Në vend që përdoruesi t'i kërkojë manualisht regjistrimet, mjafton të vendosë komandën përkatëse SQL dhe sistemi do t'i gjejë automatikisht informacionet e nevojshme.

Falë strukturës së saj të thjeshtë dhe mundësive të mëdha, SQL sot përfaqëson standardin për punën me bazat relacionale të të dhënave dhe përdoret pothuajse në të gjitha sistemet moderne informatike.

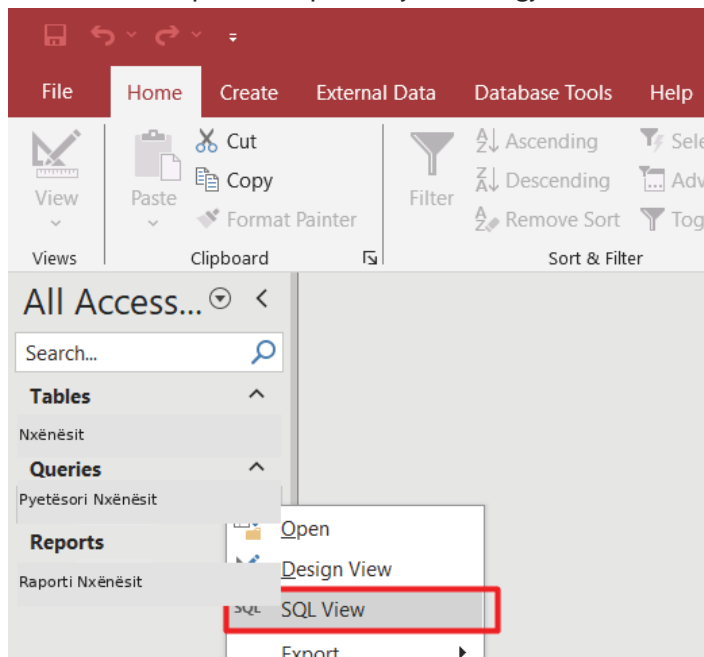


Figura 1. Pamja e SQL-së në MS Access

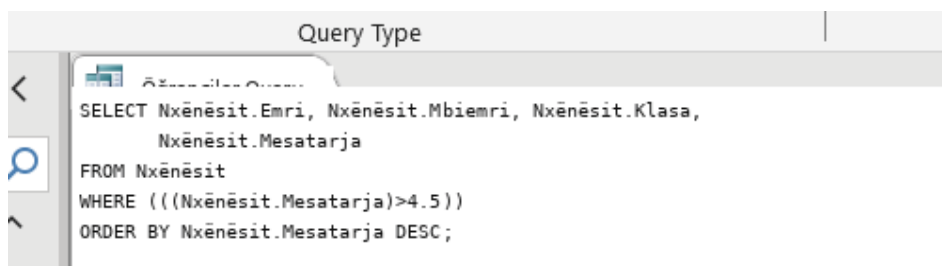


Figura 2. Komandat në SQL

SQL përfaqëson një gjuhë standarde për punën me bazat relacionale të të dhënave. Me ndihmën e saj mund të kërkohen, të renditen dhe të përpunohen të dhënat që ruhen në tabela. Edhe pse shumë programe ofrojnë mjete grafike për punën me bazat e të dhënave, në thelb të tyre zakonisht përdoren komanda SQL.

Komandat SELECT, WHERE dhe ORDER BY mundësojnë gjetjen dhe organizimin e shpejtë të informacionit, ndërsa DELETE përdoret për heqjen e regjistrimeve të panevojshme. Falë këtyre mundësive, SQL përfaqëson një mjet të rëndësishëm për punën me të dhënat në sistemet moderne informatike.

Me mësimin e SQL-së përmbyllet tema për bazat e të dhënave, sepse nxënësit fitojnë njohuri jo vetëm për organizimin e të dhënave, por edhe për mënyrën se si sistemet komunikojnë me to.



A E DI?

A e di se SQL përdoret prej më shumë se katër dekadash dhe sot përfaqëson një nga gjuhët e programimit më të përhapura në botë? Pothuajse të gjitha sistemet e mëdha informatike, aplikacionet ueb dhe shërbimet celulare përdorin SQL për të punuar me të dhënat.



PYETJE PËR PËRSËRITJE

1. Çfarë është SQL?
2. Nga cilat fjalë përbëhet shkurtesa SQL?
3. Për çfarë shërben komanda SELECT?
4. Për çfarë përdoret komanda WHERE?
5. Çfarë roli ka ORDER BY?
6. Kur përdoret komanda DELETE?
7. Pse SQL konsiderohet gjuhë standarde për punën me bazat e të dhënave?



DETYRA PRAKTIKE

Detyra 1

Shkruaj një komandë SQL me të cilën do të shfaqen të gjitha regjistrimet nga tabela Nxënësi.

Detyra 2

Shkruaj një komandë SQL me të cilën do të shfaqen vetëm fushat:

- Ime
- Prezime

prej tabelës Nxënësi.

Detyra 3

Shkruaj një komandë SQL që do të shfaqë vetëm nxënësit nga klasa II-1.

Detyra 4

Shkruaj një komandë SQL që do t'i shfaqë të gjithë nxënësit të renditur sipas mbiemrit.

Detyra 5

Shkruaj një komandë SQL që do të shfaqë nxënësit me mesatare më të madhe se 4.50.

Detyra 6

Shkruaj se çfarë do të bëjë komanda e mëposhtme SQL:

SELECT *

FROM Nxenesit

WHERE Mesatarja > 4.50

ORDER BY Mbiemri;



DETYRË HULUMTUESE

Hulumto se cilat sisteme për menaxhimin e bazave të të dhënave përdorin SQL.

Për të paktën tre sisteme, gjej:

- emri i sistemit;
- prodhuesi;
- fusha e përdorimit;
- një e dhënë interesante për sistemin.

Përgatit një raport të shkurtër ose një prezantim.



MBAJ MEND ÇFARË MËSOVE

- SQL është gjuhë standarde për punën me bazat relacionale të të dhënave.
- SELECT përdoret për shfaqjen e të dhënave.
- WHERE mundëson përcaktimin e kushteve gjatë kërkimit të të dhënave.
- ORDER BY përdoret për renditjen e rezultateve.
- DELETE përdoret për fshirjen e regjistrimeve.
- SQL mundëson punë të shpejtë dhe efikase me sasi të mëdha të të dhënave.
- Një numër i madh sistemesh moderne përdorin SQL në punën e tyre.

PËR ATA QË DËSHIROJNË TË DINË MË SHUMË



Përveç komandave bazë, SQL përmban një numër të madh mundësish të avancuara. Me të mund të lidhen disa tabela, të kryhen llogaritje statistikore, të grupohen të dhënat dhe të krijohen raporte të ndërlikuara. Në zbatimet profesionale, SQL përfaqëson një nga teknologjitë më të rëndësishme në zhvillimin e sistemeve të informacionit dhe aplikacioneve ueb.



KONTROLLO NJOHURITË E TUA

Rretho përgjigjen e saktë

1. SQL përfaqëson:

- a) sistem operativ
- b) gjuhë programimi për grafikë
- c) gjuhë për punë me bazat e të dhënave
- d) program antivirus

2. Komanda SELECT shërben për:

- a) fshirjen e të dhënave
- b) shfaqjen e të dhënave
- c) printimin e dokumenteve
- d) krijimin e formularëve

3. Komanda WHERE shërben për:

- a) renditjen e të dhënave
- b) vendosjen e kushteve
- c) fshirjen e tabelave
- d) krijimin e raporteve

4. ORDER BY përdoret për:

- a) renditjen e rezultateve
- b) fshirjen e të dhënave
- c) krijimin e fushave
- d) lidhjen e tabelave

E saktë ose e pasaktë

- 1. SQL përdoret për punë me bazat e të dhënave.
- 2. SELECT përdoret për fshirjen e të dhënave nga tabela.
- 3. WHERE mundëson përzgjedhjen e të dhënave sipas një kushti.
- 4. ORDER BY mundëson renditjen e rezultateve.
- 5. DELETE përdoret për heqjen e regjistrimeve.

PLOTËSO

- Komanda _____ shërben për shfaqjen e të dhënave.
- Komanda _____ shërben për renditjen e rezultateve.
- Komanda _____ shërben për fshirjen e regjistrimeve.

ANALIZË

Për të shqyrtuar tabelën:

Emri	Mbiemri	Mesatarja
Ana	Petrova	5.00
Marko	Stojanov	4.20
Elena	Nikolovska	4.80

Shkruaj një komandë SQL me të cilën do të shfaqen nxënësit me mesatare më të madhe se 4.50, të renditur sipas mbiemrit.

Shkruaj se çfarë do të shfaqet si rezultat.

MENDIM PËRFUNDIMTAR

Bazat e të dhënave përfaqësojnë themelin e sistemeve moderne të informacionit. Gjatë kësaj teme u tregua se si organizohen të dhënat në tabela, si futen përmes formularëve, si kërkohen me anë të pyetësorëve, si paraqiten përmes raporteve, si përdoren në letrat qarkore dhe si përpunohen me SQL. Këto njohuri përbëjnë një bazë të rëndësishme për mësimin e mëtejshëm të informatikës dhe për zhvillimin e zgjidhjeve moderne softuerike.



TEMA 4



TEMA 4: MULTIMEDIA DHE GRAFIKA KOMPJUTERIKE

Multimedia dhe grafika kompjuterike përbëjnë një pjesë të rëndësishme të teknologjisë bashkëkohore dhe të jetës së përditshme. Ato përfshijnë përdorimin e tekstit, zërit, imazheve, videos dhe animacionit, por edhe krijimin dhe përpunimin e elementeve vizuale me ndihmën e kompjuterit. Falë grafikës kompjuterike, mund të krijohet imazhe realiste, lojëra, filma dhe dizajne digjitale. Multimedia i bën informacionet më interesante dhe më të lehta për t'u kuptuar, veçanërisht në arsim dhe në fushën e argëtimit. Sot, këto teknologji përdoren në shumë fusha, si: marketingu, mjekësia, arkitektura dhe industria e lojërave. Për këtë arsye, njohja e multimedias dhe grafikës kompjuterike është e rëndësishme për të rinjtë, të cilët rriten në botën digjitale.

KONCEPTE TË REJA:

Multimedia, elemente – tekst, imazh, zë, video, grafikë, grafikë rasterike, grafikë vektoriale, piksel, rezolucion, përzgjedhje, shtresa, dukshmëri.

TASHMË DI :

- të redaktosh imazhe duke përdorur mjete bazë;
- të kopjosh dhe të zhvendosësh pjesë të një imazhi, të fshish pjesë të një imazhi, të aplikosh efekte në një imazh.
-

DETYRA PËR PËRSËRITJE

- Numëro të paktën tri mjete për redaktimin e imazheve.
- Çfarë do të thotë „kopjimi i një pjese të imazhit“?
- Cilat janë efektet më të zakonshme që mund të aplikohen në një imazh?
- Shpjego ndryshimin midis „kopjimit“ dhe „zhvendosjes“ së një pjese të imazhit.
- Shpjego se si efektet e ndryshojnë pamjen vizuale të imazhit.
- Kopjo një objekt nga një imazh dhe vendose në një imazh tjetër.
- Apliko efektin „bardhë e zi“ në një imazh të dhënë.
- Krahaso imazhin origjinal me imazhin e redaktuar – cilat janë ndryshimet?
- Analizo se si zhvendosja e objekteve e ndryshon kompozimin e imazhit.
- Shpjego pse fshirja e disa elementeve mund ta përmirësojë imazhin.

NË PËRMBAJTJET QË VIJOJNË DO TË MËSOSH:

- të shpjegosh çfarë është multimedia, elementet e saj bazë dhe zbatimin e saj, si dhe karakteristikat e grafikës rasterike dhe vektoriale;
- të zbatosh mjete dhe teknika bazë për përpunimin e imazheve, duke përfshirë korrigjimet, përdorimin e shtresave dhe kombinimin e elementeve grafike;
- të përdorësh mjete të inteligjencës artificiale për gjenerimin dhe transformimin e imazheve, të formuloni udhëzime të thjeshta (prompt) dhe të analizosh rezultatet epërfituara;
- të krijosh zgjidhje të thjeshta grafike (poster, baner ose logo), duke zbatuar teknikat e mësuara dhe mjetet e inteligjencës artificiale.

4.1



MULTIMEDIA

Në botën moderne digjitale, informacionet rrallëherë përcillen vetëm në një mënyrë. Kur lexojmë një libër elektronik, shikojmë një video edukative, ndjekim një prezantim, përdorim një aplikacion ose kërkojmë përmbajtje në internet, më së shpeshti njëkohësisht hasim tekst, imazhe, zë, video dhe elemente interaktive. Një mënyrë e tillë e paraqitjes së përmbajtjes është e afërt me mënyrën se si funksionon njeriu, sepse edhe në jetën e përditshme nuk i perceptojmë dukuritë në mënyrë të izoluar, por përmes disa shqisave dhe përvojave njëkohësisht. Kjo mënyrë e përcjelljes së informacionit përbën bazën për të kuptuar konceptin e multimedias.

Përkufizimi multimedia mund të shpjegohet si një mënyrë e paraqitjes së informacionit përmes kombinimit të disa elementeve, si: teksti, imazhi, zëri, videoja dhe animacioni. Këto elemente përdoren së bashku për të shpjeguar, paraqitur ose përcjellë një përmbajtje të caktuar në mënyrë më të plotë.

Fjala multimedia përbëhet nga dy pjesë: multi, që do të thotë „shumë“, dhe media, që i referohet mediave, përkatësisht mënyrave ose mjeteve përmes të cilave përcillet informacioni.

Në informatikë, multimedia lidhet me përmbajtjet dhe aplikacionet digjitale në të cilat kombinohen elemente të ndryshme, si: teksti, imazhi, zëri, videoja, animacioni dhe elementet interaktive. Kur të gjitha këto elemente bashkohen në një tërësi, krijohet një përmbajtje multimediale ose një aplikacion multimedial. Shembuj të përmbajtjeve multimediale janë: prezantimi, ueb-faqja dhe teksti shkollor elektronik, ndërsa shembuj të aplikacioneve multimediale janë aplikacioni edukativ dhe loja kompjuterike.

Pjesët nga të cilat përbëhet një përmbajtje multimediale quhen elemente multimediale. Elementet themelore multimediale janë: teksti, imazhi, zëri, videoja, animacioni dhe elementet interaktive. Secili prej tyre ka rolin e vet: teksti jep informacione, shpjegime ose udhëzime; imazhi paraqet në mënyrë vizuale një koncept, objekt ose dukuri; zëri përcjell të folurën, muzikën ose sinjalet zanore; videoja paraqet një lëvizje, proces ose ngjarje, ndërsa elementet interaktive i mundësojnë përdoruesit të zgjedhë, të përgjigjet ose të kontrollojë pjesë të përmbajtjes.

Është e rëndësishme të dihet se multimedia nuk nënkupton vetëm vendosjen e disa elementeve në një vend. Ato duhet të jenë të lidhura me temën dhe të kenë një qëllim të përbashkët. Për shembull, një prezantim me tekst, imazhe dhe një video të shkurtër për një temë mësimore është përmbajtje multimediale. Por, nëse teksti, imazhet dhe zëri vendosen pa ndonjë rend dhe nuk lidhen me temën, ato nuk do të ndihmojnë në kuptimin e informacionit.

Multimedia ka zbatim të gjerë në jetën e përditshme, sepse mundëson që informacionet të paraqiten në mënyrë të kuptueshme, interesante dhe praktike. Ajo përdoret në fusha të ndryshme, si për shembull:

- Në arsim, multimedia ndihmon që përmbajtjet mësimore të shpjegohen dhe të paraqiten në një mënyrë që është më e lehtë për t'u kuptuar. Për shembull, në vend që nxënësi vetëm të lexojë për

Sistemin Diellor, ai mund të shikojë një animacion për lëvizjen e planetëve, të dëgjojë një shpjegim dhe t'u përgjigjet pyetjeve interaktive. Kështu, të nxënit bëhet më i thjeshtë dhe më aktiv.

- Në mjekësi, multimedia mund të ndihmojë në shpjegimin e gjendjeve të ndryshme shëndetësore, të funksionimit të organeve ose të procedurave të caktuara mjekësore. Për shembull, mjeku mund të përdorë imazhe, paraqitje 3D, video dhe shpjegime të shkurtra, në mënyrë që pacienti të kuptojë më qartë se çfarë ndodh në trupin e tij ose si do të kryhet një procedurë e caktuar.
- Në turizëm, multimedia u ndihmon njerëzve të njihen me një vend para se ta vizitojnë. Për shembull, një ueb-faqe ose aplikacion turistik mund të përmbajë fotografi, harta, video, ture virtuale dhe informacione të shkurtra për monumentet, hotelet ose muzetë.
- Në inxhinieri, multimedia përdoret për paraqitjen më të thjeshtë të projekteve komplekse dhe sistemeve teknike. Për shembull, funksionimi i një makinerie mund të paraqitet përmes një simulimi digjital që përmban animacion të lëvizjes, shpjegime tekstuale, udhëzime zanore dhe mundësi që përdoruesi të zgjedhë se cilën pjesë të makinerisë dëshiron të shqyrtojë.
- Në sport, multimedia mund të përdoret për analizën e ngjarjeve sportive. Për shembull, analiza e një ngjarjeje sportive mund të përmbajë videoxhirime, përsëritje në lëvizje të ngadalësuar, grafikë, të dhëna statistikore dhe shpjegime për lëvizjet e lojtarëve.

Shembull i zbatimit të multimedias janë edhe lojërat kompjuterike. Një lojë kompjuterike mund të përmbajë imazhe, zë, lëvizje, udhëzime tekstuale dhe elemente interaktive, përmes të cilave lojtari kontrollon veprimin.

Multimedia është pjesë e mënyrës bashkëkohore të të nxënit, të punës, të informimit dhe të zgjidhjes së detyrave praktike. Ajo nuk lidhet vetëm me kompjuterët dhe pajisjet mobile, por edhe me mënyrën se si sot krijohen, paraqiten dhe përdoren informacione të ndryshme.



PËRFUNDIM:

Multimedia përfaqëson një pjesë të rëndësishme të botës moderne digjitale, duke mundësuar integrimin e llojeve të ndryshme të përmbajtjes. Me të, informacionet bëhen më të qarta, më interesante dhe më të lehta për t'u kuptuar. Përmes njohjes së elementeve të saj dhe zbatimit të tyre në situata reale, zhvillohen kreativiteti dhe aftësitë digjitale. Këto njohuri ndihmojnë në komunikimin dhe të nxënit më efikas në shoqërinë bashkëkohore.



PYETJE PËR PËRSËRITJEN E NJOHURIVE

1. Çfarë është multimedia?
2. Numëro të paktën katër elemente të multimedias.
3. Shpjego pse multimedia është e rëndësishme në jetën e përditshme.
4. Shpjego se si imazhet dhe zëri së bashku e përmirësojnë një prezantim.
5. Jep dy shembuj konkretë ku përdoret multimedia në arsim.
6. Përcakto cilat elemente multimediale përdoren në një video në YouTube.
7. Analizo cilat elemente multimediale janë të pranishme në një lojë online.
8. Krahaso dy shembuj të ndryshëm të multimedias (p.sh. një film dhe një ueb-faqe) sipas elementeve të tyre.
9. Vlerëso cili është elementi më i rëndësishëm i multimedias dhe shpjego pse.
10. Jep një shembull nga jeta e përditshme në të cilin multimedia të ka ndihmuar të kuptosh më mirë një informacion. Shpjego cilat elemente janë përdorur.
11. Zgjidh një fushë, për shembull mjekësinë, turizmin, sportin ose inxhinierinë, dhe propozoni se si mund të përdoret multimedia në atë fushë.





4.2

GRAFIKA VEKTORIALE DHE RASTER

Në jetën e përditshme shpesh hasim fotografi, vizatime, ikona, harta, skema, logo dhe paraqitje të tjera vizuale. Ato mund të përcjellin një mesazh, të paraqesin një objekt, të shfaqin një të dhënë ose të ndihmojnë që një udhëzim i caktuar të ndiqet më qartë.

Paraqitja e informacionit, ideve ose objekteve përmes elementeve vizuale, si imazhe, vija, forma, ngjyra dhe simbole, quhet **grafikë**.

Kur grafika krijohet, përpunohet dhe paraqitet me ndihmën e kompjuterit, bëhet fjalë për grafikë kompjuterike.

Në varësi të elementeve themelore nga të cilat përbëhet imazhi, zakonisht dallojmë dy lloje themelore të grafikës kompjuterike: grafikën rasterike dhe grafikën vektoriale.

Në grafikën vektoriale, imazhi paraqitet me vija, kurba, forma dhe objekte të tjera grafike. Këto elemente nuk ruhen si një rrjet pikselësh, por përshkruhen me rregulla matematikore. Kompjuteri i përdor këto rregulla për ta paraqitur imazhin në ekran.

Për shembull, një rreth në grafikën vektoriale nuk ruhet si një numër i madh pikash të vogla, por si një formë me pozicion, madhësi, vijë dhe ngjyrë të caktuar. Prandaj, imazhi vektorial mund të zmadhohet ose zvogëlohet pa u turbulluar ose pa u shfaqur si i përbërë nga katrorë të vegjël. Vijat dhe format mbeten të qarta dhe të mprehta edhe kur imazhi paraqitet i zmadhuar.

Grafika vektoriale përdoret më së shpeshti për logo, ikona, shenja, simbole, vizatime teknike, ilustrime dhe dizajne që duhet të përdoren në madhësi të ndryshme. Për shembull, një logo mund të përdoret në përmasa të vogla në një kartëvizitë, por edhe në përmasa shumë të mëdha – në një poster ose billboard, pa u turbulluar.

Përparësi e grafikës vektoriale është se ajo mund të zmadhohet dhe zvogëlohet pa humbur qartësinë e vijave dhe formave. Gjithashtu, elementet e veçanta të imazhit, si vijat, format dhe ngjyrat, mund të ndryshohen lehtësisht. Mangësi është se grafika vektoriale nuk është më e përshtatshme për paraqitjen e fotografive dhe imazheve reale me shumë nuanca, hije dhe detaje të imta.

Për punën me grafikë vektoriale përdoren programe si Inkscape, Adobe Illustrator, CorelDRAW, Figma dhe LibreOffice Draw.

Në grafikën rasterike, imazhi paraqitet si një mozaik me pika të vogla me ngjyra, të cilat quhen pikselë. Pikseli përfaqëson elementin më të vogël nga i cili përbëhet një imazh rasterik. Kur pikselët janë shumë të vegjël dhe të vendosur dendur, syri i njeriut nuk i vëren veçmas, por i bashkon në një imazh të plotë.

Te imazhet rasterike, një element i rëndësishëm është rezolucioni. Në përdorimin e përditshëm, ai shpesh lidhet me përmasat e imazhit në pikselë, për shembull 1920 × 1080 pikselë. Kjo do të thotë se imazhi ka 1920 pikselë në gjerësi dhe 1080 pikselë në lartësi. Gjatë printimit dhe skanimit, rezolucioni zakonisht shprehet si numri i pikave për inç, për shembull 300 dpi (dots per inch). Numri më i madh i pikselëve mundëson që imazhi të përmbajë më shumë detaje, ndërsa dendësia më e madhe e pikave gjatë printimit është e rëndësishme për një printim më të qartë.

Përparësi e grafikës rasterike është se mund të paraqesë mirë imazhe të ndërlikuara me shumë ngjyra, hije dhe kalime të buta midis ngjyrave. Prandaj, ajo është shumë e përshtatshme për fotografi dhe imazhe realiste.

Mangësi e grafikës rasterike është se gjatë zmadhimit të madh imazhi mund të humbasë cilësinë. Meqenëse përbëhet nga pikselë, gjatë zmadhimit ata mund të bëhen të dukshëm dhe imazhi mund të duket i turbulluar ose i përbërë nga katrorë të vegjël. Gjithashtu, imazhet rasterike me rezolucion të lartë mund të zënë më shumë hapësirë në memorie.

Për punën me grafikë rasterike përdoren programe për vizatim dhe përpunim të imazheve, si Microsoft Paint, Adobe Photoshop, GIMP, Krita dhe Paint.NET.

Skedarët grafikë mund të ruhen në formate të ndryshme. Nga formati mund të varet madhësia e skedarit, cilësia e imazhit, mundësia për sfond transparent dhe mënyra se si imazhi mund të përdoret ose të redaktohet.

Formatet themelore të imazheve janë:

- JPG ose JPEG është format rasterik që përdoret më së shpeshti për fotografi digjitale dhe imazhe me shumë ngjyra dhe detaje. Është i përshtatshëm për fotografi që publikohen në internet ose dërgohen përmes postës elektronike, por nuk është i përshtatshëm për imazhe me sfond transparent.
- PNG është format rasterik që e ruan imazhin pa humbje të cilësisë gjatë kompresimit. Ky format mbështet sfondin transparent, prandaj përdoret shpesh për ikona, imazhe për ueb-faqe, pamje nga ekrani, elemente grafike dhe imazhe të cilat është e rëndësishme që skajet dhe detajet të mbeten të qarta.
- SVG është format që përdoret më së shpeshti për grafikë vektoriale. Është i përshtatshëm për logo, ikona, simbole dhe ilustrime të thjeshta që duhet të paraqiten në madhësi të ndryshme pa u turbulluar ose pa u shfaqur si të përbëra nga katrorë të vegjël.

Gjatë zgjedhjes së formatit duhet të merret parasysh qëllimi i imazhit. Për fotografi zakonisht përdoret JPG, për imazhe me sfond transparent dhe elemente grafike të qarta përdoret PNG, ndërsa për logo, ikona dhe ilustrime vektoriale përdoret SVG.

Formate të tjera të skedarëve grafikë janë:

- Për grafikë rasterike: GIF, BMP, TIFF, RAW, PCX, TGA.
- Për grafikë vektoriale: AI, EPS, CDR, DXF, WMF, EMF.

Të kuptuarit e dallimit midis grafikës rasterike dhe vektoriale ndihmon në përdorimin, redaktimin dhe ruajtjen e duhur të imazheve. Kur dimë se si është ndërtuar një imazh dhe në cilin format është ruajtur, mund të zgjedhim më lehtë mënyrën e përshtatshme për përpunimin, printimin ose shpërndarjen e tij.



PËRFUNDIM:

Të kuptuarit e dallimit midis grafikës rasterike dhe vektoriale është thelbësor për përdorimin e duhur të imazheve digjitale. Njohja e koncepteve piksel dhe rezolucion ndihmon në vlerësimin e cilësisë së imazhit. Kuptimi i përparësive dhe mangësive të të dyja llojeve të grafikës mundëson zgjedhjen e qasjes më të përshtatshme, në varësi të nevojës. Njohja e formateve JPG, PNG dhe SVG kontribuon në përdorimin dhe ruajtjen më efikase të përmbajtjeve grafike.



PYETJE PËR PËRSËRITJEN E NJOHURIVE

1. Krahajo grafikën rasterike dhe vektoriale sipas mënyrës se si ndërtohet imazhi.
2. Shpjego pse fotografia digjitale zakonisht ruhet si imazh rasterik, ndërsa logoja si grafikë vektoriale.
3. Analizo se çfarë do të ndodhë nëse një imazh rasterik me rezolucion të ulët zmadhohet për t'u printuar në një poster të madh.
4. Krahajo formatet JPG dhe PNG. Në cilën situatë do të zgjedhur JPG, ndërsa në cilën PNG? Shpjego pse.
5. Krahajo imazhin rasterik dhe atë vektorial sipas mënyrës se si sillen gjatë zmadhimit (zoom-it).
6. Analizo se si ndikojnë pikselët në cilësinë e një fotografie.
7. Shpjego dallimin midis JPG dhe PNG për sa i përket cilësisë dhe madhësisë së skedarit.
8. Vlerëso formatin më të përshtatshëm për situatat e mëposhtme: fotografi për një ueb-faqe, logo për një shkollë dhe ikonë me sfond transparent. Arsyetoni zgjedhjen.
9. Imagjino se duhet të përgatisësh një poster për një ngjarje shkollore. Cilët elemente do t'i përpunoje si grafikë rasterike dhe cilët si grafikë vektoriale? Shpjego zgjedhjen.



4.3



BAZAT E PËRPUNIMIT TË IMAZHEVE

Përpunimi i imazhit paraqet procesin e ndryshimit dhe përshtatjes së një imazhi digjital, me qëllim përmirësimin e pamjes, cilësisë ose përdorshmërisë së tij. Ai përfshin veprime të ndryshme, si heqjen e pjesëve të panevojshme, ndryshimin e madhësisë, përshtatjen e ndriçimit, kontrastit dhe ngjyrave, shtimin e tekstit ose të elementeve të tjera, si dhe aplikimin e efekteve të caktuara. Nëpërmjet këtyre veprimeve, imazhi mund të përgatitet për printim, prezantim, faqe interneti ose për lloje të tjera të përmbajtjes digjitale. Prandaj, njohja e teknikave bazë të përpunimit të imazhit është e rëndësishme për përdorimin e drejtë dhe efikas të informacionit pamor.

Një nga programet që përdoren për punën me grafikën raster është GIMP (GIMP – GNU Image Manipulation Program). GIMP është një aplikacion softuerik i lirë për përpunimin e imazheve, i cili është i disponueshëm sipas Licencës së Përgjithshme Publike GNU (GNU GPL). Kjo licencë u mundëson përdoruesve ta përdorin lirisht programin, të studiojnë dhe të ndryshojnë kodin e tij burimor, si dhe të shpërndajnë versionet e modifikuara sipas kushteve të së njëjtës licencë.

GIMP ofron një mjedis pune në të cilin përdoruesi mund të hapë, të përpunojë dhe të ruajë imazhe në formate të ndryshme. Ai përmban një numër të madh mjetesh për punën me imazhe, të organizuara në menu, panele dhe paleta, të cilat mundësojnë kryerjen graduale dhe të qartë të detyrave. Një veçori veçanërisht e rëndësishme është puna me shtresa, nëpërmjet së cilës pjesë të ndryshme të imazhit mund të përpunohen në mënyrë të pavarur.

4.3.1. MJEDISI I PUNËS NË GIMP

Në modalitetin me një dritare (Single-Window Mode), mjedisi i punës i GIMP-it zakonisht është i organizuar në tri pjesë kryesore: paneli i majtë me mjetet, hapësira qendrore e punës, ku shfaqet dhe përpunohet imazhi, dhe paneli i djathtë me dialogë shtesë, si shtresat, kanalet, shtigjet dhe cilësimet e tjera. Gjerësia e paneleve mund të përshtatet duke tërhequr vijën ndarëse ndërmjet tyre.





Figura 4.1: Mjedisit e punës në GIMP

Paneli i mjeteve përmban:

1. mjetet për përpunimin e imazhit (mund të zgjidhen edhe nëpërmjet menysë Tools);
2. ikonën për zgjedhjen e ngjyrës së planit të parë (kryesore, primare) dhe ngjyrës së sfondit (dytësore), të paraqitura në formën e dy drejtkëndëshave, njëri i bardhë dhe tjetri i zi;
3. cilësitë e mjetit të zgjedhur (paneli ndihmës Tool Options).

Paneli ndihmës në anën e djathtë të mjedisit të punës është i ndarë në grupin e sipërm (i shënuar me numrin 4) dhe grupin e poshtëm (i shënuar me numrin 5) të dritareve ndihmëse. Ato ofrojnë mundësi shtesë për përpunimin e imazhit, si për shembull, punën me shtresat, kanalet, brushat etj. Dritaret ndihmëse që nuk përdoren mund të mbyllen me komandën Close Tab nga menyuja që hapet duke klikuar në shigjetën e vogël në këndin e sipërm djathtas. Ato mund të hapen përsëri duke zgjedhur Windows → Dockable Dialogs.

Pjesën qendrore të mjedisit të punës e zë paneli i imazhit. Në pjesën e sipërme të këtij paneli janë vendosur skeda me pamje në miniaturë të imazheve të ngarkuara.

Mbi imazh ndodhet vizorja horizontale, ndërsa në anën e majtë ndodhet vizorja vertikale. Poshtë imazhit dhe në anën e djathtë të tij ndodhen shiritat rrëshqitës (horizontal dhe vertikal).

Shiriti i statusit, në pjesën e poshtme të panelit të imazhit, është i ndarë në katër pjesë: koordinatat e treguesit të miut (shfaqen vetëm kur treguesi i miut ndodhet në panelin e imazhit), lista rënëse për zgjedhjen e njësisë matëse, fusha për zgjedhjen e nivelit të zmadhimit në përqindje, si dhe zona që shfaq emrin dhe madhësinë e skedarit nga i cili është ngarkuar imazhi.

Përshtatja e mjedisit të punës bëhet nëpërmjet dritares Preferences, e cila ndodhet në menyën Edit.

4.3.2. HAPJA, KRIJIMI I NJË IMAZHI TË RI, RUAJTJA DHE EKSPORTIMI I IMAZHIT

Komandat për hapjen (Open), krijimin e një imazhi të ri (New) dhe ruajtjen (Save, Save As) gjenden në menynë File. Imazhet në GIMP ruhen në formatin XCF. Në një format tjetër, imazhi mund të ruhet duke përdorur komandën Export As.

4.3.3. NDRYSHIMI I MODELIT TË NGJYRAVE TË IMAZHIT

Ndryshimi mund të bëhet duke përdorur opsionin Mode në menynë Image. GIMP ofron tri modele ngjyrash:

- RGB – modeli standard i ngjyrave, me thellësi 24-bitëshe.
- Grayscale – çdo piksel i imazhit paraqitet me një nuancë të ngjyrës gri.
- Indexed – ngjyra e çdo pikseli të imazhit përcaktohet nga indeksi (numri rendor) i ngjyrës në paletën e ngjyrave, e cila mund të përmbajë deri në 256 ngjyra.

4.3.4. NDRYSHIMI I PËRMASAVE TË IMAZHIT

Mund të bëhet nëpërmjet komandës Image -> Scale Image

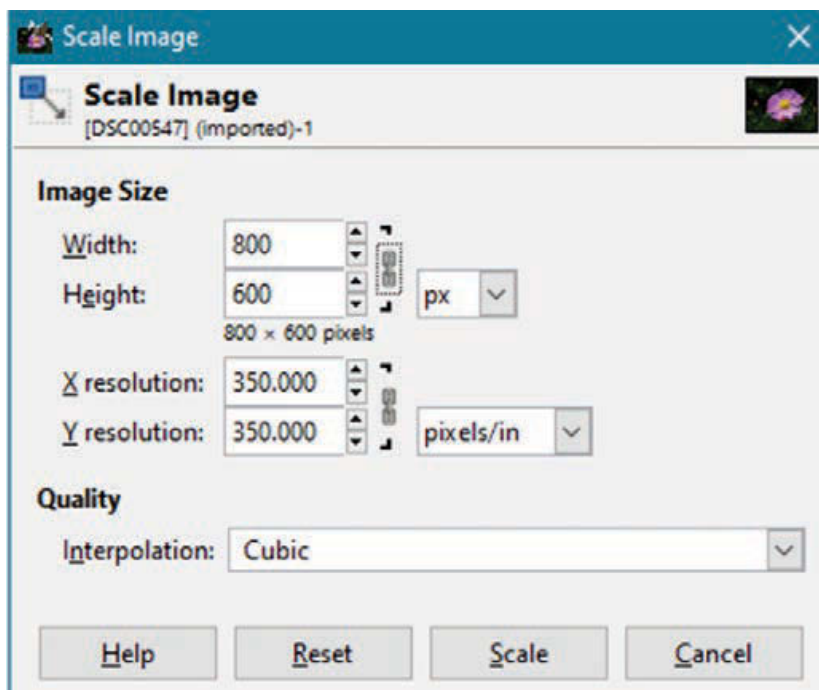


Figura 4.2 Image -> Scale Image

Gjerësia e re (Width) dhe lartësia (Height) mund të jepen në disa njësi matëse, si për shembull në piksela (px, pixels) ose në përqindje (% , percent).

4.3.5. NDRYSHIMI I PËRMASAVE TË KANAVACËS

Kanavaca (Canvas) përcakton sipërfaqen e dukshme të imazhit. Si parazgjedhje, përmasat e kanavacës përkojnë me përmasat e imazhit. Nëse kanavaca është më e madhe se imazhi, rreth tij krijohet hapësirë shtesë. Nga ana tjetër, nëse kanavaca është më e vogël se imazhi, pjesët e imazhit që ndodhen jashtë kanavacës nuk do të jenë të dukshme. Megjithatë, këto pjesë nuk humbasin, por do të shfaqen përsëri nëse rriten përmasat e kanavacës. Dritarja për ndryshimin e përmasave të kanavacës hapet duke zgjedhur Image -> Canvas Size.

4.3.6. PRERJA E PJESËS SË IMAZHIT

Prerja e pjesës së imazhit kryhet duke zgjedhur mjetin Crop (). Përcaktohet zona që dëshirohet të pritët dhe më pas shtypet Enter për ta konfirmuar. Si rezultat, krijohet një imazh që përmban vetëm pjesën e përfshirë brenda zonës së përzgjedhur për prerje. Gjithçka që ndodhet jashtë kësaj zone hiqet. Prerja mund të anulohet duke shtypur tastin Esc.

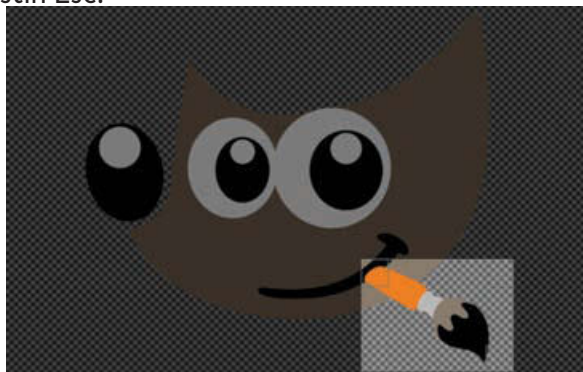


Figura 4.3: Prerja e imazhit

4.3.7. PASQYRIMI I IMAZHIT

Pasqyrimi i imazhit, horizontalisht ose vertikalisht, bëhet duke zgjedhur mjetin

Flip  nga paneli i mjeteve ose nëpërmjet menysë Image -> Transform -> Flip Horizontally / Flip Vertically.

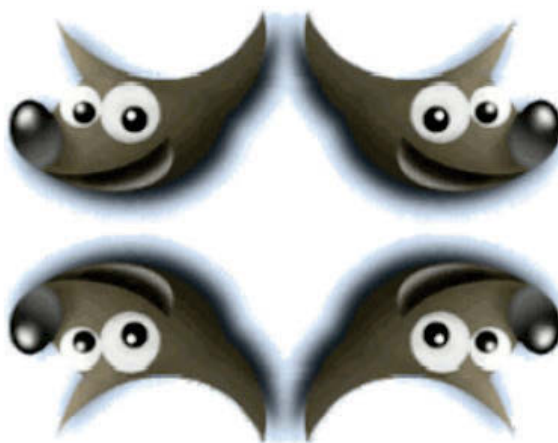


Figura 4.4: Pasqyrimi i imazhit

4.3.8. RROTULLIMI

Rrotullimi i imazhit mund të bëhet duke zgjedhur mjetin Rotate nga paneli i mjeteve

() ose nëpërmjet menysë Image -> Transform -> Rotate 90° clockwise (rrotullim 90° në drejtim të akrepave të orës) / Rotate 90° counter-clockwise (rrotullim 90° në drejtim të kundërt të akrepave të orës) dhe Rotate 180° (rrotullim 180°).

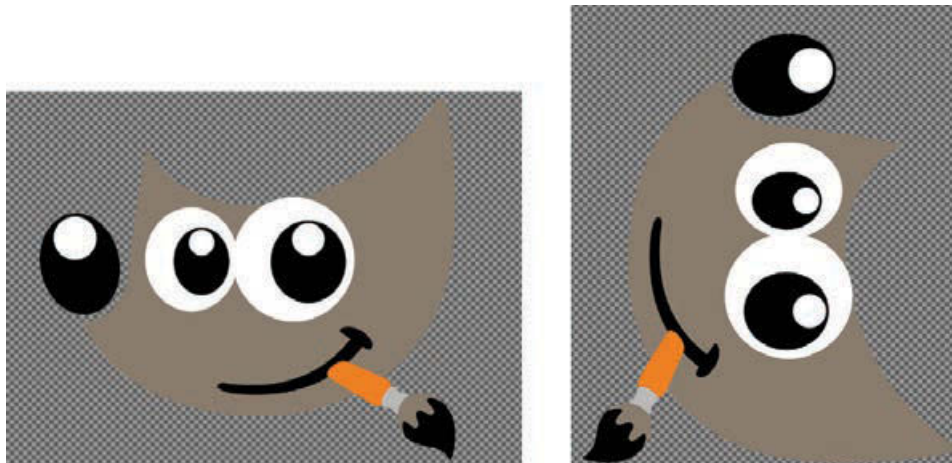


Figura 4.5: Rrotullimi i imazhit

4.3.9. VIZATIMI I VIJAVE TË DREJTA DHE TË LAKUARA

Tre mjetet e mëposhtme për vizatimin e vijave të drejta dhe të lakuara përdoren në të njëjtën mënyrë:



- lapsi për vizatimin e vijave të ashpra,



- furça për vizatimin e vijave të buta,



- spërkatësi.



Fillimisht, në dritaren me cilësitë e mjetit vendosen opsionet si ngjyra (Foreground Color), forma (Brush) dhe madhësia (Size) e brushës, e më pas shtypet butoni i majtë i miut mbi kanavacë dhe lëvizet pa e lëshuar butonin, derisa të vizatohet forma e dëshiruar. Vijat e drejta vizatohen duke përcaktuar pikat e skajshme të vijës dhe duke mbajtur të shtypur tastin Shift gjatë klikimit në kanavacë. Pika e parë vendoset pa mbajtur të shtypur tastin Shift. Çdo herë që klikohet në këtë mënyrë, programi lidh pikën e re me pikën e mëparshme me një vijë të drejtë.

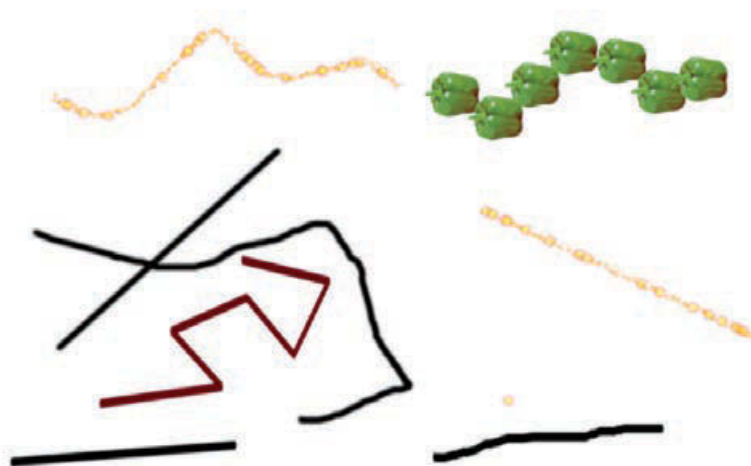


Figura 4.6: Vizatimi i vijave

4.3.10. FSHIRJA E PJESËVE TË IMAZHIT

Mjeti për fshirje  përdoret në të njëjtën mënyrë si mjetet e vizatimit dhe ka veti të ngjashme, si forma, madhësia e brushës dhe tejdukshmëria (opaciteti). Ndryshe nga mjetet e lartpërmendura të vizatimit, të cilat përdorin ngjyrën e planit të parë (Foreground Color), goma e fshirjes përdor ngjyrën e sfondit (Background Color). Nëse shtresa e imazhit ka sfond transparent, zona e imazhit e fshirë me gomë do të bëhet transparente. Nëse shtresa e imazhit nuk ka sfond transparent, zona e fshirë me gomë do të ngjyroset me ngjyrën e sfondit.

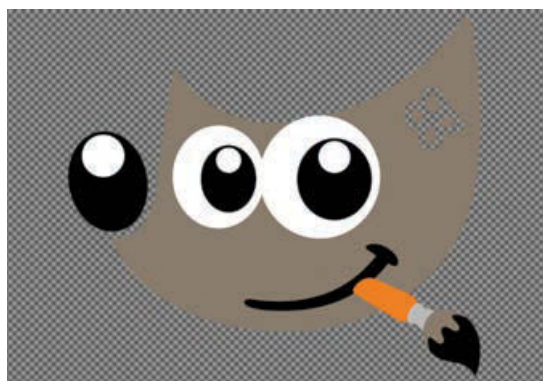
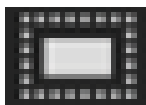


Figura 4.7: Fshirja e pjesës së imazhit me sfond transparent

4.3.11. MJETET PËR PËRZGJEDHJE

Komandat për përzgjedhje janë të grupuara në menynë Select. Komandat themelore për punën me përzgjedhjet janë: All – përzgjedhja e gjithë imazhit, None – anulimi i përzgjedhjes, Invert – përmbyjsja e përzgjedhjes dhe Float – shndërrimi i përzgjedhjes në një shtresë të përkohshme.

Ekzistojnë disa mjete për përzgjedhje. Mënyra më e thjeshtë e përzgjedhjes është përzgjedhja e drejtpërdrejtë e pjesës që dëshirohet të veçohet, duke vizatuar rreth saj. Për këtë qëllim mund të përdoren mjetet nga paneli i mjeteve, përkatësisht:



- mjeti për përzgjedhjen e një zone drejtkëndore



- mjeti për përzgjedhjen e një zone eliptike



- mjeti për përzgjedhje të lirë

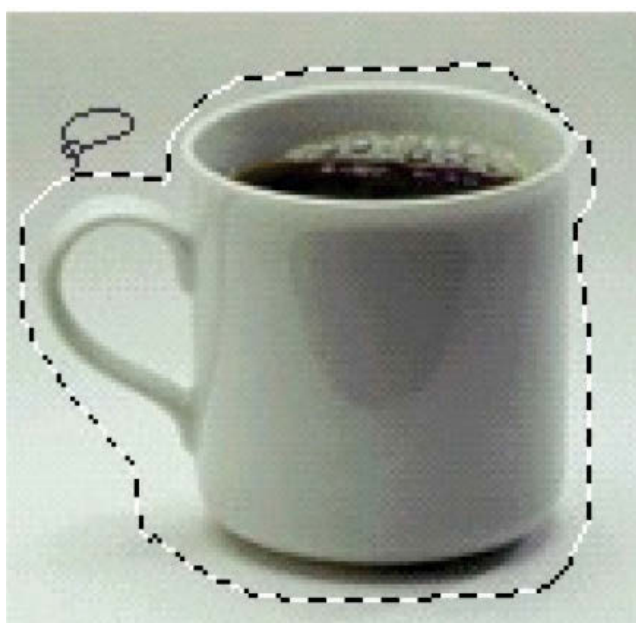
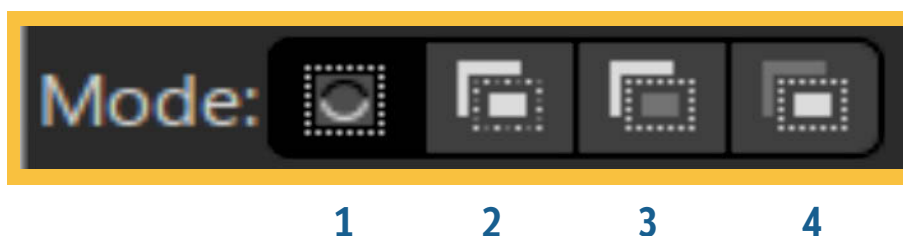


Figura 4.8: Përzgjedhja me mjetin për përzgjedhje të lirë

Përzgjedhja konfirmohet duke shtypur Enter ose duke klikuar mbi një mjet tjetër për përpunimin e imazhit. Mbajtja e shtypur e tastit Shift gjatë përzgjedhjes e shndërron drejtkëndëshin në katror, ndërsa elipsin në rreth. Për sa kohë që përzgjedhja nuk është konfirmuar, madhësia e saj mund të ndryshohet.

Me zgjedhjen e mjeteve për përzgjedhje bëhet e disponueshme edhe opsioni Mode, i cili përcakton marrëdhënien ndërmjet përzgjedhjes së re dhe përzgjedhjes ekzistuese.



1

2

3

4

Figura 4.9: Opsioni Mode

Ekzistojnë katër mundësi:

1. Replace the current selection – përzgjedhja ekzistuese anulohet dhe krijohet një përzgjedhje e re.
2. Add to the current selection – përzgjedhja ekzistuese bashkohet me përzgjedhjen e re (bashkimi i dy përzgjedhjeve).
3. Subtract from the current selection – nga përzgjedhja ekzistuese hiqen pjesët që përkojnë me përzgjedhjen e re (diferenca – nga përzgjedhja ekzistuese zbritet zona e përzgjedhjes së re).
4. Intersect with the current selection – përzgjedhja përfshin pjesët e përbashkëta të përzgjedhjes ekzistuese dhe të përzgjedhjes së re (prerja e përzgjedhjes ekzistuese me përzgjedhjen e re).

Metoda e kombinimit të përzgjedhjes së re me përzgjedhjen ekzistuese duhet të zgjidhet para se të krijohet përzgjedhja e re.



Figura 4.10: Prerja, diferenca dhe bashkimi i dy përzgjedhjeve

Në GIMP mundësohet edhe përzgjedhja sipas ngjyrës. Mjetet Fuzzy Select () dhe Select by Color () mundësojnë përzgjedhjen e një zone me ngjyrë të ngjashme me ngjyrën e pikës mbi të cilën klikohet. Dallimi ndërmjet mjeteve Fuzzy Select dhe Select by Color qëndron në faktin se i pari përzgjedh vetëm zonën rreth pikës fillestare, ndërsa i dyti përzgjedh të gjitha zonat e imazhit, jo vetëm ato rreth pikës fillestare, që kanë ngjyrë mjaftueshëm të ngjashme me ngjyrën e pikës fillestare.

Përzgjedhja mund të bëhet edhe me mjetin Gërshërë (Scissors Select) (). Mjeti Gërshërë përdoret për përzgjedhjen e pjesëve të imazhit me formë të parregullt. Përdoruesi vendos pika rreth objektit, ndërsa programi e ndjek automatikisht konturin e tij, duke u bazuar në dallimet e ngjyrës dhe kontrastit. Pas mbylljes së përzgjedhjes, pjesa e përzgjedhur mund të kopjohet, zhvendoset, fshihet ose përpunohet më tej.

4.3.12. NDRIÇIMI I IMAZHIT

Përdoret kur imazhi është shumë i errët ose kur disa detaje nuk janë mjaftueshëm të dukshme. Në GIMP ky ndryshim mund të bëhet përmes menysë Colors Brightness-Contrast. Pas zgjedhjes së këtij opsioni hapet një dritare në të cilën vlera e Brightness rritet ose zvogëlohet me ndihmën e një shiriti rrëshqitës. Me rritjen e vlerës imazhi bëhet më i ndritshëm, ndërsa me uljen e saj bëhet më i errët. Gjatë redaktimit duhet pasur kujdes që ndriçimi të mos jetë shumë i fortë, sepse mund të humbasin detajet dhe imazhi mund të duket i panatyrshëm.

4.3.13. KONTRASTI I IMAZHIT

Përdoret për të theksuar dallimin ndërmjet pjesëve të ndriçuara dhe pjesëve të errëta të imazhit. Në të njëjtën dritare, nëpërmjet shiritit rrëshqitës Contrast, vlera e kontrastit mund të rritet ose të ulet. Me rritjen e kontrastit, pjesët e ndriçuara bëhen më të ndritshme, ndërsa pjesët e errëta bëhen më të errëta, duke bërë që imazhi të duket më i theksuar. Me uljen e kontrastit, dallimet ndërmjet pjesëve të ndriçuara dhe të errëta zvogëlohen, prandaj imazhi mund të duket më i zbehtë ose më i butë. Pasi të arrihet pamja e dëshiruar, ndryshimi konfirmohet duke klikuar butonin OK.

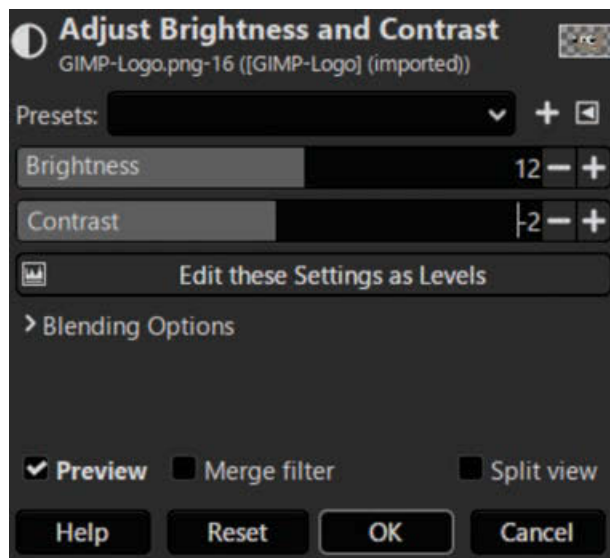


Figura 4.11: Rregullimi i ndriçimit dhe kontrastit të imazhit

Gjatë veprimeve që lidhen me përzgjedhjen, përdoret mjeti për zhvendosje.

Mund të ndryshohet edhe trashësia e vijës kufizuese të zonës së përzgjedhur duke zgjedhur Select Border, pas së cilës shfaqet dritarja për rregullimin e parametrave.

4.3.14. NGJYRA E IMAZHIT

Komanda Colors → Color Balance mundëson korrigjimin e imazheve në të cilat ngjyrat nuk janë të balancuara.

Për të përshtatur ngjyrat e errëta, te Select Range to Adjust Adjust zgjidhet Shadows. Midtones përfaqëson tonet e mesme, ndërsa Highlights tonet e çelëta. Me zgjedhjen e butonit Reset, të gjitha cilësimet kthehen në vlerat e tyre të parazgjedhura.

Ngjyrat e imazhit mund të ndryshohen edhe me komandën Colors → Hue-Saturation.

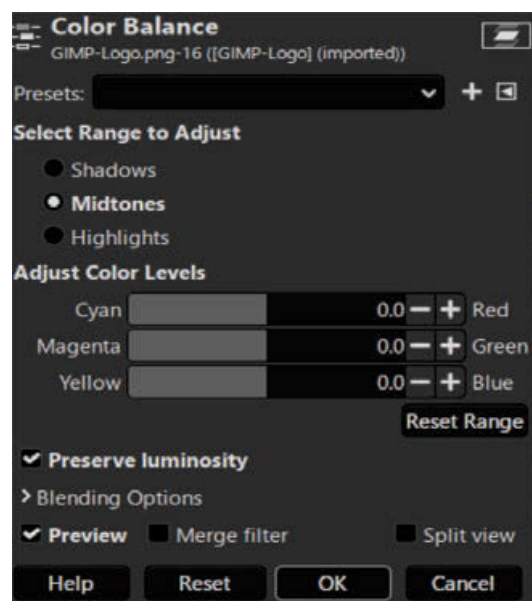


Figura 4.12: Dritarja Color Balance



Figura 4.13: Zbatimi praktik i komandës Hue-Saturation

4.3.15. KLONIMI I IMAZHIT

Përdoret për kopjimin e pjesëve të një zone të imazhit dhe transferimin e tyre në një zonë tjetër.

Në GIMP, kjo realizohet me mjetin Clone Tool (), i cili zgjidhet nga paneli i mjeteve. Fillimisht zgjidhet zona burimore e imazhit, pra zona nga e cila do të kopjohen pikselët. Kjo bëhet duke mbajtur të shtypur tastin Ctrl dhe duke klikuar mbi zonën e zgjedhur. Më pas, treguesi i miut vendoset në vendin ku dëshirohet të kopjohet përmbajtja, shtypet butoni i majtë i miut dhe, duke e tërhequr miun pa e lëshuar butonin, përmbajtja e zgjedhur kopjohet në vendin e ri. Gjatë klonimit duhet të bëhet kujdes që zona burimore të përputhet për nga ngjyra, tekstura dhe ndriçimi, në mënyrë që ndryshimi të duket sa më natyral. Ky mjet përdoret më së shpeshti për heqjen e parregullsive të vogla, mbulimin e pjesëve të padëshiruara ose përsëritjen e elementeve të caktuara të imazhit.



4.3.16. MJETI PËR PËRMIRËSIMIN E IMAZHIT

Mjeti Healing () përdoret për korrigjimin e papërsosmërive të vogla në imazh, si njollat, pikat ose gërvishtjet.

- Klikohet ikona e mjetit Healing;
- Përcaktohet zona nga e cila do të merret kampioni, duke mbajtur të shtypur tastin Ctrl dhe duke klikuar mbi atë zonë. Pas përcaktimit të burimit, lirohet tasti Ctrl. GIMP paraqet një rreth rreth pikës së zgjedhur;
- Treguesi i miut vendoset mbi zonën që duhet të korrigjohet. Nëse zona është mjaftueshëm e vogël për t'u përfshirë brenda brushës, zakonisht mjafton një klikim. Nëse zona që duhet të korrigjohet është më e madhe se madhësia e brushës, shtypet dhe mbahet i shtypur butoni i majtë i miut dhe treguesi lëviz pa e lëshuar butonin, derisa të mbulohet e gjithë zona.

Nëse është e nevojshme, për të arritur rezultatin më të mirë të mundshëm, e njëjta zonë mund të korrigjohet disa herë, derisa pjesa e përmirësuar të përshtatet sa më mirë me zonën përreth.



Figura 4.15: Shembull i përmirësimit të imazhit

4.3.17. PUNA ME TEKSTIN

Në GIMP, përveç përpunimit të imazheve, mund të shtohet dhe të redaktohet edhe tekst. Për këtë qëllim përdoret vegla Text Tool (), e cila zgjidhet nga paneli i mjeteve. Pas zgjedhjes së veglës, klikohet në vendin e imazhit ku duhet të vendoset teksti dhe më pas shkruhet përmbajtja e dëshiruar. Teksti mund të formatohet përmes opsioneve për zgjedhjen e fontit, madhësisë, ngjyrës, rreshtimit, si dhe hapësirës ndërmjet shkronjave dhe rreshtave. Gjatë shtimit të tekstit, GIMP krijon automatikisht një shtresë të re teksti, gjë që mundëson që teksti të përpunohet veçmas nga pjesët e tjera të imazhit.



Figura 4.16: Shiriti i formatimit të tekstit



PËRFUNDIM:

Mjedisi i punës së GIMP-it përbëhet nga tri panele: paneli i veglave, paneli me imazhin dhe paneli ndihmës. Imazhet e krijuara në GIMP ruhen në formatin XCF, me prapashtesën .xcf. Për ruajtjen e imazheve në një format tjetër përdoret komanda File → Export As. Komanda Image → Scale Image përdoret për ndryshimin e madhësisë së imazhit, Image → Canvas Size për përshtatjen e madhësisë së kanavacës, ndërsa Crop Tool për prerjen e një pjese të imazhit. Rrotullimi dhe pasqyrimi i imazhit kryhen përmes Image → Transform → Rotate/Flip. Për përzgjedhjen e objekteve përdoren veglat Rectangle Select, Ellipse Select dhe Free Select, të cilat mundësojnë gjithashtu fshirjen ose zhvendosjen e pjesëve të imazhit. Vegla si Eraser Tool, Clone Tool dhe Healing Tool përdoren për heqjen ose korrigjimin e pjesëve të imazhit. Përshtatja e pamjes së imazhit bëhet përmes komandave Colors → Brightness-Contrast, Color Balance dhe Hue-Saturation. Shtimi i tekstit në imazh realizohet me veglën Text Tool, duke mundësuar krijimin e zgjidhjeve grafike të rregulluara dhe kreative.





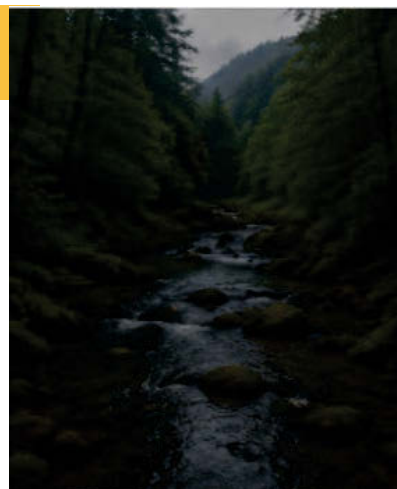
PYETJE PËR PËRSËRITJEN E NJOHURIVE

1. Krahajo ndryshimin e madhësisë së imazhit me ndryshimin e madhësisë së kanavacës. Në cilat situata do ta përdorje njërin dhe në cilat tjetër?
2. Analizo dallimin ndërmjet komandave „Save As“ dhe „Export As“ në GIMP. Pse është e rëndësishme që projekti të ruhet në formatin XCF, ndërsa imazhi përfundimtar të eksportohet në PNG ose JPG?
3. Shpjego çfarë do të ndodhë nëse imazhi shndërrohet nga modeli i ngjyrave RGB në Grayscale. Cilat informacione nga imazhi humbasin?
4. Krahajo veglat „Fuzzy Select“ dhe „Select by Color“. Në cilin rast është më mirë të përdoret njëra dhe në cilin rast tjetra?
5. Analizo dallimin ndërmjet veglës „Clone Tool“ dhe veglës „Healing“. Pse „Healing“ mund të japë një rezultat më natyral gjatë heqjes së papërsosmërive të vogla?
6. Shpjego se si mënyrat e përzgjedhjes „Add“, „Subtract“ dhe „Intersect“ ndikojnë në përzgjedhjen ekzistuese. Jep një shembull se kur do të ishte e dobishme secila mënyrë.
7. Krahajo prerjen e një pjese të imazhit me përzgjedhjen dhe fshirjen e pjesëve të imazhit. Cili është ndryshimi në rezultat?
8. Analizo se si ndryshimi i ndriçimit dhe kontrastit mund ta përmirësojë ose ta përkeqësojë cilësinë e imazhit.
9. Vlerëso se cila vegël do të ishte më e përshtatshme për heqjen e një njolle të vogël nga një fotografi: „Clone Tool“, „Healing“ apo „Eraser“. Shpjego zgjedhjen.
10. Vlerëso se kur është e arsyeshme të zvogëlohet kontrasti i imazhit. A e përmirëson gjithmonë imazhin rritja e kontrastit?
11. Vlerëso nëse vegla „Eraser“ gjithmonë fshin një pjesë të imazhit në të njëjtën mënyrë. Si ndikon fakti nëse shtresa ka ose jo sfond transparent?
12. Zgjidh veglën më të përshtatshme për përzgjedhjen e një objekti me formë të parregullt dhe shpjego pse pikërisht ajo vegël është më e përshtatshme.
13. Vlerëso rëndësinë e mjedisit të punës në GIMP. Si i ndihmojnë përdoruesit panelet, menytë, vizoret dhe shiriti i statusit gjatë përpunimit të imazhit?
14. Harto një procedurë për përpunimin e një fotografie që është e errët, pak e shtrembëruar dhe ka hapësirë të panevojshme rreth objektit kryesor. Cilat vegla dhe komanda do të përdorje dhe në çfarë rendi?
15. Kryej ushtrimet praktike të mëposhtme.



Preje imazhin në mënyrë që objekti kryesor të mbetet në fokus dhe më pas ndrysho madhësinë e tij.

Kryej korrigjimin e një imazhi të errët dhe të zbehtë.



Klono pjesë nga bari përreth dhe mbushe sipërfaqen e zbrazët, në mënyrë që imazhi të duket i rregullt dhe natyral.



Duke përdorur një vegël për përzgjedhje, ndrysho ngjyrën e luleve.



Krijoje sipas imazhit të dhënë.

4.4

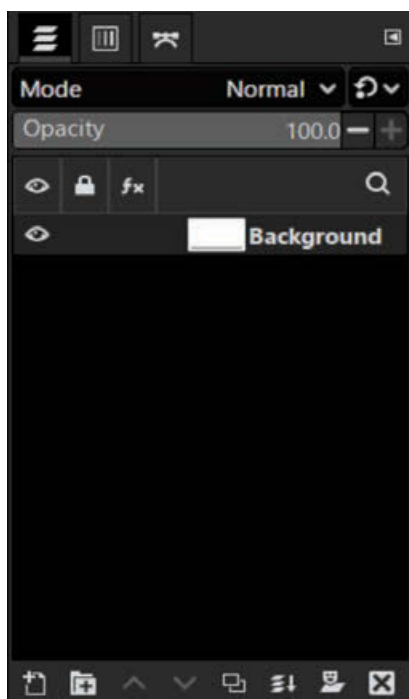


SHTRESAT (LAYERS) NË GRAFIKË

Gjatë përpunimit të një imazhi në GIMP, shpesh përdoren disa elemente, si fotografi, tekst, formë, vizatim ose një imazh tjetër. Për ta bërë punën më të thjeshtë, këto elemente mund të vendosen në shtresa të veçanta. Shtresat mundësojnë që secila pjesë e imazhit të përpunohet veçmas, pa ndryshuar drejtpërdrejt pjesët e tjera.

Shtresa mund të përfytyrohet si një sipërfaqe transparente e vendosur mbi një sipërfaqe tjetër. Nëse në një shtresë ka një fotografi, në një tjetër tekst dhe në një të tretë një formë, të gjitha së bashku krijojnë imazhin përfundimtar. Megjithatë, secila shtresë mund të zhvendoset, fshihet, bëhet e padukshme, riorganizohet ose modifikohet veçmas. Pikërisht për këtë arsye, shtresat janë të rëndësishme për një përpunim të organizuar dhe të saktë të imazhit.

Në GIMP, shtresat **shfaqen** në panelin Layers. Nëse paneli nuk është i dukshëm, ai mund të hapet përmes komandës Windows → Dockable Dialogs → Layers. Në këtë panel shfaqen të gjitha shtresat që përdoren në imazh, renditja e tyre, emrat dhe dukshmëria e tyre. Shtresa aktive është ajo që është përzgjedhur në panel dhe mbi të aplikohen veglat dhe komandat.



1. Krijimi i një shtrese të re
2. Krijimi i një grupi shtresash
3. Ndryshimi i renditjes së shtresës duke e zhvendosur një pozicion më lart
4. Ndryshimi i renditjes së shtresës duke e zhvendosur një pozicion më poshtë
5. Dublikimi i shtresës aktive
6. Bashkimi i shtresës aktive me atë poshtë saj
7. Krijimi i një maske të shtresës
8. Fshirja e shtresës

Figura 4.17: Paneli i shtresave

Shtresa e re shtohet me komandën Layer → New Layer ose duke klikuar butonin për shtresë të re në panelin Layers. Gjatë krijimit të një shtrese të re mund të përcaktohen emri, madhësia dhe lloji i mbushjes, për shembull, sfond transparent, sfond i bardhë ose një ngjyrë e caktuar për sfondin. Emërtimi i shtresave është i dobishëm, veçanërisht kur imazhi përmban shumë elemente.

Renditja e shtresave ndikon në atë se cilët elemente do të shfaqen mbi të tjerët. Shtresa që ndodhet më lart në panelin Layers shfaqet përpara shtresave që ndodhen poshtë saj. Renditja mund të ndryshohet duke e tërhequr shtresën lart ose poshtë në panel. Gjithashtu, mund të përdoren komandat Layer → Stack → Raise Layer për ta zhvendosur shtresën lart dhe Layer → Stack → Lower Layer për ta zhvendosur shtresën poshtë.

Pas përfundimit të përpunimit të imazhit, nëse është e nevojshme, shtresat mund të bashkohen. Me komandën Layer → Merge Down bashkohen shtresat fqinje. Për shembull, nëse një shtresë me tekst ndodhet mbi një shtresë me imazh, me këtë komandë teksti do të bashkohet me imazhin në një shtresë të vetme. Gjithashtu, mund të bashkohen të gjitha shtresat e dukshme në atë moment, pra ato që kanë të aktivizuar ikonën e „syrit“. Kjo realizohet me komandën Image → Merge Visible Layers. Kjo është e dobishme kur projekti përmban shumë shtresa, por ekziston nevoja që të gjitha elementet e dukshme të ndodhen në një shtresë të vetme. Me komandën Image → Flatten Image, të gjitha shtresat bashkohen në një shtresë të vetme dhe hiqet transparenca. Kjo komandë përdoret më së shpeshti para ruajtjes së imazhit në formate që nuk mbështesin shtresat, siç është JPG.

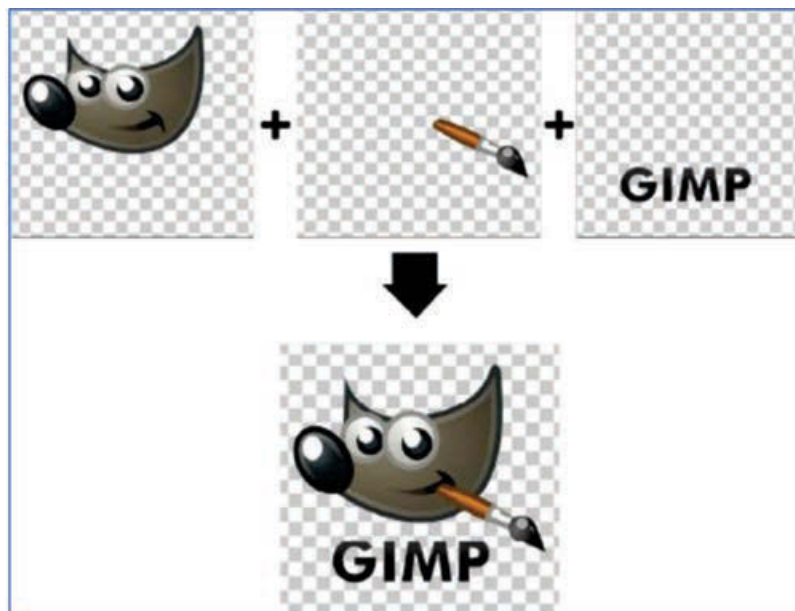


Figura 4.18. Imazh i krijuar duke përdorur shtresa

Dukshmëria e shtresës kontrollohet përmes ikonës së syrit që ndodhet pranë emrit të shtresës në panelin Layers. Kur ikona e syrit është e shfaqur, shtresa është e dukshme. Kur çaktivizohet, përmbajtja e asaj shtrese përkohësisht nuk shfaqet, por nuk fshihet. Kjo është e dobishme kur duam të krahasojmë pamje të ndryshme të imazhit ose kur një element i caktuar nuk duhet të shfaqet përkohësisht.

Fshirja e një shtrese bëhet duke e përzgjedhur shtresën në panelin Layers dhe duke përdorur komandën Layer → Delete Layer ose butonin për fshirje në panel. Nëse është e nevojshme të krijohet një kopje e një shtrese ekzistuese, përdoret komanda Layer → Duplicate Layer. Kjo është e dobishme kur duam të provojmë të bëjmë një ndryshim, duke ruajtur njëkohësisht versionin origjinal të shtresës.



Figura 4.19: Imazh i krijuar nga shtatë shtresa

Gjatë punës me shumë shtresa, çdo element mund të përpunohet veçmas. Për shembull, mund të vendoset një fotografi si sfond, mbi të të shtohet tekst me veglën Text Tool, dhe më pas të shtohet një formë ose një imazh tjetër në një shtresë të re. Meqenëse çdo element ndodhet në një shtresë të veçantë, pozicioni, madhësia, ngjyra, dukshmëria ose renditja e tij mund të ndryshohen lehtësisht.

Në GIMP mund të ndryshohet edhe **tejdukshmëria** e shtresës përmes opsionit Opacity në panelin Layers. Me zvogëlimin e vlerës së tejdukshmërisë, shtresa bëhet më transparente dhe përmes saj mund të shihen pjesërisht shtresat që ndodhen poshtë saj. Kjo mundësi përdoret për zbutjen e kufijve të mprehtë ndërmjet elementeve dhe për krijimin e kalimeve graduale të ngjyrave, për bashkimin më të njëtrajtshëm të elementeve ose për kombinimin e disa elementeve në një kompozim të vetëm.

Pas përfundimit të **përpunimit**, imazhi mund të ruhet si projekt pune në formatin XCF, përmes komandës File → Save As, duke ruajtur të gjitha shtresat. Nëse imazhi duhet të përdoret si një imazh i zakonshëm, për shembull në një dokument, prezantim ose faqe interneti, ai eksportohet përmes File → Export As në një format si PNG ose JPG. Gjatë eksportimit, shtresat bashkohen në një imazh të vetëm përfundimtar.



PËRFUNDIM:

Për përpunim më të lehtë të imazheve përdoren shtresat (Layers), të cilat shërbejnë për organizimin e elementeve të veçanta brenda një imazhi. Në secilën shtresë mund të punohet në mënyrë të pavarur, pa ndryshuar përmbajtjen e shtresave të tjera. Puna me shtresat realizohet përmes panelit Layers, ku shtresa aktive shënohet me ngjyrë blu. Pas përfundimit të përpunimit të imazhit, të gjitha shtresat mund të bashkohen në një shtresë të vetme me komandën Image → Flatten Image. Një shtresë mund të menaxhohet përmes komandave të mëposhtme: krijimi i një shtrese të re (Layer → New Layer), fshirja e një shtrese (Layer → Delete Layer), dublikimi i një shtrese (Layer → Duplicate Layer), ndryshimi i renditjes së shtresës (Layer → Stack → Raise Layer / Lower Layer) dhe ndryshimi i tejdukshmërisë së shtresës përmes opsionit Opacity në panelin Layers.

Nxënësit do të jenë në gjendje të zbatojnë dukshmërinë dhe renditjen e shtresave, duke kontrolluar se cili element shfaqet mbi ose nën një element tjetër. Përmes kombinimit të disa shtresave, ata do të krijojnë zgjidhje grafike më të ndërlikuara dhe më kreative.



PYETJE PËR PËRSËRITJEN E NJOHURIVE

1. Çfarë janë shtresat në GIMP?
2. Pse përdoren shtresat gjatë përpunimit të imazhit?
3. Ku shfaqen shtresat në GIMP?
4. Si hapet paneli Layers nëse nuk është i dukshëm?
5. Çfarë përfaqëson shtresa aktive?
6. Si shtohet një shtresë e re në GIMP?
7. Pse është e dobishme që shtresat të emërtohen?
8. Si ndikon renditja e shtresave në pamjen e imazhit?
9. Si mund të ndryshohet renditja e shtresave?
10. Për çfarë shërben ikona me sy pranë shtresës?
11. Cili është ndryshimi midis fshehjes dhe fshirjes së një shtrese?
12. Kur mund të jetë i dobishëm dublikimi i një shtrese?
13. Pse është mirë që teksti të vendoset në një shtresë të veçantë?
14. Në cilin format ruhet versioni i punës i imazhit në GIMP?
15. Cili është ndryshimi midis „Save As“ dhe „Export As“?
16. Hap një imazh të ri në GIMP. Krijë të paktën tri shtresa: sfondin, formën dhe tekstin. Mbush shtresën e sfondit me një ngjyrë, në shtresën e dytë vizato një drejtkëndësh ose një formë tjetër, ndërsa në shtresën e tretë vendos një tekst të shkurtër.
17. Duke përdorur shtresat, krijë një poster për sjellje të sigurt në internet.



4.5



INTELIGJENCA ARTIFICIALE GJENERATIVE PËR GRAFIKËN

Inteligjenca artificiale gjenerative është një teknologji që mund të krijojë përmbajtje të reja bazuar në udhëzimet e dhëna nga përdoruesi. Në fushën e grafikës, ajo përdoret për krijimin e imazheve, ilustrimeve, posterëve, banderolave, logove, sfondeve dhe elementeve të tjera vizuale. Në vend që përdoruesi ta vizatojë ose ta përpunojë tërësisht imazhin me dorë, ai mund të japë një përshkrim, dhe në bazë të këtij përshkrimi, mjeti i IA-së mund të propozojë një zgjidhje vizuale.

Një nga përdorimet më të zakonshme të inteligjencës artificiale gjenerative në grafikë është text-to-image. Ky term nënkupton krijimin e një imazhi nga një përshkrim tekstual. Përdoruesi shkruan një ose disa fjali, në të cilat shpjegon se çfarë dëshiron të paraqitet në imazh, ndërsa mjeti gjeneron një imazh sipas këtyre udhëzimeve. Për shembull, nëse jepet përshkrimi: „poster për një panair shkollor me libra, ngjyra të ndritshme dhe një titull të madh“, mjeti i IA-së mund të krijojë një imazh që përputhet me këtë përshkrim.

Udhëzimi tekstual që i jepet sistemit të IA-së quhet prompt. Prompt-i duhet të jetë i qartë, konkret dhe mjaftueshëm përshkruar. Në të mund të përfshihen objektet që duhet të paraqiten, stili i imazhit, ngjyrat, vendosja e elementeve, madhësia, atmosfera dhe qëllimi i grafikës. Sa më i saktë të jetë prompt-i, aq më e madhe është mundësia që rezultati i marrë të jetë më afër idesë së përdoruesit.

Për shembull, prompt-i „krijo një imazh me natyrë“ është tepër i përgjithshëm. Një prompt më i mirë do të ishte: „Krijo një banderolë horizontale me një peizazh malor, pyll të gjelbër, qiell blu dhe hapësirë për titull në këndin e sipërm të majtë.“ Në shembullin e dytë, mjeti merr më shumë informacion për përmbajtjen, ngjyrat, vendosjen e elementeve dhe qëllimin e imazhit.



Figura 4.20. Aplikimi i udhëzimeve të përgjithshme, burimi: Leonardo.Ai



Figura 4.21. Aplikimi i udhëzimeve specifike, burimi: Leonardo.Ai

Ekzistojnë shumë aplikacione dhe mjete online me të cilat mund të gjenerohen përmbajtje grafike duke përdorur inteligjencën artificiale. Shembuj të tillë janë Adobe Firefly, Canva Magic Media, Microsoft Designer, Midjourney, ChatGPT (me mundësi për gjenerimin e imazheve), Leonardo AI, Ideogram dhe Freepik AI. Këto mjete mundësojnë krijimin e imazheve, ilustrimeve, posterëve, banderolave dhe përmbajtjeve të tjera vizuale në bazë të një prompt-i tekstual. Disa prej tyre janë më të orientuara drejt krijimit të imazheve artistike, të tjera drejt dizajnit grafik dhe materialeve për rrjetet sociale, ndërsa disa të tjera drejt përpunimit dhe përshtatjes së imazheve ekzistuese.

Mjetet e IA-së për gjenerimin e imazheve mund të përdoren si ndihmë gjatë dizajnit. Ato mundësojnë krijimin e shpejtë të ideve fillestare, propozimeve vizuale dhe varianteve të ndryshme të të njëjtit dizajn. Përdoruesi mund të krahasojë disa rezultate, të zgjedhë më të përshtatshmin dhe më pas ta përpunojë më tej me një program për përpunimin grafik. Në këtë mënyrë, IA-ja nuk e zëvendëson plotësisht punën krijuese, por mund të ndihmojë në planifikim, frymëzim dhe realizim më të shpejtë të zgjidhjeve grafike.

Inteligjenca artificiale gjenerative mund të ndihmojë edhe në shndërrimin ose përshtatjen e grafikave nga një lloj në tjetrin. Imazhi raster është i përbërë nga pikselë dhe përdoret kryesisht për fotografi dhe imazhe komplekse me shumë detaje. Imazhi vektorial është i përbërë nga vija, forma dhe elemente të përcaktuara matematikisht, prandaj mund të zmadhohet pa humbur cilësinë. Me ndihmën e mjeteve të caktuara të IA-së ose mjeteve me mbështetje të IA-së, një imazh raster mund të thjeshtohet dhe të shndërrohet në pamje vektoriale, për shembull në logo, ikonë ose ilustrim. Nga ana tjetër, grafika vektoriale mund të eksportohet si imazh raster kur duhet të përdoret në një dokument, prezantim ose faqe interneti.

Gjatë përdorimit të mjeteve të IA-së është e rëndësishme që rezultatet të analizohen me kujdes. Imazhi i gjeneruar nuk është gjithmonë i saktë, i përshtatshëm ose cilësor. Ndonjëherë mund të ketë parregullsi në forma, tekst, përmasa, ngjyra ose në vendosjen e elementeve. Prandaj, përdoruesi duhet të kontrollojë me kujdes nëse imazhi i përgjigjet qëllimit të përcaktuar, nëse është i qartë, vizualisht i rregullt dhe nëse mund të përdoret për qëllimin e parashikuar.

Grafikat e gjeneruara mund të krahasohen me grafikët e krijuara me dorë. Grafika e krijuar me dorë mundëson kontroll më të madh mbi çdo element, sepse përdoruesi përcakton vetë format, ngjyrat, tekstin dhe vendosjen e tyre. Grafika e krijuar me IA, nga ana tjetër, mundëson marrjen më të shpejtë të ideve dhe propozimeve vizuale, por ndonjëherë kërkon kontroll dhe përpunim shtesë. Rezultatet më të mirashpesh

arrihen kur IA-ja përdoret si mjet ndihmës, ndërsa përdoruesi e përpunon dhe e formëson dizajnin përfundimtar.

Në dizajn, mjetet e IA-së mund të përdoren për krijimin e posterëve, banderolave, imazheve të kopertinës, reklamave, ilustrimeve, sfondeve dhe materialeve vizuale për rrjetet sociale. Për shembull, nxënësi mund të gjenerojë një sfond me një mjet të IA-së, më pas ta hapë atë në një program për përpunimin e imazheve, të shtojë tekst, të ndryshojë ngjyrat, të heqë pjesë të imazhit dhe të përgatisë posterin përfundimtar. Në këtë mënyrë kombinohen teknikat e mësuara të përpunimit grafik me mundësitë e inteligjencës artificiale.

Gjatë punës me mjetet e IA-së duhet të respektohen edhe rregullat e përdorimit të përgjegjshëm. Nuk duhet të gjenerohen përmbajtje që ofendojnë, mashtrojnë ose paraqesin në mënyrë të rreme persona dhe ngjarje. Gjithashtu, duhet pasur kujdes nëse imazhi lejohet të përdoret për publikim, për një projekt shkollor ose për qëllime komerciale. Përdoruesi duhet gjithmonë ta shqyrtojë në mënyrë kritike rezultatin e marrë dhe, nëse është e nevojshme, të tregojë se gjatë krijimit është përdorur një mjet i inteligjencës artificiale.

Inteligjenca artificiale gjenerative përfaqëson një mbështetje të dobishme në përpunimin grafik bashkëkohor. Ajo mundëson krijimin e shpejtë të ideve dhe zgjidhjeve vizuale, por dizajni i suksesshëm ende varet nga njohuritë, gjykimi dhe kreativiteti i përdoruesit. Prandaj, është e rëndësishme që përdoruesi të dijë të formulojë një prompt të qartë, të analizojë rezultatet e marra dhe t'i kombinojë mjetet e IA-së me teknikat manuale të përpunimit të imazheve.



PËRFUNDIM:

Përdorimi i një mjeti të inteligjencës artificiale (IA) për krijimin e imazheve digjitale është më efikas kur formulohen udhëzime (prompt-e) të thjeshta dhe të qarta, të cilat ndikojnë drejtpërdrejt në cilësinë e rezultateve të marra. Përmes analizimit të imazheve të gjeneruara zhvillohen mendimi kritik dhe aftësia për të vlerësuar saktësinë, kreativitetin dhe përshtatshmërinë e përmbajtjeve të krijuara nga IA. Krahësimi ndërmjet imazheve të krijuara me inteligjencë artificiale dhe zgjidhjeve grafike të krijuara manualisht tregon përparësitë, por edhe kufizimet e të dyja qasjeve. IA-ja mund të jetë një mjet i dobishëm që plotëson, por nuk e zëvendëson shprehjen krijuese njerëzore.





- [illegible]

PYETJE PËR PËRSËRITJEN E TEMËS

Rretho përgjigjen e saktë.

1. Çfarë përfaqëson multimedia?

- A. Përpunimi i fotografive në një program grafik
- B. Kombinimi i disa elementeve për përcjelljen e përmbajtjes
- C. Ruajtja e imazheve në formate të ndryshme
- D. Shndërrimi i tekstit në tabelë

2. Çfarë është një imazh raster?

- A. Imazh i përbërë nga vija dhe forma gjeometrike
- B. Imazh i përbërë nga pikselë
- C. Imazh që nuk mund të printohet
- D. Imazh që gjithmonë ka sfond transparent

3. Çfarë është një imazh vektorial?

- A. Imazh i përbërë nga pikselë
- B. Imazh i përbërë nga tinguj dhe video
- C. Imazh i përbërë nga vija, kurba dhe forma
- D. Imazh që mund të hapet vetëm në GIMP

4. Cili format përdoret më shpesh për fotografi?

- A. JPG B. SVG C. AI D. DXF

5. Cili format është më i përshtatshëm për një logo që duhet të përdoret në madhësi të ndryshme?

- A. JPG B. BMP C. SVG D. RAW

6. Në cilin format GIMP e ruan versionin e punës të imazhit me shtresa?

- A. JPG B. PNG C. XCF D. GIF

7. Çfarë do të thotë text-to-image?

- A. Shndërrimi i imazhit në tekst
- B. Gjenerimi i një imazhi nga një përshkrim tekstual
- C. Fshirja e tekstit nga një imazh
- D. Ruajtja e tekstit si imazh

8. Një imazh duhet të përdoret si ikonë e vogël në një faqe interneti, por edhe si shenjë e madhe në një poster. Cili lloj grafike është më i përshtatshëm?

- A. Grafikë raster, sepse përbëhet nga pikselë
- B. Grafikë vektoriale, sepse mund të zmadhohet pa humbur qartësinë
- C. Fotografi JPG, sepse gjithmonë ka madhësi të vogël
- D. Skedar GIF, sepse gjithmonë ka cilësinë më të lartë

9. Nëse një imazh raster me rezolucion të ulët zmadhohet shumë, çfarë ka më shumë gjasa të ndodhë?

- A. Do të bëhet më e mprehtë
- B. Do të shndërrohet në imazh vektorial
- C. Do të duket e turbullt ose do të shfaqen katrorë të vegjël (pikselë)
- D. Do të fitojë sfond transparent

10. Në GIMP nxënësi e redakton tekstin, por ndryshimet zbatohen te elementi i gabuar. Cila është arsyeja më e mundshme?

- A. Është zgjedhur shtresa aktive e gabuar
- B. Imazhi është shumë i vogël
- C. Fonti është shumë i madh
- D. Dokumenti është ruajtur si XCF

11. Nëse shtresa me tekst vendoset nën shtresën me sfond të errët, çfarë mund të ndodhë?

- A. Teksti mund të mos jetë i dukshëm
- B. Teksti do të bëhet automatikisht më i madh
- C. Sfondi do të fshihet
- D. Imazhi do të eksportohet si PNG

12. Cili është ndryshimi midis fshehjes së një shtrese dhe fshirjes së një shtrese?

- A. Nuk ka dallim
- B. Fshehja është e përkohshme, ndërsa fshirja e heq përgjithmonë shtresën
- C. Fshirja vetëm e zvogëlon tejdukshmërinë
- D. Fshehja e ndryshon madhësinë e imazhit

13. Nëse imazhi është shumë i errët, cila komandë është më e përshtatshme për korrigjimin fillestar?

- A. Colors → Brightness-Contrast
- B. File → Export As
- C. Layer → Delete Layer
- D. Select → None

14. Pse mjeti Healing shpesh jep një rezultat më natyral se Clone Tool gjatë heqjes së një njolle të vogël?

- A. Sepse e fshin automatikisht të gjithë imazhin
- B. Sepse e përshtat korrigjimin me përmbajtjen përreth
- C. Sepse gjithmonë e shndërron imazhin në grafikë vektoriale
- D. Sepse shton tekst në imazh

15. Kur është më mirë të përdoret Crop Tool në vend të Canvas Size?

- A. Kur duhet të hiqet një pjesë e panevojshme rreth objektit kryesor
- B. Kur duhet të shtohet hapësirë e re rreth imazhit
- C. Kur duhet të ndryshohet ngjyra e tekstit
- D. Kur duhet të krijohet një shtresë e re

16. Cili prompt është formuluar më së miri?

- A. Bëj diçka të bukur.
- B. Një imazh për shkollën.
- C. Krijo një baner horizontal për panairin e librit në shkollë, me sfond të çelët, libra, nxënës dhe hapësirë për titull.
- D. Bëj një ngjyrë.

17. Kur dëshiron të ruash projektin me të gjitha shtresat dhe të vazhdosh ta redaktosh më vonë?

- A. Kur imazhi duhet të redaktohet përsëri më vonë, duke ruajtur shtresat
- B. Kur imazhi duhet të printohet menjëherë si fotografi
- C. Kur imazhi nuk përmban asnjë element
- D. Kur duhet të fshihet teksti

18. Duhet të krijohet një poster me sfond të gjeneruar nga inteligjenca artificiale (IA), titull dhe një imazh shtesë. Cili është rendi më i përshtatshëm i punës?

- A. Eksportohet fillimisht posteri dhe më pas i shtohet teksti
- B. Gjenerohet sfondi, kontrollohet rezultati, përpunohet në një program grafik dhe më pas shtohet teksti
- C. Fshihet sfondi dhe ruhet një imazh bosh
- D. Të gjithë elementet vendosen në vetëm një shtresë

19. Nëse duhet të bëhen dy versione të të njëjtit poster, njëri me logo dhe tjetri pa logo, cila mundësi në GIMP ndihmon më së shumti?

- A. Çaktivizimi dhe aktivizimi i dukshmërisë së shtresës me logo
- B. Ndryshimi i rezolucionit të ekranit
- C. Fshirja e të gjitha shtresave
- Ç. Hapja e kalkulatorit

20. Nxënësi ka një fotografi me një njollë të vogël në qiell. Cila procedurë është më e përshtatshme?

- A. Ta shndërrojë të gjithë imazhin në SVG
- B. Të përdorë mjetin Healing ose Clone Tool për ta korrigjuar zonën
- C. Ta zvogëlojë imazhin në 10 pikselë
- D. T'i shtojë zë

21. Nëse duhet të krijohet një poster i thjeshtë në GIMP me sfond, formë, imazh dhe tekst, si është më mirë të organizohen elementet?

- A. Të gjitha të vendosen në një shtresë
- B. Çdo element kryesor të jetë në një shtresë të veçantë
- C. Teksti të fshihet pas futjes së tij
- D. Imazhi të ruhet vetëm si zë

22. Cili plan e kombinon më së miri mjetin e IA-së me përpunimin manual të imazhit?

- A. Mjeti i inteligjencës artificiale (IA) të krijojë imazhin fillestar, ndërsa përdoruesi ta analizojë, ta modifikojë dhe ta përshtatë sipas detyrës.
- B. Përdoruesi të mos kontrollojë asgjë dhe ta publikojë menjëherë imazhin.
- C. Të përdoret vetëm përpunimi manual, pa asnjë analizë.
- D. Të përdoret vetëm një format imazhi, pavarësisht nga qëllimi i përdorimit.



TEMA 5



TEMA 5: ZHVILLIMI I WEB-IT

Bazat e zhvillimit të web-it janë një mënyrë krijuese për t'u njohur me mënyrën se si funksionojnë faqet e internetit që i përdorim çdo ditë. Krijimi i faqeve të internetit kombinon kreativitetin dhe të menduarit logjik. Çdo faqe interneti ndërtohet duke përdorur tri teknologji themelore: HTML, e cila përcakton strukturën bazë; CSS, e cila rregullon pamjen dhe ngjyrat; dhe JavaScript, i cili shton ndërveprimin dhe dinamikën e faqes.

Kur t'i mësonsh këto baza, do të jesh në gjendje të krijosh faqet e tua të internetit dhe t'i shndërrosh idetë e tua në projekte të vërteta digjitale. Gjithashtu, është e rëndësishme të kuptosh se si ta bësh një faqe të duket mirë në telefon, tablet ose kompjuter, si dhe të jetë e lehtë për t'u përdorur.

Kjo është hapi i parë për këdo që dëshiron të fillojë me zhvillimin e web-it dhe ndoshta, në të ardhmen, të merret me këtë profesion.

KONCEPTE TË REJA

- struktura e faqes së internetit, dokumenti HTML, elementet bazë të HTML-së – titulli (heading), paragrafi (paragraph), imazhi (image), lidhja (link), CSS, selektorët, ngjyrat, fontet, sfondi, renditja bazë e elementeve të faqes së internetit (layout), bazat e dizajnit adaptiv (responsive design), përshtatja për pajisje të ndryshme, udhëzimet për inteligjencën artificiale (prompt).

Tashmë e di:

- Çfarë mundësish ofrojnë shërbimet e ndryshme të internetit.
- Cili është roli i shërbimit WWW në internet.
- Të japësh shembuj për Web 1.0, Web 2.0, Web 3.0, Web 4.0 dhe Web 5.0.

NË PËRMBAJTJET QË VIJOJNË, DO TË MËSOSH:

- të shpjegosh strukturën bazë të një faqeje interneti dhe të zbatosh HTML dhe CSS për krijimin dhe rregullimin e përmbajtjeve web;
- të zbatosh parimet bazë të renditjes dhe të dizajnit adaptiv (responsive) gjatë përshtatjes së faqeve të internetit për pajisje të ndryshme;
- të përdorësh mjete me inteligjencë artificiale (IA) për gjenerimin e përmbajtjeve web;
- të formulosh udhëzime (prompt), të analizosh rezultatet e marra dhe të redaktosh kodin e gjeneruar nga IA;
- të krijosh një projekt të thjeshtë web duke kombinuar kodin e krijuar manualisht me kodin e gjeneruar automatikisht.

DETYRA PËR PËRSËRITJE:

- Numëro të paktën tre shërbime të ndryshme të internetit dhe përshkruaj se për çfarë përdoren.
- Si ndikon shërbimi WWW në mënyrën e qasjes në informacion, në krahasim me shërbimet e tjera të internetit (për shembull, e-posta ose FTP).
- Shpjego, me fjalët e tua, cilat janë dallimet kryesore midis Web 1.0 dhe Web 2.0.
- Jep një shembull konkret për Web 3.0 dhe shpjego se si ky shërbim i përmirëson mundësitë e internetit.
- Krahaso Web 2.0 dhe Web 4.0 për sa i përket ndërveprimit me përdoruesit dhe shpjego cilat janë karakteristikat e tyre kryesore.
- Mendo një shembull tëndin për një aplikacion Web 5.0 dhe shpjego se si do t'u ndihmonte përdoruesve në jetën e përditshme.

5.1



KONCEPTI I FAQES WEB

Faqja e internetit është një dokument digjital që shfaqet në një shfletues web (browser), siç janë Google Chrome, Mozilla Firefox ose Microsoft Edge. Ajo është pjesë e një sistemi më të gjerë, të quajtur web-sajt (website), i cili përbëhet nga disa faqe interneti të ndërlidhura. Çdo web-sajt përbëhet nga një ose më shumë faqe interneti, të lidhura me njëra-tjetrën përmes hiperlidhjeve (hyperlinks). Faqet e internetit vendosen në serverë web, ku është i instaluar softueri për shpërndarjen e tyre.

Çdo faqe interneti ka adresën e saj unike, të njohur si URL (Uniform Resource Locator), përmes së cilës përdoruesit mund të kenë qasje në të përmes internetit. Faqet e internetit zakonisht krijohen duke përdorur gjuhën HTML (HyperText Markup Language), e cila nuk është gjuhë programuese, por gjuhë shënjimimi që përdoret për përcaktimin e strukturës dhe përmbajtjes së faqes. HTML-i i jep shfletuesit të dhëna për përmbajtjen dhe strukturën e faqes së internetit dhe përdoret për përshkrimin e elementeve dhe mënyrës se si ato do të shfaqen.

Përveç HTML-së, shpesh përdoren edhe teknologji të tjera, si:

- CSS (Cascading Style Sheets) – për stilizimin dhe dizajnin e faqes;
- JavaScript – për ndërveprim dhe dinamikë.
- Faqet e internetit mund të jenë:
- Statike – përmbajtja është e pandryshueshme dhe nuk ndryshon shpesh;
- Dinamike – përmbajtja ndryshon në varësi të përdoruesit ose të të dhënave (për shembull, rrjetet sociale ose dyqanet online).

5.1.1. STRUKTURA BAZË E NJË WEB FAQEJE INTERNETI

Çdo faqe interneti ka një strukturë logjike dhe teknike që mundëson shfaqjen dhe funksionimin e duhur të saj. Struktura bazë teknike përcaktohet me HTML dhe përbëhet nga disa elemente kryesore. Ajo ndërtohet në këtë mënyrë:

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Faqja kryesore</title>
</head>
<body>
<h1>Titulli kryesor</h1>
<p>Ky është një shembull teksti.</p>
</body>
</html>
```



Ky kod paraqet skeletin bazë të një faqeje interneti. Secili prej elementeve ka funksionin e vet:

1. `<!DOCTYPE html>`

Ky është deklarimi që i tregon shfletuesit se dokumenti është shkruar duke përdorur versionin HTML5 të gjuhës HTML. Ai është i rëndësishëm për shfaqjen e saktë të faqes së internetit.

2. `<html>`

Kjo etiketë shënon fillimin e dokumentit HTML. Gjithçka që ndodhet midis etiketave hapëse dhe mbyllëse `<html>` është pjesë e dokumentit të faqes së internetit.

3. `<head>`

Në këtë pjesë vendosen metadatat e faqes, si:

- titulli (`<title>`),
- informacioni për kodimin e karaktereve,
- lidhjet me skedarët CSS.

Kjo pjesë nuk shfaqet drejtpërdrejt në faqen që sheh përdoruesi.

4. `<meta charset="UTF-8">`

Kjo etiketë përcakton mënyrën e kodimit të karaktereve në dokument. UTF-8 mundëson shfaqjen e saktë të shkronjave dhe simboleve të ndryshme, përfshirë edhe ato të alfabetit cirilik. Kjo është e rëndësishme që teksti të shfaqet saktë në shfletues.

4. `<title>`

Përcakton titullin e faqes së internetit, i cili shfaqet në skedën (tab) e shfletuesit.

5. `<body>`

Ky është pjesa kryesore e faqes së internetit. Në të vendoset gjithçka që përdoruesi mund të shohë:

- tekst,
- imazhe,
- video;
- lidhje.

Përveç strukturës teknike, një faqe interneti ka edhe strukturë logjike, e cila mundëson organizimin dhe paraqitjen më të mirë të përmbajtjes. Kjo strukturë zakonisht përfshin:

1. Kreu (header) – pjesa e sipërme e faqes, e cila zakonisht përmban logon, titullin dhe menynë e navigimit;
2. Navigimi (navigation) – menyja që u mundëson përdoruesve të lëvizin nëpër pjesë të ndryshme të faqes. Ajo përbëhet nga lidhje që çojnë në seksione ose faqe të ndryshme;
3. Përmbajtja kryesore (main content) – pjesa qendrore e faqes ku paraqitet informacioni kryesor. Ajo mund të përmbajë tekst, imazhe, video dhe elemente të tjera;
4. Shiriti anësor (sidebar) – pjesë shtesë e faqes, e cila zakonisht ndodhet në njërën anë dhe mund të përmbajë reklama, informacione shtesë ose lidhje.
5. Fundfaqja (footer) – pjesa e poshtme e faqes, e cila zakonisht përmban të dhëna për kontakt, të drejtat e autorit dhe lidhje shtesë.

Struktura e mirë e një faqeje interneti është e rëndësishme për disa arsye:

- Lexueshmëri më e mirë – përdoruesit e kuptojnë më lehtë përmbajtjen,
- Përvojë më e mirë e përdoruesit – mundëson navigim dhe organizim më të thjeshtë,
- Optimizim për motorët e kërkimit (Search Engine Optimization – SEO) – motorët e kërkimit mund ta analizojnë dhe ta kuptojnë më lehtë përmbajtjen e faqes,
- Mirëmbajtje më e lehtë – programuesit mund ta mirëmbajnë, përditësojnë dhe përmirësojnë faqen e internetit në mënyrë më të thjeshtë dhe më efikase.

Renditja e saktë e etiketave HTML brenda njëra-tjetrës është një nga rregullat themelore në zhvillimin e web-it. E thënë thjesht, etiketat funksionojnë sipas parimit “kuti brenda kutisë” – etiketa që hapet e fundit duhet të mbyllet e para. Ky parim njihet si LIFO (Last In, First Out – e fundit brenda, e para jashtë). Shfletuesit e internetit (si Chrome, Edge dhe Firefox) nuk e shfaqin kodin HTML drejtpërdrejt. Fillimisht, ata e shndërrojnë atë në një strukturë të quajtur DOM (Document Object Model), e cila paraqitet si një strukturë pemë, me elemente prindërore dhe bijësore të organizuara në mënyrë hierarkike.

- Në shembullin e saktë: `<html>` është elementi prind, ndërsa `<body>` është elementi i tij bijë. Elementi `<body>` është i vendosur plotësisht brenda elementit `<html>`.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Strukturë e saktë</title>
  </head>
  <body>
    <h1>Përshëndetje!</h1>
    <p>Kjo është <strong>e saktë</strong> tekst i vendosur brenda.</p>
  </body>
</html>
```



Në shembullin e pasaktë (`<html><body>...</html></body>`): Po përpiqemi të mbyllim kutinë e jashtme (`<html>`), ndërkohë që kutia e brendshme (`<body>`) ende nuk është mbyllur.

```
<!DOCTYPE html>
<html>
  <body>
  <head>
    <title>Strukturë e pasaktë</title> </head>
    <h1>Gabim!</h1>
    <p>Kjo është <strong>e pasaktë</p> tekst i vendosur brenda.</strong> </body>
</html>
```

Nëse kodi është i vendosur gabimisht brenda njëri-tjetrit (i pavendosur saktë), faqja e internetit mund të shfaqet krejt ndryshe në shfletues ose pajisje të ndryshme. Gjithashtu, elementet mund të mbivendosen, të “dalin” nga pozicionet e tyre ose teksti mund të bëhet i palexueshëm.



MBAJ MEND ÇFARË MËSOVE

Faqja e internetit përfaqëson një nga elementet themelore të internetit dhe luan një rol të rëndësishëm në komunikimin dhe informimin bashkëkohor. Kuptimi i konceptit të faqes së internetit dhe i strukturës së saj është bazë për mësimin e mëtejshëm në fushën e teknologjive web. Struktura bazë teknike e një faqeje interneti përbëhet nga elemente HTML, si <html>, <head> dhe <body>, ndërsa struktura logjike përfshin pjesë si kreu, navigimi, përmbajtja kryesore dhe fundfaqja. Këto elemente, së bashku, mundësojnë që faqja e internetit të jetë funksionale, e qartë dhe e dobishme për përdoruesit.



PYETJE PËR VERIFIKIMIN E NJOHURIVE

1. Cila është etiketa fillestare që i tregon shfletuesit se dokumenti është një faqe interneti e shkruar në HTML5?
2. Rendit tri etiketat kryesore nga të cilat duhet të përbëhet çdo dokument i rregullt HTML.
3. Ku vendosen metadatat (si titulli i faqes ose kodimi i karaktereve) – në seksionin <head> apo në seksionin <body>?
4. Shpjego dallimin midis përmbajtjes që ndodhet në etiketën <head> dhe përmbajtjes që ndodhet në etiketën <body>. Cila prej tyre është e dukshme për vizitorin e faqes së internetit?
5. Pse është e rëndësishme që etiketat HTML të vendosen saktë brenda njëra-tjetrës (për shembull, pse <html><body>...</body></html> është e saktë, ndërsa <html><body>...</html></body> nuk është)?
6. Me fjalët e tua, përshkruaj rolin e etiketës <title>. Ku shfaqet përmbajtja e saj në shfletuesin e internetit?

7. Shkruaj kodin e plotë për një dokument HTML bosh që përmban të gjitha elementet e detyrueshme strukturore, ndërsa titulli i faqes në skedën e shfletuesit duhet të jetë „Faqja Ime e Parë“.
8. Në kodin e mëposhtëm ka një gabim strukturor. Cili do të jetë mesazhi që do të shfaqet? Gjeje gabimin dhe shkruaje kodin e saktë.

```
<!DOCTYPE html>
<html>
  <body>
    <head>
      <title>Mirë se vini</title>
    </head>
    <h1>Përshëndetje botë!</h1>
  </body>
</html>
```



PËR ATA QË DËSHIROJNË TË DINË MË

Hulumto mjetin W3C Markup Validation Service. Studio funksionalitetin e tij.



Markup Validation Service

Check the markup (HTML, XHTML, ...) of Web documents

Validate by URI

Validate by File Upload

Validate by Direct Input

Validate by URI

Validate a document online:

Address:

5.2



BAZAT E HTML-SË

5.2.1. ETIKETAT BAZË

Për ta kuptuar më mirë se si krijohet një faqe interneti, mendoje kodin HTML si planin arkitektonik të një shtëpie. Vetë kodi përbëhet nga etiketa që i tregojnë shfletuesit se ku të vendosë dritaren, ku derën dhe ku çatinë.

Në vazhdim do të shqyrtojmë etiketat bazë të HTML-së dhe do t'i shpjegojmë përmes shembujve praktikë, me anë të të cilëve do të mësohesh se si t'i përdorësh.

1. Etiketat për tituj (<h1> deri <h6>)

Titujt përdoren për organizimin e përmbajtjes dhe përcaktimin e rëndësisë së saj. <h1> është titulli më i rëndësishëm dhe zakonisht më i madh (si titulli i një libri), ndërsa <h6> është titulli i nivelit më të ulët (për nëntituj të nënseksioneve).

- **Shembull praktik:** Mendo sikur po krijon një faqe interneti me receta. Titulli i recetës do të vendoset në <h1>, ndërsa nëntitujt, si “Përbërësit” ose “Mënyra e përgatitjes”, do të vendosen në <h2>.

<h1>Recetë për palaçinka</h1>

<h2>Përbërësit e nevojshëm</h2>

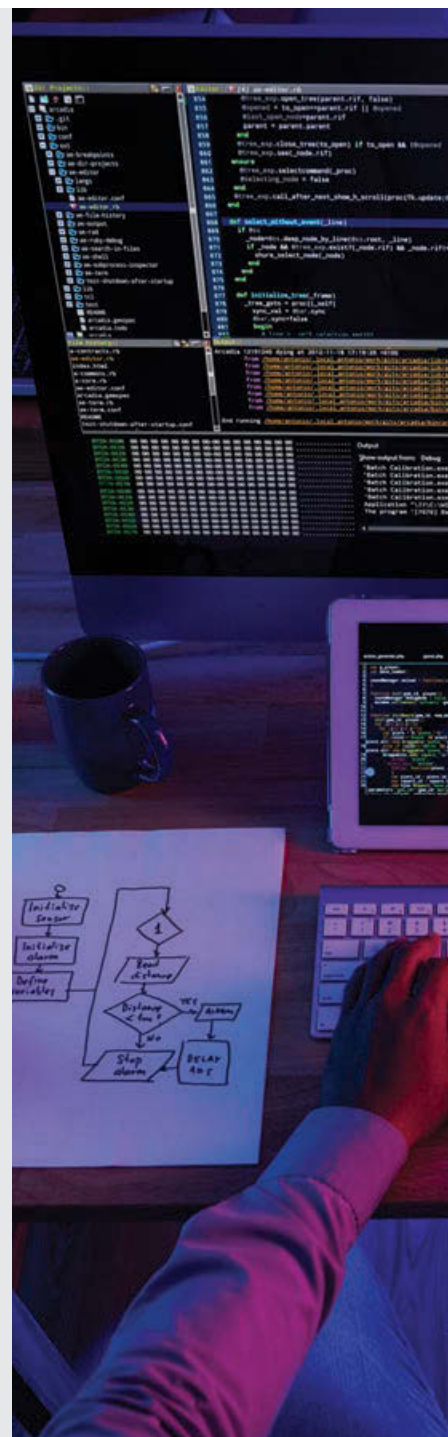
2. Etiketat për tekst dhe tituj

- **paragrafi (<p>)** – Përdoret për të shkruar tekst të zakonshëm ose fjali që do të shfaqen si paragraf. Sa herë që hap një etiketë të re <p>, shfletuesi vendos automatikisht pak hapësirë para dhe pas paragrafit, në mënyrë që teksti të jetë më i lexueshëm.

Për shembull, teksti i mëposhtëm do të shfaqet si një paragraf i veçantë.

<p>Palaçinkat janë një ëmbëlsirë e preferuar për shumë breza. Përgatiten shpejt dhe lehtë.</p>

- **Ndërprerja e rreshtit (
)** – përdoret për ta kaluar me detyrim tekstin në një rresht të ri. Kjo është një nga etiketat e rralla që **nuk ka** etiketë mbyllëse (shkruhet e vetme).
- Etiketa **...** – e shfaq tekstin me shkronja të **theksuara (Bold)** dhe përdoret për të vënë në pah fjalë ose pjesë të rëndësishme të tekstit.
- Etiketa **...** – e shfaq tekstin me shkronja të pjerrëta (*Italic*).



3. Etiketa për lidhje (<a>)

Etiketa <a> (anchor) përdoret për të lidhur një faqe interneti me një faqe tjetër në internet ose me një dokument tjetër. Për t'i treguar shfletuesit se ku duhet të çojë lidhja, përdoret atributi href. Për shembull, nëse dëshiron t'i drejtosh vizitorët te Google për të kërkuar më shumë ide për receta palaçinkash, shkruan kodin e mëposhtëm:

```
<a href="https://www.google.com">Kërkoni ide në Google</a>
```

4. Etiketa për imazhe ()

Faqet e internetit do të ishin jo interesante pa elemente vizuale. Etiketa përdoret për të vendosur një imazh në faqe. Është e rëndësishme të dish se kjo etiketë nuk ka etiketë mbyllëse (mbyllet vetë) dhe përdor atributet src (shtegu drejt imazhit) dhe alt (teksti alternativ që shfaqet nëse imazhi nuk ngarkohet). Për të shfaqur një imazh të shijshëm të palaçinkave të përgatitura, përdoret kodi i mëposhtëm:

```

```

Kur t'i zbatosh të gjitha këto etiketa në një tërësi logjike brenda trupit kryesor (<body>), fiton kodin e mëposhtëm HTML, të thjeshtë, por të plotë:

```
<!DOCTYPE html>
<html>
<head>
  <title>Receta ime e parë</title>
</head>
<body>
  <h1>Recetë për krepa të shpejta</h1>
  <p>Kjo është një nga recetat më të shijshme dhe më të thjeshta për krep, të cilat mund t'i përgatitni për vetëm 15 minuta.</p>
  
  <p>Nëse ju pëlqeu kjo recetë, vizitoni faqen tonë partnere për më shumë ide:</p>
  <a href="https://www.google.com">Më shumë receta këtu</a>
</body>
</html>
```

5. Etiketat për krijimin e listave me pika ose me numra:

- – listë e parenditur me pika,
- – listë e numëruar me numra 1, 2, 3...,
- – element i listës. Gjithmonë vendoset brenda ose .

Mund të përdoren për krijimin e një liste me përbërësit e preferuar, të cilët do të shfaqen në faqen tënde të internetit.

```
<h1>Mirë se vini në shkollën time</h1>
<p>Këto janë lëndët e mia <strong>të preferuara</strong>:</p>
  <ul>
    <li>Matematikë</li>
    <li>Dizajn web</li>
  </ul>
```

6. Etiketa për përcaktimin e një blloku – <div> (shkurtim i fjalës division – ndarje, sektor).

Ajo është një element bllok joseksantik në HTML dhe shërben si një kontejner universal (kuti) për grupimin e elementeve të tjera, me qëllim organizimin, strukturimin dhe stilizimin e tyre më të lehtë me ndihmën e CSS-së, ose ndryshimin e tyre përmes JavaScript-it. Për shembull, mund të përdoret për të krijuar një kartë të profilit të përdoruesit:

```
<div class="karta-profli">
  
  <h2>Marko Markovski</h2>
  <p>Profesioni: Dizajner web</p>
</div>
```



A E DI?

A e di se si mund të vendosen hapësira boshe në tekst? Shfletuesi i njeh hapësirat e shumta boshe vetëm si një hapësirë. Ai i bashkon të gjitha hapësirat e njëpasnjëshme në një të vetme. Nëse dëshiron të krijosh teknikisht disa hapësira njëra pas tjetrës, duhet të përdorësh një entitet të veçantë të karakterit, (non-breaking space – hapësirë që nuk ndahet).



HULUMTO

Hulumto në cilat raste paraqitet nevoja për përcaktimin e hapësirave boshe dhe si përdoret entiteti për krijimin e hapësirave të shumta.



PYETJE PËR KONTROLLIMIN E NJOHURIVE:

1. Shpjego dallimin midis strukturës logjike të një faqeje interneti (asaj që sheh përdoruesi) dhe strukturës së saj teknike (asaj që shfletuesi e sheh në kod).
2. Cili është roli i etiketës <head>, ndërsa cili është roli i etiketës <body> brenda një dokumenti HTML? Çfarë ndodh nëse një tekst që duhet të shfaqet vendoset në <head> në vend të <body>?
3. Ke për detyrë të krijosh një meny online për një restorant. Shkruaj kodin HTML që do të krijojë një listë të numëruar () me tri ushqime sipas zgjedhjes sate.
4. Shiko pjesën e mëposhtme të kodit:

```
<p>Mirë se vini në <strong>blogun tim për programim.</p></strong>
```

Gjej gabimin në këtë kod, shpjego çfarë është e pasaktë sipas rregullave të vendosjes së etiketave brenda njëra-tjetrës (nesting) dhe shkruaj formën e saktë sintaksore.

5. Analizo se si do ta shfaqë shfletuesi tekstin e shkruar në kodin:

```
<h1>Mësojmë HTML së bashku!</h1>
```

Cila karakteristikë e etiketave HTML për tekstin vihet në pah në këtë shembull?

6. Po shqyrton një dokument të thjeshtë HTML, në të cilin programuesi e ka vendosur përmbajtjen në mënyrën e mëposhtme:

Në fillim, titullin kryesor dhe më të rëndësishëm të artikullit e ka shënuar me etiketën <h3>.

Pak më poshtë, një nëntitull më të vogël e ka shënuar me etiketën <h1>.

Në fund, për një paragraf të zakonshëm dhe të gjatë me tekst, ai ka përdorur një varg prej pesë etiketash të veçanta <h1> (për secilën fjali veçmas), me arsyetimin se dëshironte që teksti të ishte shumë i rëndësishëm dhe i theksuar.

Vlerëso këtë mënyrë të përdorimit të etiketave bazë HTML. Përcakto se çfarë problemesh në hierarkinë logjike krijon një strukturë e tillë dhe si do të ndikojë ajo te motorët e kërkimit (si Google), të cilët përpiqen të kuptojnë se cila është tema kryesore e faqes. Sugjero se çfarë duhet të ndryshohet në kod.

7. Vlerëso kodin dhe përcakto se çfarë gabimesh teknike dhe logjike janë bërë:

```
<div>
```

```
<p>Shikoni ekspozitat tona më të njohura:</p>
```

```
<ul> <a href="https://muzej.mk/monaliza">
```

```
<li>Mona Liza</li>
```

```
</a>
```

```
<li>Statuja e Davidit</li>
```

```

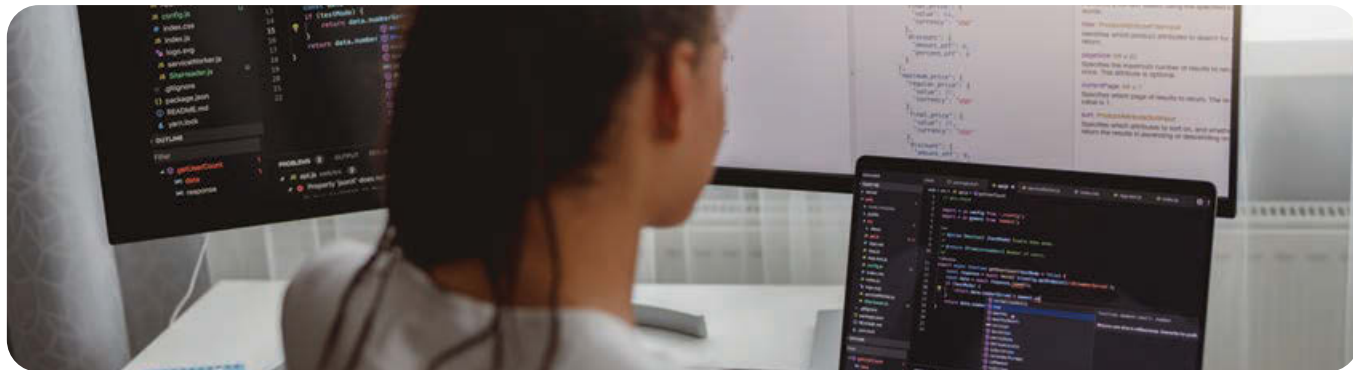
```

```
<li>Nata me yje</li>
```

```
</ul>
```

```
</div>
```

Propozo një zgjidhje për paraqitjen e saktë të listës në faqen e internetit të muzeut.



HULUMTO

Si i skanojnë robotët e Google (Web Crawlers) hierarkinë e titujve (<h1> deri te <h6>)?



MBAJ MEND ÇFARË MËSOVE

Çdo dokument HTML përbëhet nga dy pjesë: koka (head) dhe trupi (body). Në kokën e dokumentit vendosen informacionet për faqen që nuk shfaqen si përmbajtje në faqen e internetit. Përmbajtja vendoset në trupin e dokumentit. Etiketa hapëse <html> i tregon shfletuesit se aty fillon dokumenti HTML, ndërsa etiketa mbyllëse </html> tregon përfundimin e dokumentit. Koka përcaktohet nga etiketat <head> dhe </head>, ndërsa trupi nga etiketat <body> dhe </body>. <!DOCTYPE> përdoret si deklaratë që i tregon shfletuesit se dokumenti është HTML5 dhe se faqja duhet të interpretohet dhe të shfaqet në mënyrë të saktë. Në kodin bazë duhet të vendoset edhe etiketa që përcakton kodimin e karaktereve, e cila vendoset në kokën e dokumentit: <meta charset=»UTF-8»>. Etiketat kryesore për formatimin dhe strukturimin e tekstit janë paraqitur në tabelën e mëposhtme:

Etiketa (Tag)	Emri / Kuptimi	Funksioni kryesor (Roli)
<!DOCTYPE html>	Deklarimi i dokumentit	I njofton shfletuesit se po përdoret versioni më i fundit, HTML5.
<html>	Elementi rrënjë	E përfshin të gjithë kodin dhe përcakton fillimin dhe përfundimin e dokumentit HTML.
<head>	Koka e dokumentit	Përmban informacione të fshehura dhe metadatat (si <meta charset=»UTF-8»> dhe <title>).
<body>	Trupi i dokumentit	Përmban të gjithë pjesën e dukshme të faqes (tekst, imazhe dhe lidhje) që shfaqen për përdoruesin.
<h1> deri <h6>	Titujt	Përcaktojnë titujt sipas një hierarkie të qartë (nga më i rëndësishmi <h1> deri te më i vogli <h6>).
<p>	Paragrafi (Paragraph)	Shërben për organizimin dhe paraqitjen e tekstit të zakonshëm në formën e një blloku.
<a>	Hiperlidhje (Link)	Lidh faqe të ndryshme interneti përmes atributit href (destinacioni i lidhjes).
	Imazhi	Vendos një imazh në faqe përmes burimit src dhe përmban përshkrimin e detyrueshëm alt.
/	Listë e parenditur / Listë e renditur	Krijojnë lista me pika () ose lista me numra (), duke përdorur për secilin element të listës.
<div>	Ndarje / Kuti (Division)	Kontejner universal që përdoret për grupimin dhe stilizimin e mëvonshëm të elementeve.



PYETJE PËR KONTROLLIMIN E NJOHURIVE:

- Cila nga linjat e mëposhtme të kodit paraqet vendosje të SAKTË (nesting) të etiketave HTML?
 - A) <p>Mësojmë HTML modernkod.</p>
 - B) <p>Mësojmë HTML modernkod.</p>
 - C) <p>Mësojmë HTML modern</p>
 - D) <p>Mësojmë HTML modern</p>
- Çfarë do të shfaqë shfletuesi në ekran nëse në VS Code shkruan tekstin e mëposhtëm: <p>Përshëndetje studentë!</p> (me 5 hapësira boshe)?
 - A) „Përshëndetje studentë!“ (saktësisht 5 hapësira, sepse HTML-i i ruan të gjitha)

- B) „Përshëndetje studentë!“ (hapësirat do të fshihen plotësisht, sepse hapësira boshe hiqet)
 - C) „Përshëndetje studentë!“ (vetëm një hapësirë, për shkak të karakteristikës *White-space Collapse*)
 - D) Shfletuesi do të raportojë një gabim sintaksor dhe nuk do ta shfaqë fare tekstin.
3. Cili është funksioni kryesor i etiketës <div> në HTML?
- A) Shërben për ta bërë automatikisht tekstin të theksuar dhe më të madh në ekran.
 - B) Funksionon si një kontejner universal (kuti) për grupimin e elementeve të tjera.
 - C) Përdoret vetëm për vendosjen e imazheve ose lidhjeve në faqen e internetit.
 - D) E përkthen automatikisht përmbajtjen e faqes në gjuhën angleze.
4. Cila nga etiketat e mëposhtme bën pjesë në grupin e “Elementeve bosh” (Void Elements), të cilat nuk kanë etiketë mbyllëse dhe mbyllen vetë?
- A) <p>
 - B) <h1>
 - C)
 - D) <a>
5. Pse është e nevojshme të shkruhet rreshti <meta charset=»UTF-8»> në pjesën <head> të kodit, edhe pse VS Code i ruan vetë skedarët në këtë format?
- A) Për t’i treguar shfletuesit (Chrome, Safari...) se cilin standard të karaktereve duhet të përdorë, në mënyrë që alfabeti cirilik të shfaqet pa karaktere të pakuptueshme.
 - B) Për t’i treguar programit VS Code se ke lehtë të shkruash kod në gjuhën maqedonase.
 - C) Për të aktivizuar shkurtesat automatike (Emmet) për shkrim më të shpejtë të kodit.
 - D) Me këtë përcaktohet titulli kryesor që shfaqet në skedën e shfletuesit.
6. Mendo sikur po krijon një faqe interneti dhe dëshiron të vendosësh një imazh të logos së shkollës sate. Shkruaj rreshtin e plotë HTML për vendosjen e imazhit dhe shpjego shkurtimisht pse është e rëndësishme të specifikosh atributin alt në atë etiketë (jep të paktën një arsye).

PËR ATA QË DËSHIROJNË TË DINË MË SHUMË



HTML-ja moderne nuk shërben vetëm për shfaqjen e tekstit dhe imazheve. Ajo vjen me teknologji të avancuara të integruara (API), të cilat në të kaluarën kërkonin programe komplekse, ndërsa sot mund të aktivizohen me vetëm disa rreshta kodi:

- 1. Gjeolokacioni (Geolocation):** Përmes HTML-së dhe pak JavaScript-i, mund t’i kërkojë leje përdoruesit që faqja të përcaktojë saktësisht vendndodhjen e tij në hartë.
- 2. Ruajtja lokale (Local Storage):** HTML5 i mundëson faqes tënde të ruajë të dhëna drejtpërdrejt në shfletuesin e përdoruesit (si zgjedhja e mënyrës “Dark Mode” ose produktet në shportë). Edhe nëse përdoruesi e fik kompjuterin dhe kthehet të nesërmen, faqja do ta mbajë mend zgjedhjen e tij.
- 3. Transmetimi i audios dhe videos:** Etiketat <audio> dhe <video> kanë kontrolle të avancuara që mundësojnë krijimin e një luajtësi tëndin, të ngjashëm me YouTube ose Spotify, drejtpërdrejt në shfletues, pa pasur nevojë për shtojca të jashtme.



5.2.2 EDITUES PËR SHKRIMIN E KODIT HTML

Një faqe interneti është një dokument që përmban kod HTML dhe ka prapashitesën .html. Hape faqen tënde të preferuar të internetit, kliko me të djathtën në cilëndo pjesë të saj dhe nga menjya zgjidh View page source. Do të shfaqet kodi HTML i faqes, i cili duket si një dokument tekstual dhe mund të shkruhet në një redaktues teksti, si Notepad ose Notepad++. Nuk rekomandohet shkrimi i kodit në përpunues teksti (Microsoft Word, Apple Mac Pages, etj.), sepse ata krijojnë skedarë që shfletuesi nuk mund t'i lexojë siç duhet. Kodi HTML mund të shkruhet edhe në redaktues të tjerë, si Visual Studio Code, i cili duhet të instalohet dhe është falas, ose në redaktues online, si CodePen.io, OneCompiler dhe të tjerë. Në këtë tekst mësimor, për të mësuar HTML-në, do të përdoret redaktuesi Visual Studio Code (VS Code), i cili është me burim të hapur dhe përfaqëson një standard të industrisë për zhvillimin web. Gjatë shkrimit të kodit, ai ofron sugjerime inteligjente, shfaq automatikisht skedarët e disponueshëm dhe mundëson shfaqjen e faqes së internetit në një shfletues të hapur drejtpërdrejt nga VS Code.

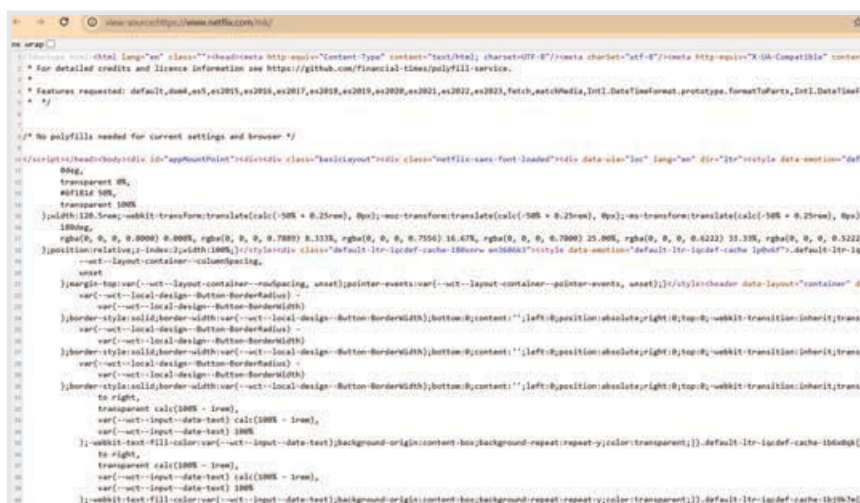


Figura 1: Pamja e kodit HTML

5.2.3. INSTALIMI I EDITUESIT HTML VISUAL STUDIO CODE

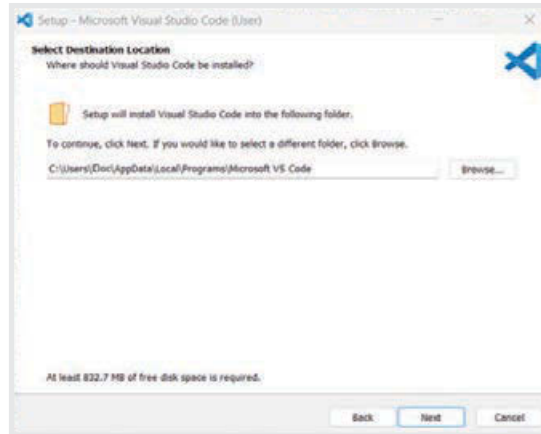
Instalimi i Visual Studio Code (VS Code) është një proces shumë i shpejtë dhe i thjeshtë. Hap faqen zyrtare të Visual Studio Code dhe zgjidh versionin që përputhet me sistemin tënd operativ. Kliko versionin e dëshiruar dhe prit disa sekonda derisa skedari i instalimit (me prapashitesën .exe) të shkarkohet në kompjuterin tënd.

1. Gjeje skedarin e shkarkuar (për shembull, VSCodeUserSetup-x64.exe) në kompjuterin tënd dhe kliko dy herë (double-click) mbi të për të filluar procesin e instalimit.
2. Aktivizohet një magjistar (Wizard) që të udhëheq nëpër procedurë. Fillimisht shfaqet dritarja me Marrëveshjen e licencës (License Agreement). Lexoje dhe zgjidh opsionin „I accept the agreement“, e më pas kliko butonin Next (Vazhdo).
3. Më pas, zgjidh vendndodhjen në kompjuterin ku dëshiron ta ruash skedarin.



Figura 2: Pranimi i kushteve të marrëveshjes

Figura 3: Përcaktimi i vendndodhjes së instalimit



4. Kur të hapet dritarja me titullin "Select Additional Tasks", do të kesh mundësinë të zgjedhësh opsionet e mëposhtme:
- Create a desktop icon: Për të krijuar një shkurtore (ikonë) të VS Code drejtpërdrejt në desktopin tënd, për hapje të shpejtë.
 - Add "Open with Code" to Windows Explorer: Ka dy opsione të tilla; zgjidhi të dyja. Ky është një opsion i dobishëm. Më vonë, kur të kesh një dosje me projekte HTML, mund të klikosh me të djathtën mbi dosje dhe të zgjedhësh "Open with Code", me ç'rast i gjithë projekti do të hapet menjëherë në redaktues.
 - Register Code as an editor for supported file types: Mundëson që, me dy klikime mbi cilindo skedar .html ose .css, ai të hapet automatikisht në VS Code.
 - Add to PATH: Ky opsion është i aktivizuar si parazgjedhje dhe mundëson që programi të funksionojë siç duhet me sistemin operativ.

Pasi t'i zgjedhësh këto opsione, kliko **Next** për të filluar procesin e instalimit.

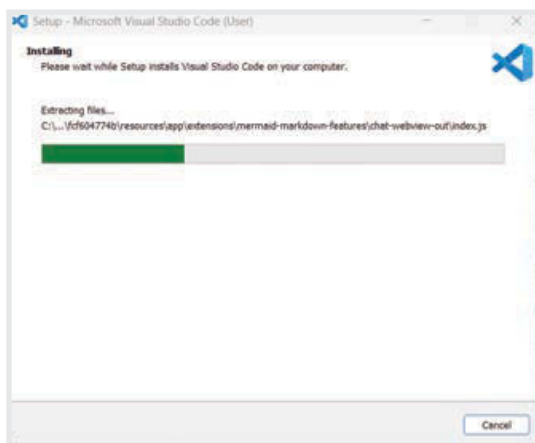


Figura 4: Procesi i instalimit

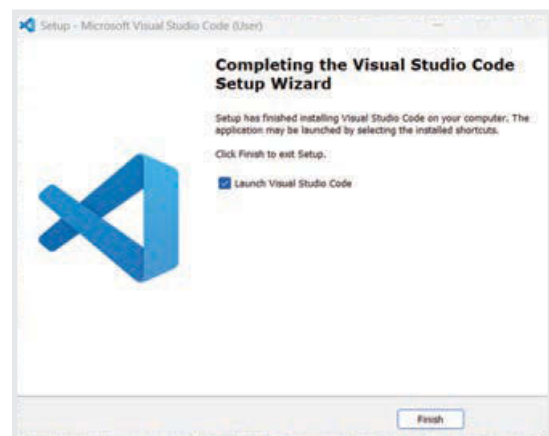


Figura 5: Instalimi i përfunduar i VS Code

5. Procesi i instalimit zgjat rreth 10 deri në 20 sekonda, derisa vija e gjelbër të mbushet plotësisht. Me klikimin e butonit Finish, në redaktues (editor) hapet dritarja fillestare e punës.

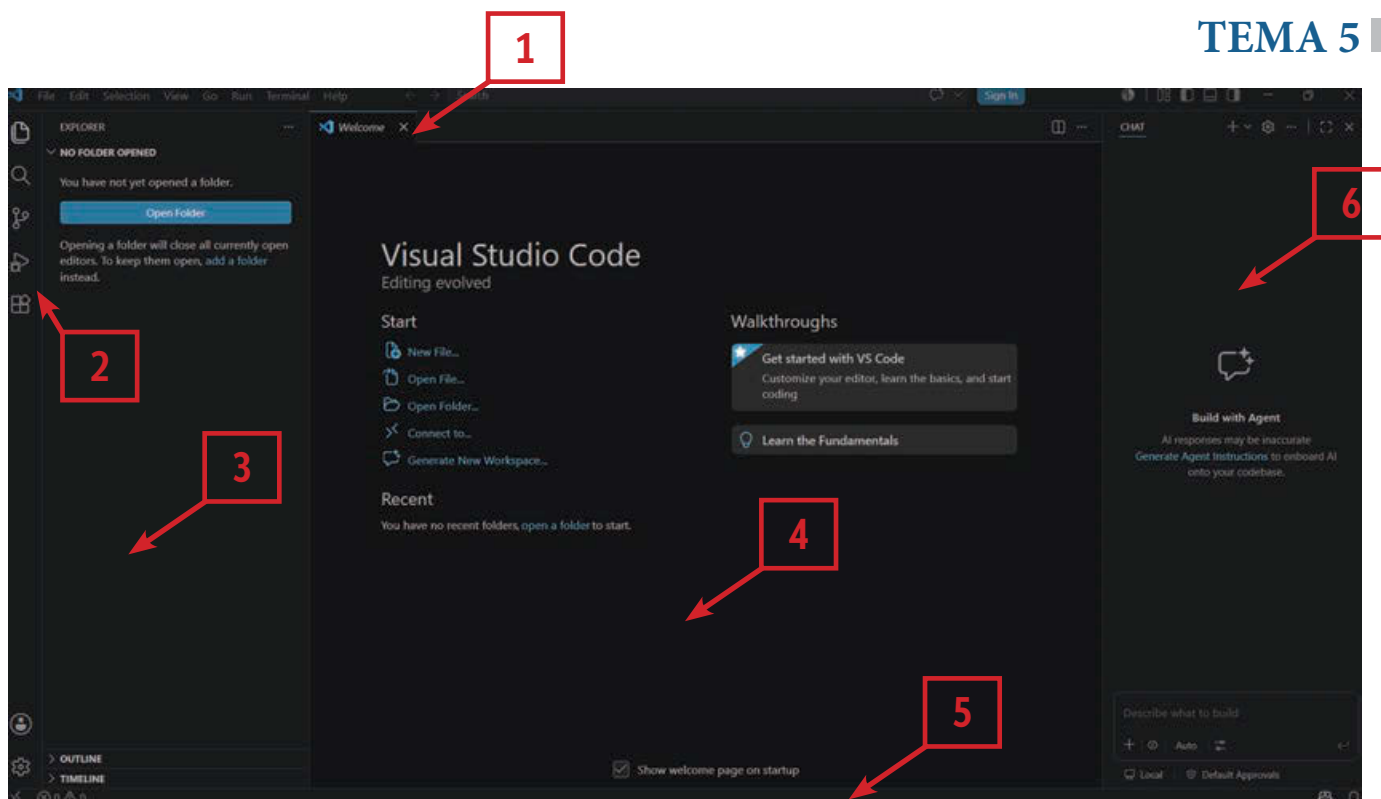


Figura 6: Dritarja fillestare e punës në VS Code

1. Skeda fillestare: „Welcome“ (Mirë se vini)

Në qendër të ekranit do të hapet një skedë e madhe e quajtur Welcome. Kjo është faqja jote fillestare, ku gjenden shkurtoret për:

- **New File** – për krijimin e një dokumenti të ri dhe bosh.
- **Open Folder** – për hapjen e një dosjeje nga kompjuteri (opsioni më i rëndësishëm për programuesit).
- **Clone Git Repository** – për përdoruesit më të avancuar që shkarkojnë kod nga interneti.

2. Shiriti i mjeteve në skajin e majtë (Activity Bar)

Ky është shiriti vertikal me ikona në skajin e majtë të dritares. Këto ikona përfaqësojnë mënyrat kryesore të punës:

- **Explorer (ikona me dy dosje):** Këtë do ta përdorësh 95% të kohës. Kur klikon këtu, hapet paneli anësor ku shfaqen dosjet dhe skedarët e tu.
- **Search (lupa):** Për kërkimin dhe zëvendësimin e fjalëve në të gjithë skedarët njëkohësisht.
- **Source Control (degë/nyje):** Përdoret për Git dhe menaxhimin e versioneve të kodit.
- **Run and Debug (insekt me trekëndësh):** Për testimin e kodit dhe gjetjen e gabimeve.
- **Extensions (katër katrorë, njëri prej tyre i shkëputur):** Dyqani i shtesave falas. Nga këtu mund të instalosh mjete të dobishme për shkrimin më të lehtë të kodit.

3. Paneli anësor (Sidebar)

Ky hapësirë ndodhet menjëherë në të djathtë të shiritit të mjeteve. Nëse është aktivizuar opsioni Explorer, këtu do të shfaqet struktura e projektit tënd – skedarët HTML, dosjet për imazhe dhe elemente të tjera. Në fillim, kur nuk ke hapur ende një dosje, këtu shfaqet një buton i madh blu “Open Folder”.

4. Hapësira kryesore për redaktim (Editor)

Kjo është pjesa më e madhe dhe qendrore e ekranit. Kur klikon mbi një skedar (për shembull, index.html), përmbajtja e tij do të hapet pikërisht këtu. Ky është „kanavaca“ jote, ku do të shkruash

dhe ndryshosh kodin HTML. Mund të hapësh disa skedarë njëkohësisht dhe ata do të renditen në skeda në pjesën e sipërme, njësoj si në Chrome.

5. Shiriti i poshtëm i statusit (Status Bar)

Në pjesën e poshtme të dritares ndodhet një shirit i hollë (zakonisht me ngjyrë blu ose vjollcë). Ai të jep informacione të menjëhershme për skedarin me të cilin po punon: në cilin rresht ndodhesh, cili gjuhë programimi është aktive (poshtë djathtas duhet të shkruajë HTML) dhe çfarë kodimi përdoret (zakonisht UTF-8). Dritarja është krijuar në mënyrë që të mos ta shpërqendrojë vëmendjen: në të majtë është organizimi i skedarëve, ndërsa në qendër ndodhet vetë kodi.

6. Hapësira e punës me asistentin e IA-së (AI Assistant).

5.2.4. PUNA ME REDAKTUESIN

Procedura e fillimit të punës në VS Code është shumë e thjeshtë dhe e shpejtë. Kur e hap programin për herë të parë, gjëja më e rëndësishme që duhet të mbash mend është se **skedarët HTML nuk mund të qëndrojnë të veçuar** – ata duhet të ruhen gjithmonë në mënyrë të rregullt në një dosje të veçantë në kompjuter. Më poshtë janë shpjeguar hapat një nga një:

Hapi 1: Krijo një dosje për projektin

Para se të hapësh **Visual Studio Code (VS Code)**, krijo një dosje të re dhe bosh në kompjuterin tënd. Emërtoje thjesht, për shembull: *moj-prv-proekt* (është më mirë të përdorësh shkronja të vogla latine, pa hapësira dhe një emër që lidhet me përmbajtjen e faqes së internetit).

Hapi 2: Hape dosjen në VS Code

Tani hape VS Code. Ke dy mënyra për ta hapur dosjen brenda programit:

1. Përmes menysë **File -> Open Folder...** dhe zgjidh dosjen që krijove në Hapin 1.
2. Ose thjesht **tërhiqe dhe lëshoje (Drag, Drop)** dosjen me anë të mausit drejtpërdrejt në hapësirën bosh të VS Code.

Pse është e rëndësishme kjo? Në këtë mënyrë, VS Code e kupton se gjithçka që do të punosh më tej i përket atij projekti konkret. Ana e majtë e ekranit bëhet hapësira jote e punës (Explorer).

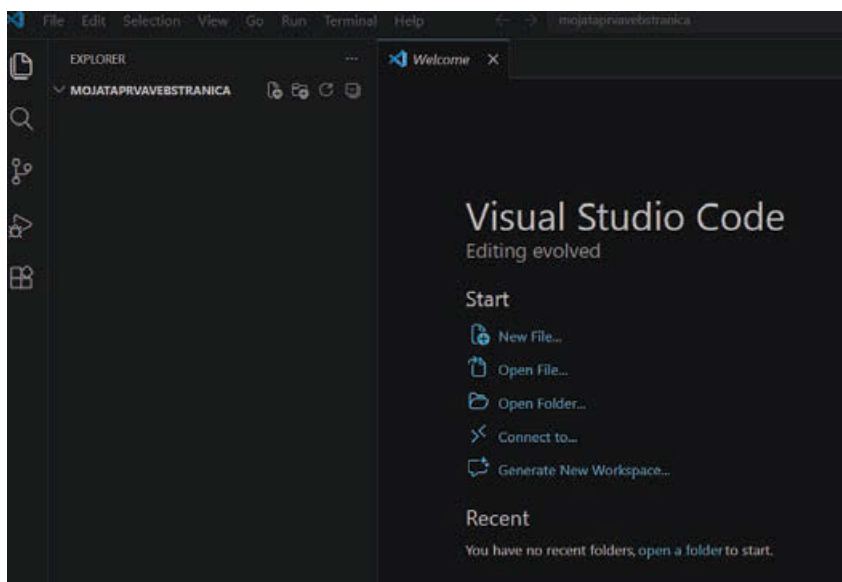


Figura 7: Krijimi i një dosjeje në Visual Studio Code

Hapi 3: Krijë një skedar të ri HTML

Në anën e majtë, në seksionin ku shfaqet emri i dosjes tënde, kliko ikonën e vogël **New File** (duket si një fletë letre me shenjën plus).

Shkruaj emrin e skedarit: **index.html** dhe shtyp Enter.



Figura 8: Skedari i emërtuar në Visual Studio Code

E rëndësishme: Emri duhet të përfundojë me **.html**, në mënyrë që kompjuteri ta kuptojë se bëhet fjalë për një faqe interneti dhe jo për një dokument të zakonshëm tekstual. Emri **index** është standardi ndërkombëtar për faqen kryesore (fillestare) të çdo sajti.

Hapi 4: Aktivizo skeletin fillestar

Tani, pasi skedari është hapur dhe është bosh:

1. Në rreshtin e parë shkruaj vetëm një pikëçuditëse: **!**
2. Do të shfaqet një menu e vogël (shkurtesa Emmet). Shtyp **Enter** ose **Tab**.
3. VS Code do të gjenerojë automatikisht të gjithë kodin bazë të nevojshëm fillestar (skeletin me `<html>`, `<head>`, `<body>` dhe përcaktimin e kodimit UTF-8).

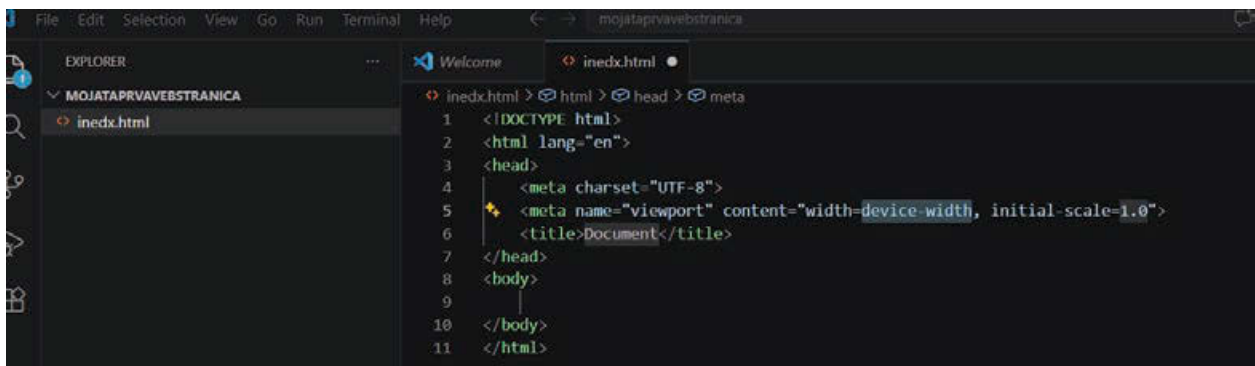


Figura 9: Skeleti bazë i krijuar i faqes së internetit

Hapi 5: Shkruaj kodin tënd të parë

Shfletuesi shfaq në ekran vetëm atë që ndodhet midis etiketës fillestare `<body>` dhe etiketës përmbyllëse `</body>`.

Vendose kursorin poshtë etiketës së parë `<body>` dhe shkruaj:

```
<body>
  <h1>Përshëndetje, Botë!</h1>
  <p>Kjo është faqja ime e parë e internetit e shkruar në VS Code.</p>
</body>
```

Hapi 6: Ruaj kodin (Save)

Në pjesën e sipërme, në skedën ku shkruan index.html, do të vësh një rreth të vogël të bardhë. Kjo do të thotë: „Ke shkruar kod të ri që ende nuk është ruajtur!“.

- Zgjidh shkurtoren e tastierës: **Ctrl + S** (për Windows) ose **Cmd + S** (për Mac).
- Rrethi i bardhë do të zhduket dhe do të shndërrohet në shenjën x. Tani kodi yt është ruajtur në mënyrë të sigurt në hard disk.

Me këtë, kodi yt i parë është i përfunduar dhe i gatshëm për t'u parë.

Ka disa mënyra për ta shfaqur faqen e internetit, por më poshtë do të shpjegohen tri prej tyre.

Mënyra 1: Hapja drejtpërdrejt nga dosja. Kjo është mënyra më e thjeshtë, e cila funksionon në çdo kompjuter dhe nuk kërkon internet ose instalimin e shtojcave shtesë.

- Hap dosjen në kompjuterin tënd ku e ke ruajtur skedarin index.html.
- Kliko dy herë mbi skedarin index.html.
- Kompjuteri do ta hapë automatikisht skedarin në shfletuesin tënd standard (Chrome, Edge, Firefox).
-

Shënim i rëndësishëm për këtë mënyrë: Sa herë që ndryshon diçka në kodin në VS Code, fillimisht duhet ta ruash ndryshimin me **Ctrl + S** dhe më pas duhet të hapësh manualisht shfletuesin dhe të klikosh butonin **Refresh** ose të shtypësh **F5** në tastierë për të parë ndryshimet.

Mënyra 2: Me shtojcën „Live Preview“ (paraqitje e drejtpërdrejtë në VS Code)

Nëse nuk ke dy monitorë ose nuk dëshiron ta ndash ekranin në dy pjesë me Chrome, Microsoft krijoi një shtojcë që hap një mini-shfletues drejtpërdrejt brenda vetë programit.

1. Zgjidh **Extensions** (ikona me katër katrorë në anën e majtë) dhe kërko **Live Preview** (shtojca zyrtare nga Microsoft).
2. Kliko **Install**.

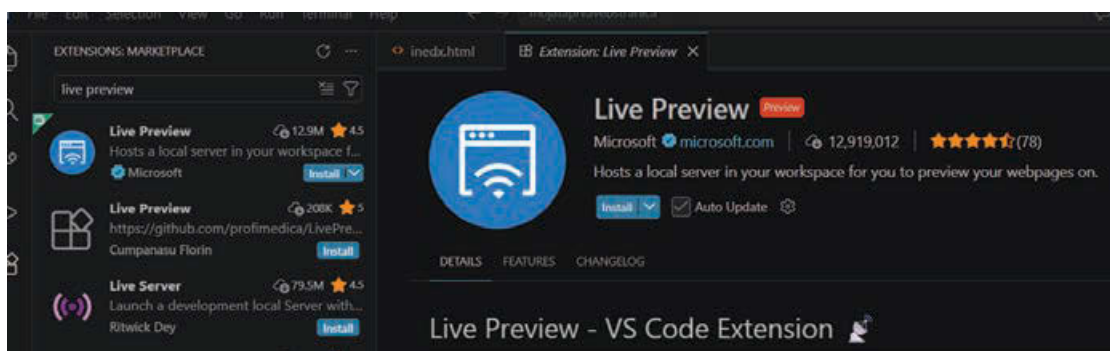


Figura 10: Instalimi i Live Preview

3. Sapo të përfundojë instalimi, butoni do të ndryshojë në një ikonë me ingranazh (që tregon se shtojca është instaluar). Që nga ai moment, shtojca është e integruar në VS Code dhe është e gatshme për t'u përdorur me një klikim në ikonën që ngjan me një ekran të vogël, në këndin e sipërm djathtas
4. Kur të kthehesh te kodi yt index.html, në këndin e sipërm djathtas të ekranit do të vësh një ikonë të re që ngjan me **një ekran të vogël kompjuteri me një lupë**.

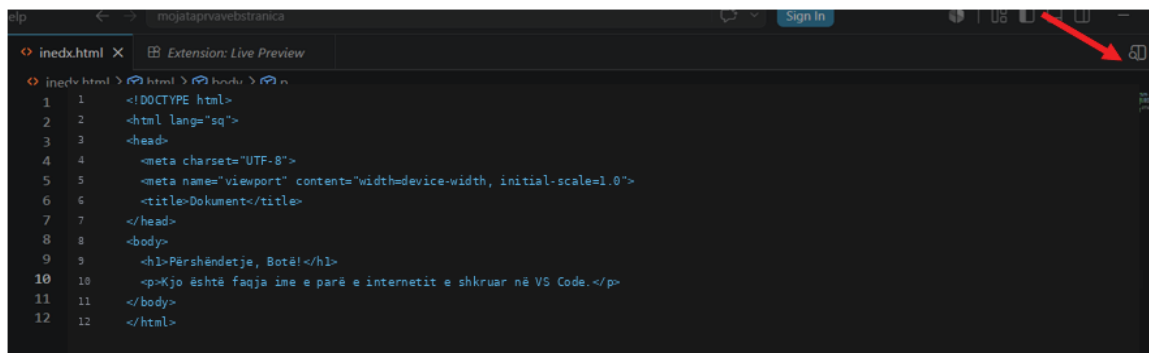


Figura 11: Live Preview i instaluar

5. Kliko në atë ikonë dhe ekran i VS Code do të ndahet në dy pjesë: në të majtë do të shfaqet kodi, ndërsa në të djathtë do të shfaqet menjëherë faqja jote në kohë reale. Çdo ndryshim i ruajtur do të shfaqet menjëherë aty.

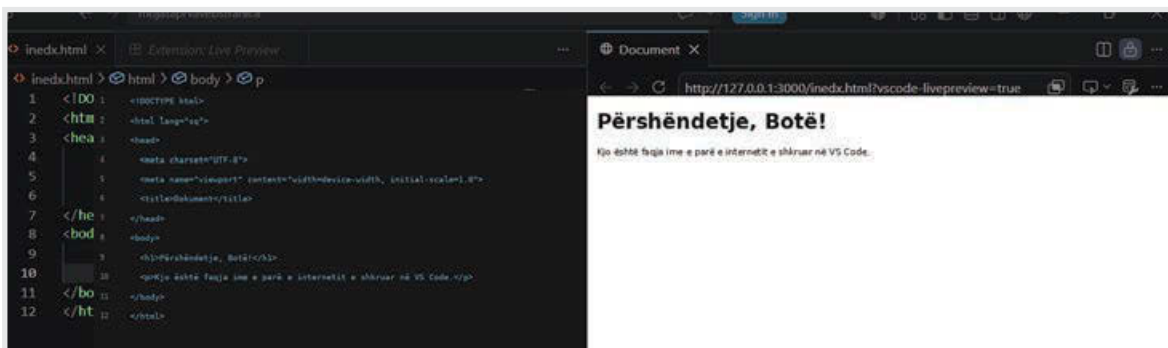


Figura 12: Hapësira e punës dhe paraqitja e faqes së internetit

Mënyra 3: Përmes asistentit të AI-së në anën e djathtë

Meqenëse tashmë e ke dritaren e AI-së në anën e djathtë, mund të shfrytëzosh edhe mundësitë e saj për një paraqitje të shpejtë.

1. Përzgjidh të gjithë kodin tënd HTML.
2. Pyete asistentin e AI-së në chat: „Si do të duket ky kod kur të renderohet?“ ose „Show me a preview of this HTML“.
3. Shumë nga agjentët modernë të AI-së (si Copilot ose Cursor) kanë një buton të integruar „**Preview**“ drejtpërdrejt në dritaren e tyre, me një klikim të të cilit ata të shfaqin një pamje vizuale të mënyrës se si duket dizajni i faqes në atë moment.

Për mësimin fillestar, më e mira është Mënyra 2 (Live Preview), sepse gjithçka ndodhet në një vend dhe nuk ke nevojë ta lëvizësh mausin jashtë VS Code.





PYETJE PËR KONTROLLIMIN E NJOHURIVE:

1. Cila kombinim tastesh është standardi botëror në VS Code për ruajtjen e ndryshimeve në kodin tënd?

- A) Ctrl + C
- B) Ctrl + S
- C) Ctrl + V
- D) Ctrl + Z

2. Çfarë tregon ikona me katër katrorë (ku njëri katror është i ndarë) në shiritin vertikal të majtë, në VS Code?

- A) Menyja për kërkimin e fjalëve në kod
- B) Menyja për punë me dosje dhe skedarë (Explorer)
- C) Dyqani për instalimin e shtojcave (Extensions), si Live Preview
- D) Seksioni ku hapet inteligjenca artificiale për bisedë

3. Në vend që të përdorësh shtojca për paraqitje, e hap faqen tënde të internetit me dy klikime drejtpërdrejt nga dosja e kompjuterit në shfletues. Pasi të shtosh kod të ri HTML në VS Code dhe ta ruash me Ctrl + S, cili është hapi i radhës i domosdoshëm për ta parë ndryshimin në ekran në Chrome?

- A) Duhet ta mbyllësh plotësisht shfletuesin dhe ta hapësh përsëri me dy klikime nga dosja.
- B) Duhet të klikosh manualisht butonin Refresh (Rifresko / Ringarko përsëri) në shfletues ose të shtypësh butonin F5 në tastierë.
- C) Nuk duhet të bësh asgjë; faqja do të ndryshojë vetë në të njëjtin moment sapo të shtypësh Ctrl + S.
- D) Duhet ta fshish skedarin e vjetër HTML nga dosja dhe të krijosh një të ri.

4. Në vend që ta shkruash manualisht të gjithë skeletin fillestar të dokumentit HTML, cilin simbol (shkurtore) duhet të shkruash në një skedar bosh që VS Code, përmes Emmet, ta gjenerojë automatikisht kodin me një shtypje të Enter?

- A) Pikëpyetje (?)
- B) Hashtag (#)
- C) Pikëçuditëse (!)
- D) Vijë e pjerrët (/)



5. Kur punon me asistentin e integruar të AI-së (në anën e djathtë të ekranit), cila është mënyra më efikase për t'i treguar agjentit të të shpjegojë ose të korrigjojë vetëm një rresht specifik të kodit tënd?

- A) Duhet ta rishkruash të gjithë kodin me dorë në dritaren e bisedës.
- B) Mjafton ta përzgjedhësh (ta shënosh) atë rresht të kodit me miun në anën e majtë dhe më pas ta bësh pyetjen në dritaren e AI-së në anën e djathtë.
- C) Duhet ta ruash skedarin në një format të veçantë .txt dhe t'ia dërgosh asistentit.
- D) Asistentët e AI-së nuk mund të shohin atë që shkruan, prandaj duhet t'i tregosh rreshtin e saktë (p.sh. „Rreshti 14“).

6. Cili është rendi i saktë i hapave që duhet të ndjekësh për të filluar një projekt të ri në një VS Code bosh, në mënyrë që programi të fillojë menjëherë të të ndihmojë (të ngjyrosë etiketat dhe të japë sugjerime)?

- A) Fillimisht shkruan kodin, pastaj krijon një skedar të ri dhe në fund e vendos në një dosje.
- B) Fillimisht ruan një tekst bosh në desktop, pastaj e riemërton në dosje dhe më pas hap Chrome.
- C) Fillimisht krijon një skedar të ri, shkruan pikëçuditësen dhe vetëm në fund mendon se ku do ta ruash në kompjuter.
- D) Fillimisht hap dosjen e projektit (**Open Folder**), brenda saj krijon një skedar të ri (**New File**) dhe menjëherë e ruan me prapashtesën **.html**.
- E) Rendi nuk është aspak i rëndësishëm; VS Code e di vetë se çfarë dëshiron të shkruash edhe para se ta ndezësh kompjuterin.

PËR ATA QË DËSHIROJNË TË DINË MË SHUMË



Nëse shkruan shpesh të njëjtën strukturë (për shembull, skeletin e një karte, një tabelë me klasa specifike ose një meny), nuk ka nevojë ta rishkruash vazhdimisht ose të bësh Copy/Paste nga projektet e vjetra. Mund të krijosh **shkurtoren tënde të personalizuar Emmet**. Kliko te ikona e ingranazhit poshtë majtas (Settings) dhe zgjidh **User Snippets**. Më pas zgjidh **html.json** dhe aty mund të përcaktosh shkurtoren tënde. Për shembull, nëse përcakton shkurtoren mojakarticka, në momentin që e shkruan atë në një skedar bosh HTML dhe shtyp Tab, VS Code do të ngarkojë automatikisht të gjithë kodin e paracaktuar, me tituj, klasa dhe imazhe. Gjithashtu, kur skedari yt HTML të ketë 500 ose 1000 rreshta, „lëvizja“ lart e poshtë bëhet një makth. Programuesit e avancuar përdorin navigim inteligjent.

1. **Breadcrumbs (Gjurmët e navigimit):** Në pjesën e sipërme të kodit shfaqet gjithmonë një shteg navigimi (p.sh. src > index.html > body > div.container > table). Mund të klikosh në cilëndo etiketë në këtë shteg për të „kërcyer“ menjëherë te ajo pjesë, pa pasur nevojë të lëvizësh me scroll.
2. **Outline (Struktura):** Në pjesën e poshtme majtas (poshtë dosjeve) ndodhet një menu e quajtur Outline. Ajo ta paraqet dokumentin HTML si një strukturë në formë peme. Nëse klikon mbi një etiketë në atë pemë, redaktori të çon menjëherë te ajo pjesë e kodit.

5.2.5. PËRDORIMI I ETIKETAVE HTML

Në gjuhën HTML, për t'u shtuar karakteristika shtesë etiketave përdoren atributet. Nëse vetë etiketat janë skeleti i faqes së internetit (tregojnë se ku ndodhet titulli, ku ndodhet paragrafi etj.), atributet u japin atyre etiketa informacione shtesë. Atributet shkruhen gjithmonë brenda etiketës fillestare (asnjëherë në etiketën përmbyllëse). Ato shkruhen në formën: emri_i_atributit="vlera", pra kanë një emër dhe një vlerë. Emri i atributit shkruhet me shkronja të vogla, ndërsa vlera e atributit duhet të vendoset gjithmonë midis thonjzave të dyfishta (,,") ose thonjzave të vetme ('). Më shpesh përdoren thonjzat e dyfishta. Thonjzat e vetme përdoren kur brenda vlerës ka thonjza të dyfishta, për shembull: ime='bulevardi „8 Shtatori“ '.

Gjithashtu, që etiketa e lidhjes (<a>) të dijë se ku duhet të na „çojë“, përdorim atributin href.

```
<a href="https://www.google.com">Shko te Google</a>
```

Për hiperlidhjet përdoren edhe attribute të tjera: **target="_blank"** – e hap lidhjen në një **skedë të re**, në mënyrë që përdoruesi të mos largohet nga faqja jote, ndërsa **title** – shfaq një këshillë të vogël tekstuale (tooltip) kur mban kursurin e miut mbi lidhje.

```
<a href="https://mk.wikipedia.org" title="Shko te Wikipedia">Vizito Wikipedia</a>
(hiperlidhje e thjeshtë)
```

```
<a href="https://www.google.com" target="_blank">Hape Google në një skedë të re</a>
(hiperlidhje që hapet në një skedë të re)
```

Etiketa për imazh () është specifike, sepse është e pavarur (nuk ka etiketë përmbyllëse). Për të shfaqur një imazh, ajo përdor atributet src (burimi i imazhit) dhe alt (teksti alternativ):

```

```

Ekzistojnë edhe attribute për gjerësinë dhe lartësinë e imazhit, width dhe height, me të cilat madhësia e imazhit shprehet në pikselë.

```

```

Një nga përdorimet më të zakonshme në internet është kur dëshiron që **klikimi mbi vetë imazhin të të çojë te një lidhje** (për shembull, kur klikon mbi një logo për t'u kthyer në faqen kryesore).

Për ta bërë këtë, thjesht vendosim etiketën brenda, midis etiketës fillestare dhe asaj përmbyllëse të <a>.

```
<a href="https://www.youtube.com" target="_blank">
  
</a>
```

Të gjitha këto etiketa mund të kombinohen në një kod të vetëm për të krijuar një faqe interesante. Për shembull, kodi i mëposhtëm krijon një faqe me një imazh dhe një hiperlidhje që të çon te më shumë informacione rreth imazhit:

```
<h1>Foto e një këlyshi të vogël</h1>
```

```

```

```
<br> <a href="https://sq.wikipedia.org/wiki/Qeni" target="_blank">Lexo më shumë për qentë në
Wikipedia</a>
```

Për krijimin e një table në HTML, përdoren attribute të veçanta për bashkimin e qelizave. Kur krijojmë një tabelë, përdorim një etiketë kryesore, brenda së cilës vendosen disa etiketa ndihmëse për rreshtat dhe kolonat:

Etiketat kryesore për tabelën:

- **<table>** – Fillon dhe përfundon të gjithë tabelën.
- **<tr>** (Table Row) – Krijon një **rresht të ri horizontal**.
- **<td>** (Table Data) – Krijon një **qelizë të zakonshme (kolonë)** në atë rresht, ku vendosen të dhënat.
- **<th>** (Table Header) – Krijon një **qelizë për titullin** (teksti automatikisht shfaqet me të trasha dhe i qendëruar).

Nëse krijojmë një orar mësimor dhe një lëndë zhvillohet për dy orë, atëherë duhet të bashkojmë dy kolona ose dy rreshta. Për këtë përdorim dy atributet e mëposhtme, të cilat shkruhen ekskluzivisht brenda etiketave <td> ose <th>:

Atributi colspan (Bashkimi i kolonave)

Ky atribut i tregon qelizës të „shtrihet“ horizontalisht në më shumë se një kolonë njëherësh.

- **Shembull:** Nëse ke dy orë radhazi të lëndës së Informatikës, në vend që të krijosh dy qeliza të veçanta, krijon një të vetme si kjo:

```
<td colspan="2">Informatikë (orë bllok)</td>
```

Atributi rowspan (Bashkimi i rreshtave)

Ky atribut e „shtrin“ qelizën vertikalisht, pra poshtë përgjatë disa rreshtave.

- **Shembull:** Nëse çdo ditë, në të njëjtën kohë, ke „Pushim të madh“, mund t'i bashkosh rreshtat për ditët:

```
<td rowspan="3">Pushim i madh</td>
```

Kodi i plotë për lidhjen e të gjitha këtyre etiketave dhe attributeve:

```
<table border="1"> <tr>
<th>Emri</th>
<th>Mbiemri</th>
<th>Lënda</th>
</tr>
<tr>
<td>Marko</td>
<td>Petrovski</td>
<td>IT</td>
</tr>
<tr>
<td>Ana</td>
<td>Stojanova</td>
<td>Matematikë</td>
</tr>
</table>
```



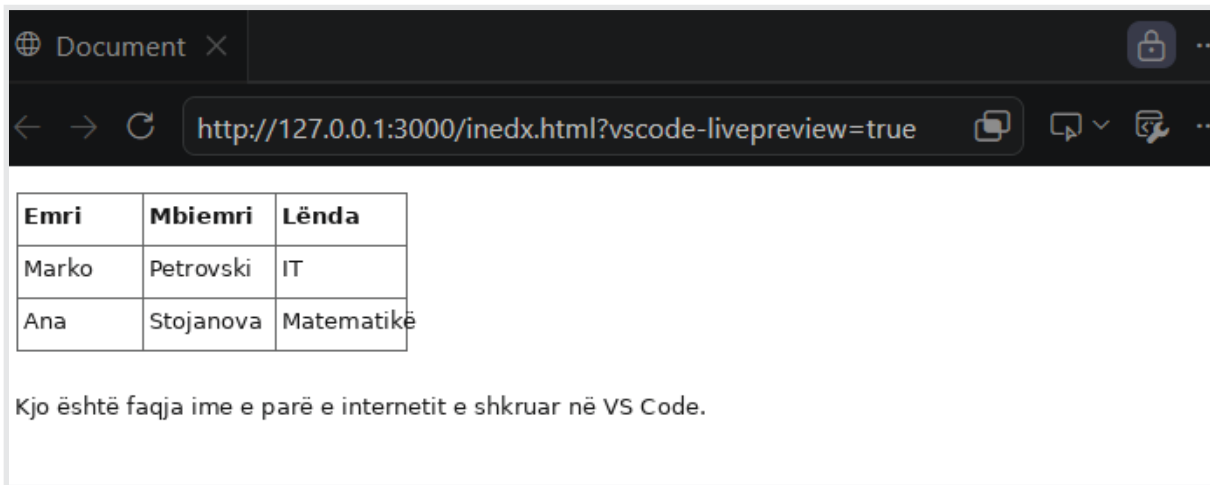


Figura 13: Orari i krijuar me HTML

Njohuritë për krijimin e tabelave mund t'i përdorim për të krijuar një formular regjistrimi për një kurs, i cili do të përmbajë tabelë, fusha për të dhëna dhe bashkim të qelizave. Ky shembull është edhe më i avancuar, sepse përdor një tabelë pa kornizë (rrjetë të padukshme) për të rreshtuar në mënyrë të përsosur fushat e plotësimit dhe, në fund, përdor colspan për ta shtrirë butonin në të dyja kolonat.

<h2>Regjistrimi në kursin për etiketat HTML</h2>

<p>Të lutemi plotëso fushat e detyrueshme për të rezervuar vendin tënd.</p>

```

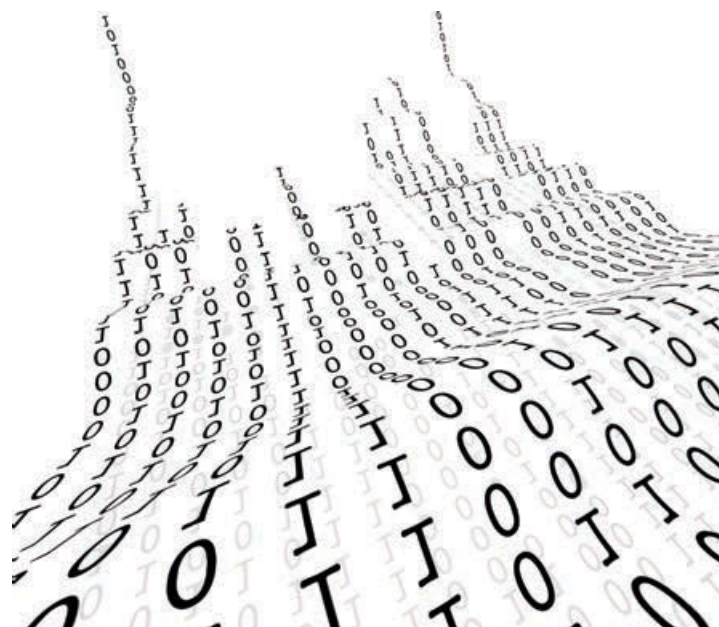
<table>
  <tr>
    <td><b>Emri dhe mbiemri:</b></td>
    <td><input type="text" placeholder="Shkruani emrin tuaj..." required></td>
  </tr>

  <tr>
    <td><b>E-maili:</b></td>
    <td><input type="email" placeholder="shembull@email.com" required></td>
  </tr>

  <tr>
    <td><b>Data e lindjes:</b></td>
    <td><input type="date"></td>
  </tr>

  <tr>
    <td colspan="2" style="text-align: center;">
      <br>
      <input type="submit" value="Dërgo aplikimin">
    </td>
  </tr>
</table>

```



Në tabelat e mëposhtme janë paraqitur të gjitha etiketat dhe atributet, së bashku me përshkrimet e tyre, që përdoren në HTML.

Etiketa	Çfarë tregon?	Përshkrimi / Si përdoret?	A ka etiketë përmbyllëse?
<h1>deri te <h6>	Tituj	<h1> është titulli kryesor dhe më i madh, ndërsa <h6> është nëntitulli më i vogël.	Po (p.sh. </h1>)
<p>	Paragraf	Përdoret për të shkruar tekst të zakonshëm.	Po (</p>)
<a>	Hiperlidhje	Krijon një lidhje që mund të klikohet.	Po ()
	Imazh	Vendos një imazh në faqe.	Jo (është etiketë e pavarur)
<table>	Tabelë	Fillon dhe përfundon të gjithë tabelën.	Po (</table>)
<tr>	Rresht në tabelë	Krijon një rresht të ri horizontal brenda tabelës.	Po (</tr>)
<td>	Fushë (qelizë)	Qelizë e zakonshme me të dhëna brenda rreshtit.	Po (</td>)
<th>	Fushë për titullin	Fushë për titullin e kolonës (teksti shfaqet me shkronja të trasha).	Po (</th>)

	Rresht i ri	E kalon tekstin ose elementin në një rresht të ri.	Jo (është etiketë e pavarur)

Tabela 2: Etiketat në HTML

Atributi	Në cilën etiketë vendoset?	Çfarë kontrollon? (Përshkrim)	Shembull si shkruhet
href	Në etiketën e lidhjes (<a>)	Tregon se ku (në cilën lidhje) do të na çojë kur të klikojmë.	href="https://google.com"
target	Në etiketën e lidhjes (<a>)	Vlera "_blank" tregon që lidhja të hapet në një skedë të re.	target="_blank"
src	Në etiketën e imazhit ()	Tregon shtegun ose lidhjen e saktë nga ku merret imazhi.	src="qen.jpg"
alt	Në etiketën e imazhit ()	Tekst alternativ që e përshkruan imazhin me fjalë.	alt="Këlysh i zi"
width	Në imazh () ose video	Përcakton gjerësinë e elementit në pikselë.	width="300"
colspan	Në qelizat e tabelës (<td> / <th>)	Bashkon disa kolona horizontalisht (p.sh. për një orë të dyfishtë).	colspan="2"
rowspan	Në qelizat e tabelës (<td> / <th>)	Bashkon disa rreshta vertikalisht në tabelë.	rowspan="2"
border	Në etiketën e tabelës (<table>)	Shton një kornizë të dukshme (vija) rreth tabelës.	border="1"
title	Në çdo etiketë	Shfaq një flluskë të vogël me tekst kur vendos kursoren e miut mbi të.	title="Më shumë informacion"

Është shumë e lehtë të krijosh një prezantim të thjeshtë të një produkti ose libri. Përdoret një titull, një paragraf me përshkrim, një imazh të kopertinës, një tabelë me të dhëna teknike për librin dhe një lidhje për blerje, e cila hapet në një skedë të re.

<h1>Libri: Bota Digjitale</h1>

<p title="Përshkrim i shkurtër i përmbajtjes">Ky është një libër emocionues për fillimet e teknologjisë kompjuterike dhe zhvillimin e internetit.</p>

 width="250">

<h3>Detaje teknike</h3>

<table border="1">

<tr>

<th>Karakteristika</th>

<th>Informacioni</th>

</tr>

<tr>

<td>Autori</td>

<td>Marko Petrovski</td>

</tr>

<tr>

<td>Gjuha</td>

<td>Maqedonisht</td>

</tr>

<tr>

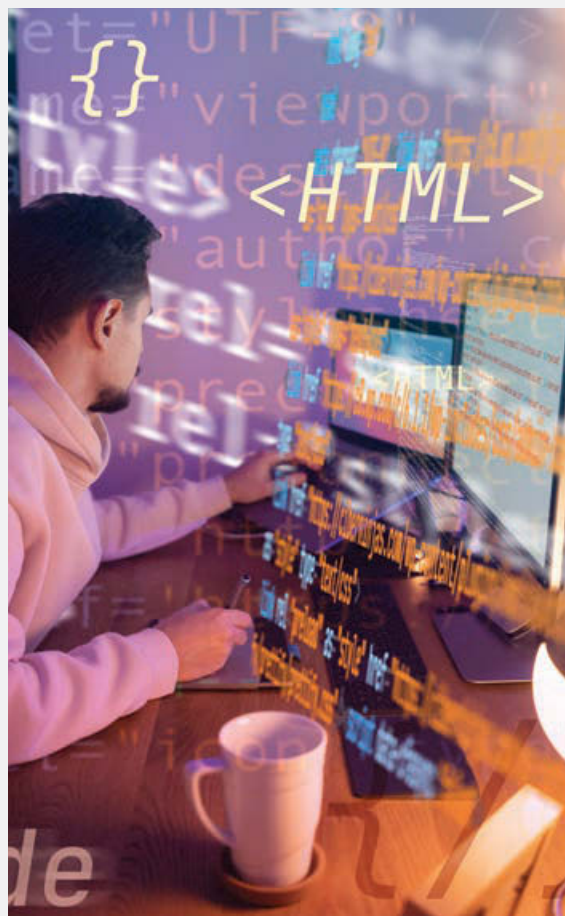
<td>Numri i faqeve</td>

<td>240 faqe</td>

</tr>

</table>

Kliko këtu për ta porositur librin



PYETJE PËR KONTROLLIN E NJOHURIVE:

1. Ku shkruhen GJITHMONË atributet HTML?

- A) Në fund të dokumentit, pas etiketës </html>
- B) Brenda etiketës fillestare (opening tag)
- C) Brenda etiketës përmbyllëse (closing tag)
- D) Midis etiketës fillestare dhe përmbyllëse, së bashku me tekstin

2. Cila nga etiketat e mëposhtme të tabelës përdoret për krijimin e një rreshti të ri horizontal?

- A) <table>
- B) <td>
- C) <tr>
- D) <th>

3. Me fjalët e tua, shpjego cili është dallimi midis etiketës HTML (tag) dhe atributit HTML (attribute). Si bashkëpunojnë ato për të shfaqur diçka në ekran?

4. Nëse një imazhi i vendosim atributin `width="300"`, ndërsa e lëmë pa atributin `height`, çfarë do të ndodhë me lartësinë e imazhit në shfletues?

- A) Imazhi nuk do të shfaqet fare, sepse mungon lartësia.
- B) Lartësia do të bëhet 0 pikselë dhe imazhi do të jetë i padukshëm.
- C) Shfletuesi do ta përshtatë vetë lartësinë në mënyrë proporcionale, pa e deformuar imazhin.
- D) Imazhi do të shtrihet vertikalisht në të gjithë ekranin.

5. Dëshiron të vendosësh një imazh (logotip.png) i cili, kur përdoruesi të klikojë mbi të, do ta çojë në faqen <https://www.google.com> në një skedë krejtësisht të re. Shkruaj kodin e saktë HTML për këtë qëllim.

6. Cili nga kodet e mëposhtme shfaq saktë një lidhje me një përshkrim alternativ kur kalon miun mbi të?

- A) `<a src="https://sajtedu.mk" alt="Shkollë">Shko te faqja</a>`
- B) `<a href="https://sajtedu.mk" title="Vizito shkollën tonë">Shko te faqja</a>`
- C) `<a target="https://sajtedu.mk">Shko te faqja</a>`
- D) `<link href="https://sajtedu.mk" title="Shkollë">Shko te faqja</link>`

7. Shiko kodin e mëposhtëm HTML për një tabelë:

HTML

```
<table border="1">
  <tr>
    <td>Maqedonisht</td>
    <td colspan="2">Matematikë (orë e dyfishtë)</td>
  </tr>
  <tr>
    <td>Anglisht</td>
    <td>IT</td>
    <td>Art figurativ</td>
  </tr>
</table>
```

Analizo kodin dhe shpjego pse kjo tabelë do të deformohet vizualisht në shfletues. Ku qëndron gabimi në numërimin e qelizave?

8. Imagjino se po krijon një faqe interneti për persona me dëmtim të shikimit, të cilët përdorin softuer „lexues ekrani“ (Screen Readers). Cili nga hapat e mëposhtëm është thelbësor për ta dhe pse?

- A) Të përdoret gjithmonë atributi `target="_blank"` në lidhje.
- B) Të shkruhet gjithmonë një tekst i saktë dhe përshkruhet në atributin `alt` të etiketës `<img>`.
- C) Tabelat të kenë gjithmonë `border="1"`.
- D) Të gjithë titujt të jenë vetëm `<h1>`.

9. Dizajno kod HTML për një kartëvizitë online (profil) sa më të thjeshtë. Kodi duhet të përmbajë: titullin tënd të preferuar, një paragraf për veten, një imazh tëndin dhe një lidhje drejt faqes tënde të preferuar. Kujdesu për hapjen dhe mbylljen e saktë të etiketave dhe sintaksën e saktë të attributeve.

10. Imagjino se duhet të krijosh orarin për ditën e nesërme në shkollë, ku ora e tretë dhe e katërt janë orë të dyfishta (bllok) të së njëjtës lëndë. Shkruaj vetëm kodin HTML për atë rresht (`<tr>`) të tabelës, duke përdorur atributin përkatës për bashkimin e qelizave.



Etiketa `<a>` me atributin href nuk shërben vetëm për navigim nga një faqe interneti në një tjetër. Ajo mund të komunikojë edhe me sistemin operativ të kompjuterit ose telefonit të përdoruesit.

Lidhje për dërgim të drejtpërdrejtë të e-mailit (mailto:): Nëse dëshiron që, kur dikush të klikojë në lidhje, t'i hapet menjëherë programi i e-mailit (si Outlook ose Gmail) me adresën tashmë të vendosur, përdor këtë:

```
<a href="mailto:kontakt@skolo.mk">Dërgo e-mail</a>
```

Lidhje për telefonatë të drejtpërdrejtë (tel:): Kjo është shumë praktike për përdoruesit që e shohin faqen nga telefoni. Kur klikojnë lidhjen, telefoni automatikisht fillon të formojë numrin:

```
<a href="tel:+3892123456">Telefono tani</a>
```

Nëse ke një faqe interneti shumë të gjatë dhe dëshiron që me një klikim përdoruesi të „kërcejë“ direkt te seksioni i kontaktit në fund të faqes, përdor kombinimin e href dhe atributit id:

```
<a href="#seksion-kontakti">Kalo te kontaktet</a>
```

```
<h2 id="seksion-kontakti">Këtu gjenden të dhënat tona</h2>
```

Eksperimento edhe me imazhe inteligjente duke përdorur atributin loading="lazy", me etiketa për tekst të avancuar që kompjuteri mund ta „kuptojë“, atributin download dhe të tjera.

5.2.6 CSS – STILET NË FAQET E INTERNETIT

Ndërsa HTML është përgjegjës për strukturën (skeleton) e faqes së internetit, CSS (Cascading Style Sheets) është gjuha përgjegjëse për pamjen, dizajnin dhe estetikën e saj. Pa CSS, të gjitha faqet e internetit do të dukeshin si tekst i zakonshëm bardh e zi në Notepad, me lidhje blu.

CSS përshkruan se si duhet të shfaqen etiketat HTML në ekran. Ekzistojnë tri mënyra për të shtuar CSS në dokumentin tonë HTML:

- **Inline (në rresht):** Vendoset drejtpërdrejt brenda etiketës HTML përmes atributit style. (Shembull: `<p style="color: red;">`).
- **Internal (i brendshëm):** Vendoset brenda vetë dokumentit HTML, në etiketën speciale `<style>`, e cila vendoset në pjesën `<head>`.
- **External (i jashtëm)** – mënyra profesionale: Kodi shkruhet në një skedar të veçantë me prapashtesën .css (për shembull: style.css), i cili më pas lidhet me HTML përmes etiketës `<link>`.

Për të ngjyrosur ose ndryshuar një element, fillimisht duhet t'i tregojmë kompjuterit cilin element duam të ndryshojmë. Këtë e bëjmë me ndihmën e **selektorëve**.

Selektorët bazë:

- **Selektori sipas etiketës (Element selector):** Përzgjedh të gjitha etiketat e të njëjtit lloj në faqe.

```
p {
  color: blue; /* I bën të gjithë paragrafët blu */
}
```

- **Selektori sipas klasës (Class selector):** Përzgjedh një grup elementesh që kanë atributin class. Gjithmonë fillon me **pikë (.)**.

```
.tekst-i-kuq {
  color: red;
}
```

- **Selektori sipas identifikuesit (ID selector):** Përzgjedh vetëm një element unik me atributin id. Gjithmonë fillon me **simbolin hash (#)**.

```
#titull-kryesor {
  font-size: 30px;
}
```

Në CSS, ngjyra e tekstit ndryshohet me komandën color. Ngjyrat mund t'i përcaktojmë në tri mënyra:

1. Me **emrin e ngjyrës**: red, blue, green, orange.
2. Me **kodin HEX** (me simbolin #): #ff0000 (e kuqe), #ffffff (e bardhë), #000000 (e zezë).
3. Me **vlera RGB**: rgb(255, 0, 0).

```
h2 {
  color: #2c3e50; /* Nuancë e kaltër e errët */
}
```

Formatimi i tekstit është thelbësor për një pamje të bukur të faqes së internetit. Komandat më të rëndësishme janë:

- **font-family:** Zgjedhja e stilit të shkronjave (p.sh. Arial, Verdana, Times New Roman). Rekomandohet që gjithmonë të vendoset një familje gjenerike, si sans-serif (pa zbukurime) ose serif (me zbukurime), si alternativë nëse kompjuteri nuk e ka fontin e caktuar.
- **font-size:** Madhësia e shkronjave, zakonisht e shprehur në pikselë (px).
- **font-weight:** Trashësia e tekstit (bold për tekst të trashë, normal për tekst të zakonshëm).

```
body {
  font-family: Arial, sans-serif;
  font-size: 16px;
}
```

Sfondi i të gjithë faqes (etiketa <body>) ose i një elementi të caktuar (si tabela ose paragrafi) rregullohet me:

- **background-color:** Ndryshon ngjyrën e sfondit.
- **background-image:** Vendors një imazh si sfond përmes lidhjes (url('slika.jpg')).

```
body {
  background-color: #f4f4f4; /* Sfond i butë gri i çelët */
}
```

Çdo element në HTML (titull, imazh, paragraf) kompjuteri e konsideron si një **kuti drejtkëndore të padukshme**. Hapësirën dhe renditjen rreth këtyre kutive i kontrollojmë me tri komanda kryesore:

- **padding (hapësira e brendshme):** Hapësira **brenda** kutisë, midis tekstit dhe kornizës së saj.
- **border (Korniza):** Vija rreth elementit (mund të ketë trashësi, stil dhe ngjyrë, p.sh. 2px solid black).
- **margin (hapësira e jashtme):** Hapësira jashtë kutisë, e cila krijon distancë midis këtij elementi dhe elementeve të tjera përreth tij.

```
.kuti {
  border: 1px solid #ccc;
  padding: 20px; /* Teksti brenda do të jetë i ndarë nga korniza */
  margin: 15px; /* Kutia do të largohet nga elementet e tjera */
}
```

Në shembullin e mëposhtëm paraqitet mënyra se si përdoren komandat për ndryshimin e ngjyrës së shkronjave dhe sfondit.

```
<!DOCTYPE html>
<html lang="mk">
<head>
  <meta charset="UTF-8">
  <title>Shembull i thjeshtë CSS</title>
</head>
<body style="background-color: #f0f0f0; font-family: Arial;">

  <h1 style="color: blue;">Titulli im i parë i stilizuar</h1>

  <p style="background-color: red; color: white; padding: 10px;">
    Ky tekst ka sfond të kuq dhe shkronja të bardha!
  </p>

</body>
</html>
```

Cila nga komandat përcakton gjuhën në të cilën do të shfaqet teksti? Në çfarë mënyre është shtuar CSS? Kur dëshirojmë të stilizojmë elemente konkrete (specifike) të faqes, është më mirë, në vend që t'i stilizojmë të gjitha njëherësh, të përcaktojmë secilin veçmas. Për këtë qëllim përdoren dy lloje të ndryshme „shënuesish“ (selektorësh): **Klasa (Class)** dhe **Identifikuesi (ID)**.

```
<!DOCTYPE html>
<html lang="mk">
<head>
  <meta charset="UTF-8">
  <title>Selektorët sipas Klasës dhe ID-së</title>
<style>
  /* Selektor për KLASË (fillon me pikë) */
  .njoftim-i-rendesishem {
    background-color: yellow;
    padding: 8px;
    border: 1px solid orange;
  }

  /* Selektori për ID (fillon me simbolin #) */
```

```
#titull-kryesor {
  color: crimson;
  text-align: center;
}
</style>
</head>
<body>
```

```
<h1 id="titull-kryesor">Mirë se vini</h1>
```

```
<p class="njoftim-i-rendesishem">Kujdes: Nesër nuk ka mësim në IT.</p>
```

```
<p>Ky është tekst i zakonshëm që nuk do të ngjyrosset me të verdhë.</p>
```

```
<p class="njoftim-i-rendesishem">Kujdes: Projektet duhet të dorëzohen deri të premten.</p>
```

```
</body>
```

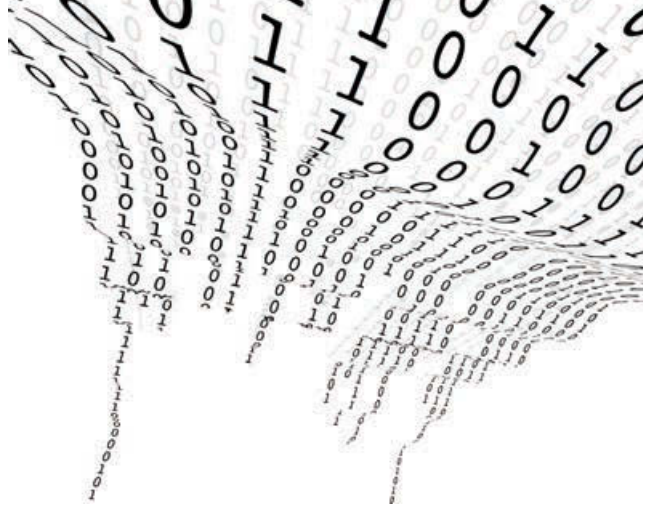
```
</html>
```



Figura 13: Stilizimi me klasë dhe identifikues



PYETJE PËR KONTROLLIMIN E NJOHURIVE:



1. Çfarë nënkupton shkurtesa CSS?

- A) Computer Style Sheets
- B) Creative Style Systems
- C) Cascading Style Sheets
- D) Colorful Style Software

2. Me cilin simbol fillon selektori për KLASËN (class) në CSS?

- A) Me simbolin diez (#)
- B) Me pikë (.)
- C) Me pikëçuditje (!)
- D) Me pikëpyetje (?)

3. Shpjego dallimin në veprimin ndërmjet selektorit p dhe selektorit #glaven-paragraf. Cilin element të faqes do të ndryshojë secili prej tyre?

4. Nëse një elementi i cakton background-color: black; dhe qëllimisht harron deklaratën color, pse teksti në atë element do të bëhet i padukshëm për përdoruesin?

5. Shkruaj kodin e saktë CSS me të cilin të gjithë titujt e tipit <h2> në faqen e internetit do të kenë ngjyrë të gjelbër të tekstit, madhësi 24 pikselë dhe fontin Arial.

6. Cili nga kodet e mëposhtme CSS vendos në mënyrë të saktë një kornizë të zezë me trashësi 2 pikselë rreth një imazhi?

- A) `img { margin: 2px black; }`
- B) `img { border: 2px solid black; }`
- C) `img { padding: 2px solid; }`
- D) `img { line: 2px black; }`

7. Shqyrto situatën e mëposhtme: Ke dy kuti, njëra mbi tjetrën. Dëshiron të krijosh një hapësirë më të madhe (hapësirë të zbrazët) ndërmjet tyre. Cila deklaratë nga „Box Model“ duhet të përdorësh dhe pse deklarata padding nuk do të të ndihmojë në këtë rast?

8. Shoku yt i klasës e ka shkruar të gjithë kodin CSS duke përdorur stile inline (drejtpërdrejt në çdo etiketë HTML me `style="..."`). Projekti ka 5 faqe të ndryshme ueb. Vlerësoje këtë qasje: Çfarë do të ndodhë nëse profesori kërkon që nesër të ndryshojë sfondin në të gjitha 5 faqet, nga gri në të bardhë? A do të ishte skedari i jashtëm CSS një zgjedhje më e mirë?

9. Dizajno një stil të plotë CSS për klasën e quajtur .karta-produkt. Kutia duhet të ketë sfond ngjyrë gri të çelët, hapësirë të brendshme (padding) prej 15 pikselësh, një kornizë të hollë dhe teksti brenda saj të jetë me fontin Verdana.

10. Shkruaj një kod të shkurtër HTML dhe CSS ku do të krijosh një titull kryesor me identifikuesin unik `id="naslov-akcija"`. Në pjesën CSS, bëje këtë titull me shkronja të theksuara (bold) dhe me sfond të verdhë (background-color).

5.2.7. RENDITJA DHE DIZAJNI ADAPTIV (RESPONSIVE)

Kur krijojmë një faqe interneti, nuk mjafton vetëm t'u japim ngjyra elementeve. Është thelbësore të dimë se si t'i renditim ato në ekran (majtas, djathtas, njëri pranë tjetrit), si dhe si ta përshtatim automatikisht këtë renditje kur faqja shikohet në kompjuter, tablet ose telefon celular. Kjo arrihet përmes një renditjeje të përshtatshme dhe dizajnit adaptiv.

Renditja përfaqëson strukturën ose „skeletin“ e faqes së internetit. Ajo përcakton se ku do të vendoset logoja, ku menuja, ku teksti kryesor dhe ku shiriti anësor (sidebar).

Në CSS, elementet zakonisht vendosen njëri nën tjetrin (elemente bllok, si <div>, <h1>, <p>). Për të krijuar një renditje të ndryshme, përdorim deklaratën **display**. Një nga mënyrat më moderne dhe më të thjeshta për renditjen e elementeve është **Flexbox** (Flexible Box). Kur një kutie prind i caktojme deklaratën `display: flex;`, të gjithë elementet e saj të brendshme vendosen automatikisht **njëri pranë tjetrit (në një rresht)**, në vend që të vendosen njëri nën tjetrin.

```
.kontejner {
    display: flex;

    justify-content: space-between; /* I largon elementet në mënyrë të barabartë, majtas dhe djathtas */
}
```

Dizajni ueb i përshtatshëm (Responsive Web Design) është një qasje në krijimin e faqeve të internetit, e cila mundëson që pamja dhe përmbajtja e tyre të përshtaten automatikisht me madhësinë e ekranit të pajisjes nga e cila aksesohet faqja – pavarësisht nëse bëhet fjalë për telefon inteligjent, tablet, laptop apo monitor të madh desktop.

Në vend që të krijohen versione të veçanta të faqes për telefonat celularë dhe kompjuterët, me dizajnin responsive përdoret i njëjti kod, i cili riorganizohet në mënyrë fleksibile dhe zvogëlohet ose zgjerohet në varësi të hapësirës së disponueshme.

Parimet kryesore të dizajnit responsive:

1. **Fleksibiliteti i paraqitjes (Fluid Layouts):** Në vend që të përdorim dimensione fikse në pikselë (p.sh. `width: 900px;`), përdorim përqindje (`width: 100%;` ose `width: 50%;`). Në këtë mënyrë, elementet zgjerohen ose ngushtohen në varësi të madhësisë së ekranit.
2. **Imazhet fleksibile:** Që imazhi të mos dalë kurrë jashtë ekranit në telefon, në CSS i vendosim rregullin e mëposhtëm themelor:

```
img {
    max-width: 100%;
    height: auto;
}
```

Pse është i rëndësishëm për faqet moderne të internetit?

- **Dominimi i trafikut mobil:** Më shumë se gjysma e trafikut global të internetit vjen nga pajisjet mobile. Nëse faqja juaj duket keq ose është e vështirë për t'u naviguar në telefon, humbni një numër të madh vizitorësh potencialë.
- **Përvojë më e mirë e përdoruesit (UX):** Përdoruesit nuk duan të zmadhojnë vazhdimisht faqen, të lëvizin horizontalisht ose të shtypin butona të vegjël. Dizajni i përshtatshëm ofron një pamje të qartë, tekst lehtësisht të lexueshëm dhe navigim të thjeshtë në çdo pajisje.

- Rëndësia për SEO-n (optimizimin për motorët e kërkimit): Motorët e kërkimit, si Google, përdorin “mobile-first indexing”. Kjo do të thotë se Google vlerëson kryesisht versionin mobil të faqes suaj kur përcakton pozicionin e saj në rezultatet e kërkimit.
- Mirëmbajtje më e lehtë dhe kosto më e ulët: Në vend që të shpenzosh kohë dhe para për zhvillimin dhe përditësimin e dy faqeve të ndryshme të internetit (një për kompjuter dhe një për pajisjet mobile), mirëmban vetëm një sistem që funksionon në të gjitha pajisjet.

Në ditët e sotme, dizajni ueb i përshtatshëm (Responsive Design) nuk është më vetëm një „shtesë“ ose luks, por një standard bazë për çdo faqe interneti profesionale që synon të jetë e suksesshme dhe e dukshme në internet.

Mjeti më i rëndësishëm në CSS për përshtatjen e faqes së internetit quhet **Media Queries** (kërkesa mediatike). Ato funksionojnë si filtra ose kushte: „Nëse ekrani është më i vogël se 600 pikselë, atëherë zbato këto rregulla të reja CSS“. Imagjino se në kompjuter ke tre elemente – kuti të vendosura njëra pranë tjetrës (flex). Në një pajisje mobile, ekrani është më i ngushtë dhe këto elemente – kuti do të „ngjeshen“. Me ndihmën e Media Query, u tregojmë që në pajisjen mobile të renditen **njëra nën tjetrën**.

```
/* CSS standard për kompjuter (elementet vendosen njëri pranë tjetrit) */
```

```
.menu-kryesore {
display: flex;
flex-direction: row;
}
```

```
/* Rregull i veçantë për telefonat celularë (ekrane më të ngushta se 600px) */
```

```
@media (max-width: 600px) {
.menu-kryesore {
flex-direction: column; /* I vendos elementet njëri nën tjetrin */
}
}
```

Çdo element/kuti në HTML ka të ashtuquajturin Box Model (modeli i kutisë). Edhe nëse nuk e sheh kornizën, kompjuteri për çdo element llogarit katër pjesë:

- **Përmbajtja (content):** Vetë teksti ose imazhi.
- **Hapësira e brendshme (padding):** Hapësira e zbrazët rreth tekstit, në mënyrë që ai të mos jetë i ngjitur me skajet.
- **Korniza (border):** Vija rreth elementit (mund të jetë e padukshme, por mund edhe ta ngjyrosim).
- **Hapësira e jashtme (margin):** Hapësira që e largon kutinë nga kutitë fqinje, në mënyrë që të mos preken me njëra-tjetrën.

Kur dizajnojmë një faqe interneti, në fakt luajmë me këto kuti – i zgjerojmë, i ngushtojmë, ua ndryshojmë ngjyrën e sfondit dhe përcaktojmë nëse do të qëndrojnë në të njëjtin rresht apo do të vendosen njëra nën tjetrën!

Në shembullin më poshtë është paraqitur dizajni adaptiv i një faqeje interneti. Përdore kodin në editorin VS Code dhe testo madhësinë e dritares duke e zvogëluar dhe zmadhuar atë, për të vërejtur se si funksionojnë display: flex; dhe @media.

```

<!DOCTYPE html>
<html lang="mk">
<head>
  <meta charset="UTF-8">
  <title>Dizajn i shkurtër adaptiv</title>
  <style>
    body {
      font-family: Arial, sans-serif;
      background-color: #f4f4f4;
      padding: 20px;
    }

    /* KONTEJNER: Në kompjuter, kartelat renditen horizontalisht (në një rresht) */
    .grupi-kutive {
      display: flex;
      flex-direction: row;
      gap: 15px; /* Hapësira ndërmjet kutive */
    }

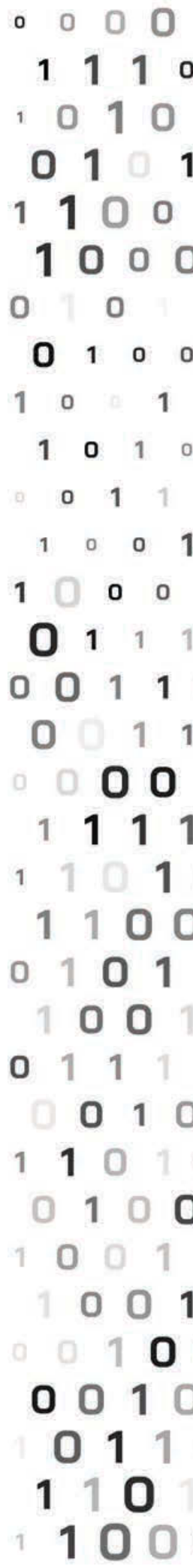
    /* KARTELA: Stil për secilën kuti individuale */
    .karta {
      background-color: white;
      padding: 20px;
      border: 1px solid #ddd;
      flex: 1; /* U tregon kutive që ta ndajnë gjerësinë në mënyrë të barabartë */
    }

    /* KËRKESË MEDIATIKE: Për ekranet më të vogla se 600 pikselë (pajisjet mobile) */
    @media (max-width: 600px) {
      .grupi-kutive {
        flex-direction: column; /* I vendos kutitë njëra nën tjetrën. */
      }
    }
  </style>
</head>
<body>

  <h2>Renditja jonë adaptuese</h2>

  <div class="grupi-kutive">
    <div class="karta">
      <h3>Kutia 1</h3>

```



```

    <p>Kjo është përmbajtja e kutisë së parë.</p>
  </div>
  <div class="karta">
    <h3>Kutia 2</h3>
    <p>Kjo është përmbajtja e kutisë së dytë.</p>
  </div>
  <div class="karta">
    <h3>Kutia 3</h3>
    <p>Kjo është përmbajtja e kutisë së tretë.</p>
  </div>
</div>
</body>
</html>

```



PYETJE PËR KONTROLLIMIN E NJOHURIVE:

1. Cila deklaratë CSS përdoret për të aktivizuar modelin fleksibil të renditjes (Flexbox)?
 - A) display: block;
 - B) display: flex;
 - C) position: static;
 - D) float: left;
2. Çfarë arrihet me përdorimin e rregullit CSS @media (Media Queries)?
 - A) Shtohen skedarë video dhe audio në faqe.
 - B) Zbatohen stile të ndryshme CSS në varësi të madhësisë së ekranit.
 - C) Ndryshohet gjuha e faqes së internetit.
 - D) Faqja lidhet me rrjetet sociale.
3. Pse është më mirë të përdorim përqindje (%) në vend të pikselëve fiks (px) për gjerësinë e kontejnerëve kryesorë kur dizajnojmë një faqe interneti responsive?
4. Shpjego ndryshimin në sjelljen e elementeve kur flex-direction është vendosur në row, në krahasim me rastin kur është vendosur në column.
5. Shkruaj rregullin CSS që do të mundësojë që të gjitha imazhet () në faqe të jenë responsive (të mos tejkalojnë gjerësinë e elementit prind dhe të ruajnë raportin e përmasave).
6. Cili nga kodet e mëposhtme i targeton në mënyrë të saktë pajisjet, gjerësia maksimale e ekranit të të cilave është 768 pikselë (si tabletët)?
 - A) @media (width: 768px)
 - B) @media (max-width: 768px)
 - C) @screen (max-device: 768px)
 - D) @media (min-width: 768px)

7. Një nxënës ka krijuar një renditje ku menuja e navigimit dhe përmbajtja kryesore qëndrojnë njëra pranë tjetrës në kompjuter. Kur faja hapet në pajisje mobile, përmbajtja „ngjeshet“ dhe teksti nuk mund të lexohet. Analizoje këtë problem dhe trego se cili parim i dizajnit responsive është shkelur dhe si duhet të ndryshohet renditja në pajisjen mobile.

8. Imagjino se duhet të zgjedhësh një strategji për zhvillimin e një ueb-faqeje të re për shkollën. Njëra mundësi është të krijosh dy versione të veçanta (një të plotë për kompjuter dhe një të veçantë për pajisjet mobile, me lidhje të ndryshme), ndërsa mundësia e dytë është të krijosh një faqe responsive me Media Queries. Vlerëso efektin e të dyja mundësive nga aspekti i mirëmbajtjes dhe ndryshimit të informacioneve në të ardhmen.

9. Krijo një kod të thjeshtë CSS për klasën .kuti-bloq, e cila në kompjuter do të zërë saktësisht 33.3% të gjerësisë së ekranit, ndërsa me ndihmën e Media Query, në pajisjet mobile (nën 600 pikselë), gjerësia e saj automatikisht të bëhet 100%.

10. Dizajno strukturën HTML dhe kodin CSS për një kokë të thjeshtë të faqes (Header), e cila përmban titullin (majtas) dhe lidhjet (djathtas). Në kompjuter, ato duhet të qëndrojnë në të njëjtin rresht (njëra pranë tjetrës) me ndihmën e Flexbox, ndërsa kur ekranin ngushtohet nën 480 pikselë, titulli dhe lidhjet duhet të qëndrëzohen dhe të renditen njëri nën tjetrin.



PËR ATA QË DËSHIROJNË TË DINË MË

Kur krijojmë një dizajn responsive, shpeshherë përqindjet (%) nuk janë të mjaftueshme, sepse ato varen nga madhësia e „kutisë prind“ (parent). Prandaj, profesionistët përdorin njësi matëse që lidhen drejtpërdrejt me madhësinë e të gjithë ekranit (Viewport):

vw (Viewport Width): 1vw është e barabartë me 1% të gjerësisë së ekranit. Nëse vendosim width: 50vw, elementi do të zërë gjithmonë saktësisht gjysmën e ekranit, pavarësisht se në çfarë pajisjeje hapet.

vh (Viewport Height): 1vh është e barabartë me 1% të lartësisë së ekranit. Kjo është një mënyrë e shkëlqyer për krijimin e seksioneve fillestare („Hero“), të cilat duhet të mbushin të gjithë ekranin në lartësi kur ngarkohet faja:

```
.seksion-fillestar {
    height: 100vh; /* Zë të gjithë lartësinë e ekranit */
}
```

Një nga sfidat më të mëdha në dizajnin responsive është madhësia e fontit. Nëse titulli është shumë i madh në kompjuter, në pajisjen mobile ai mund të „thyeret“ dhe të shfaqet në tre rreshta. Në vend që të shkruash dhjetëra Media Queries vetëm për madhësinë e fontit, në CSS-në moderne ekziston funksioni clamp().

Ai pranon tri vlera: madhësinë minimale, madhësinë e rekomanduar (fluide) dhe madhësinë maksimale.

```
h1 {
    font-size: clamp(20px, 5vw, 45px);
}
```

Ndërsa Flexbox është i shkëlqyer për radhitjen e elementeve në një drejtim (ose vetëm në rresht, ose vetëm në kolonë), CSS Grid është krijuar për një organizim të plotë dydimensional (si të rreshtave, ashtu edhe të kolonave njëkohësisht). Me Grid mund të krijosh një strukturë të ngjashme me paraqitjen e një gazete ose reviste të vërtetë.

```
.renditje-reviste {  
    display: grid;  
    grid-template-columns: repeat(3, 1fr); /* Krijon saktësisht 3 kolona të  
    barabarta */  
    grid-gap: 20px;  
}
```

Ndërsa në pajisjet mobile, me vetëm një Media Query, thjesht ndryshon numrin e kolonave:

```
@media (max-width: 600px) {  
    .renditje-reviste {  
        grid-template-columns: 1fr; /* Në pajisjet mobile, gjithçka shndërrohet  
        në 1 kolonë */  
    }  
}
```

Ndonjëherë, një tabelë e madhe ose një imazh kompleks thjesht nuk ka kuptim të shfaqet në telefonat mobilë, sepse vetëm do ta ngadalësonte internetin e përdoruesit dhe do ta „ngjeshte“ hapësirën. Me Media Queries, mund ta fshehësh plotësisht një element në pajisjet mobile, ndërsa ta lësh të dukshëm në kompjuter:

```
/* Në pajisjet mobile, fshih shiritin anësor me reklama */  
@media (max-width: 600px) {  
    .reklame-anesore {  
        display: none;  
    }  
}
```

5.2.8. INTELIGJENCA ARTIFICIALE GJENERATIVE PËR ZHVILLIMIN E WEB-IT

Teknologjia e inteligjencës artificiale gjenerative (Generative AI) ka sjellë një revolucion në mënyrën se si krijohen faqet web. Mjetet si ChatGPT, Claude, Microsoft Copilot dhe shtojcat e specializuara për VS Code (si GitHub Copilot) tani mund të shkruajnë kod kompleks HTML dhe CSS brenda pak sekondash.

Kjo përmbajtje do të të ndihmojë të kuptosh se si funksionon ky proces, si t'i japësh udhëzime të sakta kompjuterit dhe si ta analizosh në mënyrë kritike kodin që inteligjenca artificiale krijon për ty.

1. Kuptimi i udhëzimit (prompt-it) dhe formulimi i udhëzimeve të qarta

Një nga konceptet më të rëndësishme gjatë punës me inteligjencën artificiale është **prompt-i (udhëzimi)**. Prompt-i është një tekst me udhëzime ose kërkesa që përdoruesi ia jep një mjeti të inteligjencës artificiale për të marrë një rezultat specifik (në rastin tonë, kod web).

Nëse i thua inteligjencës artificiale: „Krijë një faqe web për një shkollë“, rezultati do të jetë shumë i përgjithshëm dhe i thjeshtë. Për të marrë kod profesional, duhet të përdorësh një prompt të strukturuar dhe të detajuar.

Si të formulosh një prompt të qartë dhe të saktë për zhvillimin web?

Një prompt cilësor duhet t'i përmbajë elementet e mëposhtme:

- **Roli (Role):** Përcakto rolin e inteligjencës artificiale (p.sh. „Ti je një dizajner web me përvojë“).
- **Konteksti/Qëllimi:** Shpjego saktësisht se cili është qëllimi i faqes.
- **Kërkesat teknike:** Përcakto cilat gjuhë dhe teknologji duhet të përdoren (HTML, CSS me stilim të brendshëm/të jashtëm, paraqitje me Flexbox).
- **Detaje specifike:** Përcakto ngjyrat, numrin e seksioneve dhe përshatshmërinë për pajisjet mobile.

Shembull i një prompt-i të keq: „Më shkruaj HTML për një faqe web kontakti.“

Shembull i një prompt-i të shkëlqyer: „Ti je një programues profesionist web. Më shkruaj kod të plotë HTML dhe CSS të brendshëm për një faqe web responsive «Kontakt». Faqja duhet të përmbajë një formular me fushat Emri, E-maili dhe Mesazhi, si dhe një buton për dërgim. Përdor një dizajn modern me sfond blu të errët (#2c3e50) dhe tekst të bardhë. Përdor Flexbox për ta vendosur formularin në qendër të ekranit. Në pajisjet mobile (nën 600px), formulari duhet të zërë 95% të gjerësisë.“

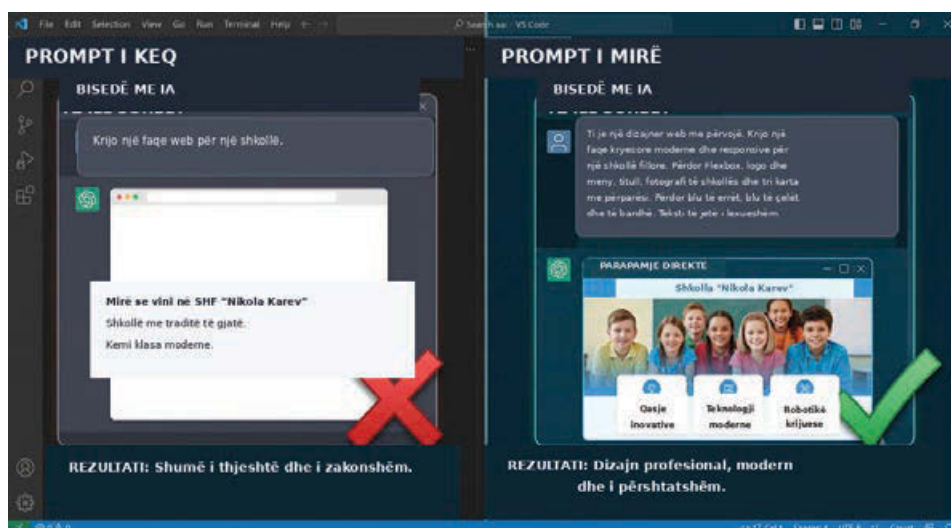


Figura 14: Shembull i një prompt-i të përgjithshëm dhe specifik

2. Krijimi automatik i faqeve web dhe analiza e rezultateve

Procesi i punës me një asistent të inteligjencës artificiale (IA) gjatë krijimit të faqeve web zhvillohet në disa hapa kyç. Për të marrë kod cilësor, procesi nuk është thjesht „kopjo-ngjit“, por përfshin planifikim, dhënie të udhëzimeve dhe kontrollin.

Ja i gjithë procesi hap pas hapi:

- Planifikimi i strukturës (para se të fillosh të shkruash kodin)

Para se ta hapësh mjetin e inteligjencës artificiale, duhet të kesh një ide të qartë se çfarë dëshiron të krijosh. Skicoje strukturën (layout-in): ku dëshiron të vendoset menyuja kryesore, sa elemente ose kuti do të përmbajë faqja dhe cilat ngjyra do të përdorësh.

- Formulimi i një prompt-i të qartë (udhëzimi)

Ky është hapi më i rëndësishëm. Në vend të fjalive të shkurtra dhe të paqarta, asistentit të inteligjencës artificiale duhet t'i japësh një përshkrim të detajuar.

- **Vendos rolin:** „Ti je një programues web me përvojë.“
- **Përcakto qëllimin:** „Më krijo një faqe web për...“
- **Përcakto kërkesat teknike:** „Përdor HTML5 dhe CSS të brendshëm brenda etiketës <style>. Krijoje paraqitjen me Flexbox. Kodi duhet të jetë responsive për pajisjet mobile, duke përdorur Media Queries.“

- Gjenerimi dhe marrja e kodit

Asistenti i inteligjencës artificiale do ta shkruajë kodin brenda pak sekondash. Detyra jote është:

1. Ta shqyrtosh kodin menjëherë në dritaren e bisedës.
2. Të kopjosh bllokun përkatës të kodit.
3. Ta transferosh në skedarin tënd të ri (për shembull, index.html) në VS Code dhe ta ruash.

- Analiza dhe testimi në shfletues (Kontrollimi i rezultatit)

Pasi ta ruash skedarin, hape atë në një shfletues (Chrome, Edge). Tani bëj një analizë kritike të punës së realizuar:

- **Pamja:** A janë ngjyrat dhe fontet ashtu siç i ke kërkuar?
- **Struktura:** A janë etiketat HTML të vendosura në mënyrë të saktë dhe logjike (titujt, paragrafët, kutitë)?
- **Përshtatshmëria (Responsive):** Zvogëloje dritaren për të parë nëse elementet kalojnë siç duhet njëri poshtë tjetrit në një ekran më të vogël.

- Iteracioni dhe përpunimi manual (Korrigjimi i gabimeve)

Pothuajse kurrë përgjigjja e parë nga IA nuk është 100% e përsosur. Këtu hyn në vepër ti si programues:

- **Nëse ka një gabim të vogël:** Korrigjoje vetë, manualisht, në VS Code (për shembull, ndrysho ngjyrën nga e kuqe në të kaltër në pjesën CSS). Kjo është mënyra më e mirë për të mësuar.

- **Nëse ka një gabim më të madh ose mungon një seksion i tërë:** Kthehu te asistenti i inteligjencës artificiale dhe jepi një prompt shtesë, për shembull: „Kodi është i mirë, por shto edhe një buton poshtë tekstit, i cili të jetë me ngjyrë të gjelbër.“

Me këtë proces ciklik, inteligjenca artificiale të ndihmon ta përfundosh shpejt punën e vështirë dhe mekanike, ndërsa ti ruan kontrollin e plotë mbi logjikën dhe pamjen përfundimtare të faqes web.

Për t'u bërë një përdorues i mirë i inteligjencës artificiale, duhet t'i njohësh përparësitë dhe mangësitë e të dyja qasjeve. Ja një krahasim i drejtpërdrejtë:

Karakteristika	Kod i shkruar manualisht (Hand-coded)	Kod i gjeneruar nga IA (AI-generated)
Shpejtësia e zhvillimit	I ngadalshëm (kërkon shkrimin manual të çdo rreshti dhe kllape).	Ekstremisht shpejt (e gjithë faqja përfundohet brenda pak sekondash).
Kontrolli dhe kuptimi	Maksimale. E di saktësisht pse ndodhet çdo rresht i kodit dhe mund ta korrigjosh lehtësisht nëse ka ndonjë gabim.	E kufizuar. Nëse kodi është shumë i gjatë, është e vështirë të kuptosh se ku ka lindur problemi nëse diçka prishet.
Optimizimi dhe logjika	Kodi është i pastër, i shkruar saktësisht sipas nevojave të tua, pa përsëritje të panevojshme.	Mund të përmbajë „kod të panevojshëm“ (udhëzime të panevojshme që IA i ka shtuar pa ndonjë arsye reale).
Mësimi i materialit	Mënyra më e mirë për të mësuar. Përparimi vjen duke bërë dhe korrigjuar gabimet e tua.	Nuk ndihmon në mësim nëse thjesht kopjon dhe ngjit verbërisht, pa e lexuar dhe kuptuar kodin.

Inteligjenca artificiale është **një asistent i shkëlqyer, por një udhëheqës i keq**. Mënyra më e mirë për ta përdorur është të shfrytëzosh shpejtësinë e saj për të krijuar një skelet (shabllon) të shpejtë të faqes, dhe më pas ta shqyrtosh vetë manualisht, ta pastrosh dhe ta përshtatësh kodin sipas dëshirave dhe kërkesave të tua të sakta!

- Analiza e kodit ekzistues HTML/CSS sipas strukturës dhe pamjes

Kur inteligjenca artificiale e gjeneron kodin bazuar në prompt-in tënd, puna nuk ka përfunduar. Si përdorues i inteligjencës artificiale, duhet ta analizosh rezultatin. Inteligjenca artificiale nuk është e pagabueshme – shpesh gjeneron kod që duket bukur, por në prapaskenë mund të përmbajë mangësi serioze.

Çfarë duhet të kesh parasysh gjatë analizës së kodit të gjeneruar nga IA?

- **Funksionaliteti:** A funksionojnë të gjitha lidhjet? A lejon formulari futjen e tekstit?
- **Përshtatshmëria (Responsive):** Lëvize shfletuesin majtas-djathtas. A ka shtuar IA Media Queries të përshtatshme për pajisje të ndryshme, apo kodi deformohet në pajisjet mobile?
- **Pastërtia e kodit:** A janë klasat të emërtuara qartë (p.sh. karta-produkt), apo janë të paqarta dhe konfuze?

Për të ditur nëse IA ka bërë një punë të mirë, duhet ta „lexosh“ kodin dhe të analizosh **strukturën (HTML)** dhe **pamjen (CSS)** e tij.

Të supozojmë se IA ka gjeneruar pjesën e mëposhtme të kodit:

HTML

```
<div class="karta">karta">
```

```
<h2 class="titull">Mirë se vini</h2>
```

```
<p> Lexo lajmet më të fundit këtu.</p>
```

```
</div>
```

CSS

```
.karta {
```

```
background: #fff;
padding: 20px;
border-radius: 5px;
}
.titull {
color: red;
}
```

Analiza e strukturës (HTML): Kodi është logjik. Kemi një kontejner div që brenda përmban një titull dhe një paragraf. Kjo mundëson që të gjithë elementet të lëvizin dhe të rregullohen së bashku si një tërësi.

Analiza e pamjes (CSS): Kutia ka sfond të bardhë, kënde të rrumbullakosura dhe hapësirë të brendshme (padding) që teksti të mos jetë i ngjitur me skajet. Titulli ka një ngjyrë të kuqe specifike. Struktura dhe pamja janë të ndara mirë.



PYETJE PËR KONTROLLIMIN E NJOHURIVE:

1. Cili term përdoret për udhëzimin tekstual ose kërkesën që përdoruesi ia jep mjetit të IA-së për të marrë kod?

- A) Skript (Script)
- B) Prompt (Prompt)
- C) Etiketë (Tag)
- D) Koment (Comment)

2. Cila është roli kryesor i Media Queries (@media) që asistenti i IA-së i gjeneron në kodin CSS?

- A) Të mundësojnë që faqja të përshtatet me madhësi të ndryshme të ekranit.
- B) Të përshpejtojnë ngarkimin e imazheve në sajt.
- C) Të lidhin faqen web me rrjetet sociale.
- D) Të gjejnë gabime logjike në strukturën HTML.

3. Prompt-i „Krijo një faqe web për atlete“ konsiderohet i paplotë dhe i dobët sepse është shumë i përgjithshëm. Nuk përcakton dizajnin, ngjyrat, strukturën, seksionet, teknologjitë që duhet të përdoren, përshtatshmërinë për pajisjet mobile etj.

4. Nëse përdoren gjerësi fikse në pikselë (px), faqja mund të mos përshtatet me ekranin e telefonit. Elementet mund të bëhen shumë të mëdha për ekranin, duke shkaktuar dalje jashtë ekranit, deformim të paraqitjes ose nevojën për lëvizje horizontale. Me përqindje (%) ose njësi fluide, elementet përshtaten më mirë me madhësinë e ekranit.

5. Shkruaj një prompt të detajuar dhe të strukturuar që do t'i kërkojë asistentit të IA-së të krijojë një meny navigimi (Header) me tre lidhje, me sfond jeshil të errët dhe që përdor Flexbox për renditjen.

6. Nëse në kodin e gjeneruar nga asistenti i IA-së e vëren rreshtin e mëposhtëm: `<meta name="viewport" content="width=device-width, initial-scale=1.0">`, për cilën pajisje asistenti i IA-së është siguruar që faqja të shfaqet saktë?

- A) Vetëm për kompjuterë desktop me monitorë të mëdhenj.

- B) Vetëm për printerë kur faqja shtypet në letër.
- C) Për të gjitha pajisjet, duke përfshirë edhe telefonat mobilë.
- D) Vetëm për shfletuesit që nuk mbështesin CSS.

7. Një nxënës përdor asistentin e IA-së për të gjeneruar një formular regjistrimi. Pasi e kopjon kodin në VS Code, vëren se butoni „Dërgo“ shfaqet në krye të faqes, në vend që të jetë në fund, pas fushave. Analizo në cilën pjesë të kodit (HTML apo CSS) është bërë gabimi logjik në strukturë dhe si do ta korrigjonte.

8. Shqyrto kodin e mëposhtëm CSS të gjeneruar nga IA: `.tekst { color: blue; font-size: 16px; font-style: italic; }`. Nxënësi kërkon që kodi të mos përmbajë shkronja të pjerrëta. Përkthyer se cila deklaratë nga ky kod duhet të fshihet ose të ndryshohet për të përmbushur kërkesën.

9. Vlerëso përparësitë dhe mangësitë e përdorimit të kodit të gjeneruar nga IA, në krahasim me kodin e shkruar manualisht, nga aspekti i të mësuarit dhe të kuptuarit të programimit të fillestarët.

10. Imagjino se asistenti i IA-së të ka gjeneruar një paraqitje të shkëlqyer me dy karta të vendosura njëra pranë tjetrës në kompjuter, por kur e teston kodin në shfletues, vëren se ka harruar të shtojë përshtatshmërinë për pajisjet mobile. Krijë (shkruaj) Media Query-në CSS që mungon, në mënyrë që kartat me klasën .karta të shfaqen njëra poshtë tjetrës në ekranet me gjerësi më të vogël se 600 pikselë.



PËR ATA QË DËSHIROJNË TË DINË MË SHUMË

Inteligjenca artificiale nuk e sheh vetëm prompt-in që i shkruan në atë moment, por „kujton“ edhe bisedën e mëparshme. Kjo hapësirë për ruajtjen e informacionit quhet dritare e kontekstit (context window). Nëse dëshiron të mësosh më shumë, mënyra më e mirë për ta shfrytëzuar këtë është t'i japësh IA-së rregulla të përcaktuara paraprakisht, para se t'i kërkojë të gjenerojë kod.

Për shembull, prompt-i i parë i një dite mund të formulohet kështu:

„Gjatë sesionit të sotëm, do të të kërkoj të më shkruash kod HTML dhe CSS. Detyra jote është të mos përdorësh kurrë deklaratat e vjetruara CSS (si float), të përdorësh gjithmonë Flexbox, dhe komentet në kod të jenë gjithmonë në gjuhën maqedonase.“

Në këtë mënyrë, ti e udhëzon asistentin të punojë sipas standardeve të tua që në fillim, para se të fillojë krijimin e kodit.

2.5.9. PËRSHTATJA DHE PËRMIRËSIMI I KODIT TË GJENERUAR NGA IA

Le të shqyrtojmë një shembull të krijimit të një faqeje web me një mjet të IA-së. Siç e përmendëm më parë, VS Code ka një asistent të IA-së. Hap VS Code dhe shkruaj udhëzimin e mëposhtëm:

Ti je një dizajner dhe programues profesionist web. Më shkruaj një kod të plotë, të pastër dhe të lehtë për t'u kuptuar, HTML dhe CSS të brendshëm, për një faqe web të një klubi shkollor të quajtur „Klubi i IT-së dhe Gaming-ut Kod-Elit“.

Faqja duhet t'i plotësojë standardet e mëposhtme teknike dhe të dizajnit:

1. Struktura (HTML):

- Të ketë një titull kryesor (header) me titull dhe nëntitull që e përshkruajnë klubin.
- Të ketë një kontejner kryesor me dy karta (kuti) për përmbajtjen: njëra për „Zhvillim Web dhe AI“ dhe tjetra për „Zhvillim të Lojërave“.
- Çdo kartë duhet të ketë titull, tekst të shkurtër dhe një buton (link) për veprim.

2. Pamja dhe paraqitja (CSS):

- Përdor një temë moderne të errët (sfond në ngjyrë gri të errët ose blu të errët), me shkronja shumë të çelëta për kontrast të mirë dhe lexueshmëri të lehtë.
- Paraqitjen kryesore të kartave realizoje me Flexbox (display: flex), në mënyrë që në ekranet e kompjuterëve kutitë të vendosen horizontalisht, njëra pranë tjetrës, me gjerësi të barabartë dhe hapësirë të përshtatshme ndërmjet tyre.
- Kartat duhet të kenë cepa të rrumbullakosur (border-radius) dhe korniza diskrete.

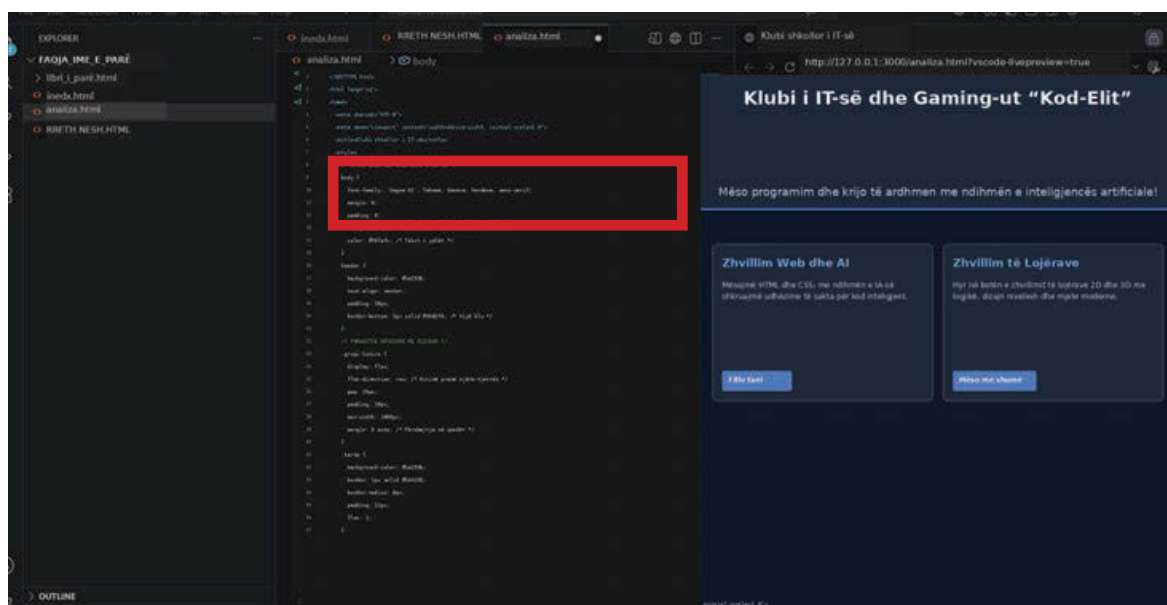
3. Përshtatshmëria (Responsive):

- Përfshi një media query (@media) për pajisjet mobile me gjerësi maksimale prej 768 pikselësh.
- Kur faqja hapet në një pajisje mobile, paraqitja me Flexbox duhet të ndryshojë në kolonë (column), në mënyrë që kartat të vendosen automatikisht njëra poshtë tjetrës dhe të jenë të përshtatshme për t'u parë në ekran të ngushtë.
- Ruaje kodin e gjeneruar me prapashtesën index.html në VS Code dhe hape në shfletues për të parë rezultatin. Vërehet se një pjesë e kodit shfaqet në faqen e internetit, konkretisht: initial-scale=1.0»>.

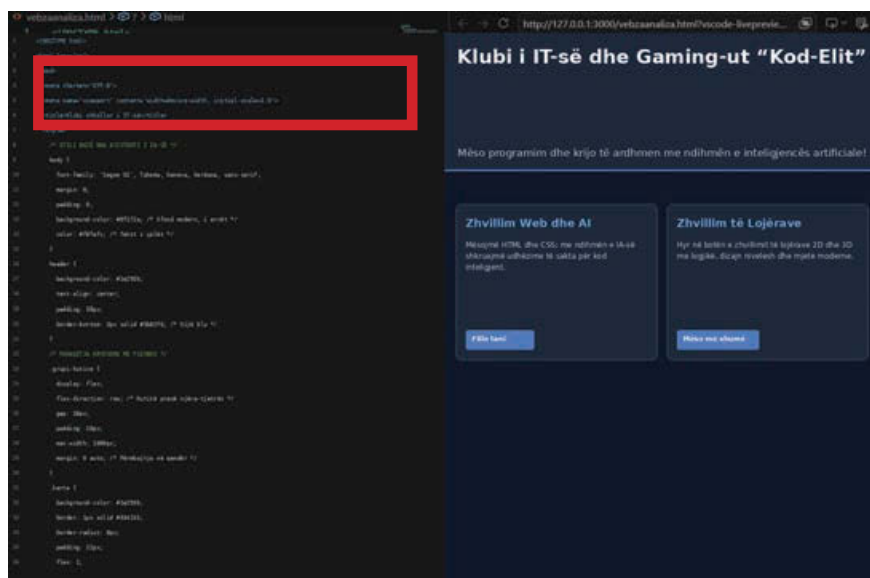


Kjo do të thotë se asistenti i AI-së ka bërë një gabim diku në kod. Analizo pse shërben dhe kur duhet të shkruhet kjo komandë. Kërko nga asistenti i AI-së që ta korrigjojë kodin. Nëse nuk arrin ta korrigjojë, korrigjoje manualisht duke vendosur, brenda seksionit <head>, rreshtin:

<meta name="viewport" content="width=device-width, initial-scale=1.0">



Ruaji ndryshimet, shtyp F5 dhe faqja e internetit do të shfaqet siç duhet.



Kërko ndryshime të reja në faqen e internetit, për shembull: „Në këtë kod dua të ndryshoj temën kryesore. Ndrysho CSS-in në mënyrë që sfondi të jetë jeshil i errët, ndërsa butonat ose vijat kryesore blu të bëhen të verdha neon.“ Ose: „Analizo strukturën e kutive në këtë kod dhe shto edhe një kuti të tretë me titullin ‘Siguria Kibernetike’, e cila të ketë të njëjtin stil dhe të përshtatet me paraqitjen Flexbox.“ Mund të kërkohet gjithashtu kontrolli i përshtatshmërisë (responsive) së faqes së internetit, për shembull: „Shiko media query-n në fund të kodit. Si mund ta ndryshoj kodin që në pajisjet mobile teksti në header të bëhet më i vogël dhe të qendërzohet?“



PYETJE PËR KONTROLLIN E NJOHURIVE:

1. Nxënësi/ja ka kërkuar nga asistenti i AI-së të krijojë një faqe responsive. Në kodin e marrë, rreshti për viewport është shkruar gabimisht: `<meta name="viewport" content="width=device-width", initial-scale=1.0">>` Çfarë do të ndodhë kur kjo faqe të hapet në shfletues?

- A) Faqja nuk do të ngarkohet fare dhe do të shfaqet një ekran i zbrazët.
- B) Në fund të faqes do të shfaqet tekst shtesë, si `initial-scale=1.0»>`.
- C) I gjithë stili CSS do të fshihet dhe teksti do të bëhet i kuq.
- D) Formulari i kontaktit do të fshihet automatikisht.

2. Shiko kodin e mëposhtëm CSS, i cili duhet t'i vendosë kartat njëra pas tjetrës:

```
.grupi-kutive {
  display: block;
  flex-direction: row;
  gap: 15px;
}
```

3. Cila komandë është e gabuar dhe e pengon funksionimin e duhur të paraqitjes Flexbox?

- A) `gap: 15px;`
- B) `flex-direction: row;`
- C) `display: block;`
- D) Emri i klasës `.grupacija-kutii` ka një gabim shtypi.

4. Shiko kodin e mëposhtëm HTML të gjeneruar nga asistenti i AI-së:

```
<div class="karta">
  <h3>Zhvillimi i Web-it</h3>
  <p>Mëso të kodosh me ndihmën e AI-së.
</div>
```

Shpjego çfarë gabimi strukturor ekziston në këtë pjesë të kodit dhe si mund të ndikojë ai në pamjen e elementeve të tjera që ndodhen poshtë tij.

5. Në pjesën e media query-ve, asistenti i AI-së ka bërë gabimin e mëposhtëm në sintaksë:

```
@media max-width: 768px {
  .permbajtja-kryesore {
    flex-direction: column;
  }
}
```

Pse kjo media query nuk funksionon në shfletues dhe cilët shenja mungojnë në rreshtin e parë që kodi të jetë i saktë?

6. Nxënësi/ja dëshiron ta përshtatë kodin e gjeneruar nga AI-ja për të ndryshuar ngjyrën e sfondit të kartës. Asistenti i AI-së ka shkruar kodin e mëposhtëm:

```
.karta {
  background-color: #0f172a;
  padding: 20px;
  border-radius: 12px;
}
```

Cila shenjë sintaksore mungon në këtë kod dhe cili rregull CSS nuk do të zbatohet fare (do të shpërfilllet nga shfletuesi)?

7. Cilin rregull duhet ta fshish ose ta ndryshosh për të përmbushur kërkesën që kodi të mos ketë shkronja të pjerrëta (italike)?

```
.buton {
  display: inline-block;
  background-color: #3b82f6;
  color: white;
  font-style: italic;
}
```

8. Shiko këtë përpjekje të asistentit të AI-së për të krijuar një paraqitje responsive për telefonat celularë:

```
.karta {
  width: 400px;
}
```

```
@media (max-width: 600px) {
```

```
.karta {
  width: 400px;
}
}
```

Analizojë kodin dhe shpjego pse ky dizajn nuk do të jetë i përshtatshëm për ekranet me gjerësi 360 pikselë (një madhësi standarde për telefonat celularë) dhe si duhet të rregullohet vlera e width në media query.

9. Nxënësi/ja përdor kodin e mëposhtëm për të krijuar titullin kryesor të faqes:

```
<h1 id="titull-kryesor" class="titull-kryesor">Klubi IT Kod-Elit</h1>
```

Nëse në pjesën CSS ka dy rregulla: njëri targeton me .titull-kryesor { color: red; }, ndërsa tjetri me #titull-kryesor { color: blue; }, me cilën ngjyrë do të shfaqet titulli dhe pse? Shpjego parimin e specifikës së selektorëve.

10. Asistenti i AI-së ka gjeneruar kod ku dy kartat duhet të jenë njëra pranë tjetrës në kompjuter, ndërsa njëra nën tjetrën në pajisjet mobile. Por, kodi duket kështu:

```
.permbajtja-kryesore {
  display: flex;
  flex-direction: column;
}

@media (max-width: 768px) {
  .permbajtja-kryesore {
    flex-direction: column;
  }
}
```

Vlerëso funksionalitetin e këtij kodi për të dyja pajisjet (kompjuter dhe celular) dhe propozo vlerat e sakta për flex-direction për të korrigjuar paraqitjen.

11. Ky është një kod HTML për dy karta, në të cilin asistenti i AI-së ka bërë një gabim logjik gjatë përcaktimit të klasave:

```
<div class="permbajtja-kryesore">
  <div class="karta">
    <h3>Zhvillimi i Web-it</h3>
    <p>Mësojmë kodimin me AI.</p>
  </div>
  <div class="permbajtja-kryesore">
    <h3>Zhvillimi i lojërave</h3>
    <p>Krijojmë 2D dhe 3D lojra.</p>
  </div>
</div>
```

Krijë kodin e saktë HTML duke gjetur dhe korrigjuar klasën e gabuar në kutinë e dytë, në mënyrë që ajo të fitojë të njëjtin stil si karta e parë.



Qasja „Fillimisht celulari“ (Mobile-First) përmes AI-së

Shumica e fillestarëve i kërkojnë AI-së të krijojë një faqe për kompjuter, dhe më pas e përshtatin atë për celular. Por, në industrinë moderne të IT-së përdoret qasja Mobile-First. Kjo do të thotë se faqja fillimisht dizajnohet për ekranin më të vogël, dhe më pas zgjerohet për monitorët më të mëdhenj.

Kur i kërkon AI-së të gjenerojë kod duke përdorur këtë qasje, udhëzimet duhet t'i formulosh kështu:

Kërko që CSS-i bazë të shkruhet në një kolonë (për telefon celular).

Kërko që media query-t të përdorin min-width (gjerësi minimale), në vend të max-width.

Në këtë mënyrë, me @media (min-width: 1024px), inteligjenca artificiale do të shtojë rregulla që do të aktivizohen vetëm pasi ekrani të bëhet mjaftueshëm i madh, si në kompjuter, gjë që rezulton në ngarkim më të shpejtë të faqes në telefonat celularë.

Pastrimi automatik i „kudit të panevojshëm të AI-së“ (Refactoring)

Inteligjenca artificiale ka tendencë të shkruajë shumë kod. Shpesh, për të njëjtin funksion mund të krijojë tri klase të ndryshme ose të shtojë rregulla CSS që shfletuesi i konsideron të paracaktuara. Procesi i pastrimit dhe optimizimit të këtij kodi quhet refaktorim (Refactoring).

Pasi AI-ja të gjenerojë kodin, një teknikë e avancuar është t'ia dërgosh përsëri të njëjtin kod, së bashku me një kërkesë për optimizim:

„Ky është kodi që gjenerove. Tani refaktorizoj: bashko klasat CSS të dyfishta, hiq margjinat e panevojshme që përsëriten dhe bëje strukturën HTML më të shkurtër dhe më të thjeshtë për t'u lexuar.“

Integrimi në VS Code me shtojca inteligjente

Programuesit e avancuar nuk e kopjojnë kodin nga shfletuesi (si ChatGPT), por përdorin shtojca direkt në redaktuesin VS Code (si GitHub Copilot ose Codeium).

Këto mjete kanë qasje në të gjithë skedarët e dosjes tënde njëkohësisht. Kjo do të thotë se ato jo vetëm që gjenerojnë kod të ri, por mund të analizojnë edhe stilin tënd të shkruarit dhe të të propozojnë automatikisht rreshtin e ardhshëm të kodit, edhe para se të fillosh ta shkruash, vetëm me shtypjen e një tasti (Tab).

5.2.10 KRIJIMI I NJË PROJEKTI WEB

Pasi mësove se si funksionon inteligjenca artificiale dhe si bëhen përshtatjet në kod, është koha të shohim të gjithë procesin e krijimit të një faqeje interneti, nga ideja e parë në letër deri te pamja përfundimtare.

Krijimi profesional i një faqeje interneti zhvillohet gjithmonë përmes 4 hapave kryesorë, të cilët duhet të ndiqen sipas një renditjeje të saktë.

1. Planifikimi (Çfarë do të krijojmë dhe për kë?)

Planifikimi është themeli i faqes së internetit. Para se të shkruajmë qoftë edhe një rresht të vetëm kod, duhet t'u përgjigjemi pyetjeve të mëposhtme:

- **Cili është qëllimi i faqes?** (A është një portofol, një faqe interneti e shkollës apo një dyqan online?)
- **Cila është audienca e synuar?** (A po e krijojmë për adoleshentë, mësues apo lojtarë?)
- **Çfarë do të përmbajë?** Bëhet një listë e pjesëve të nevojshme: header-i, kartat me produkte, galeria me fotografi dhe formulari i kontaktit.

2. Struktura (Si do të organizohet kodi?)

- Struktura i referohet skeletit të faqes, përkatësisht organizimit logjik të elementeve HTML. Dizajnerët profesionistë në këtë fazë krijojnë një **skicë strukturore (Wireframe)** – një skicë të thjeshtë me drejtkëndësha, e cila tregon se ku do të vendoset secili element.

Në kod, këtë e shndërrojmë në HTML5 të pastër dhe semantik:

- Përdorim <header> për pjesën e sipërme.
- Përdorim kontejnerë (<div> ose <section>) me klasa të qarta për të grupuar elementet.
- Kujdesemi për hierarkinë e titujve (<h1> për titullin kryesor, ndërsa <h2> dhe <h3> për nëntitujt e kartave).

3. Dizajni (Si do të duket faqja?)

Pasi të kemi një strukturë të qëndrueshme, kalojmë te stilizimi me CSS. Dizajni përcakton identitetin vizual dhe përvojën e përdoruesit.

Këtu zbatojmë të gjitha rregullat CSS të mësuara:

- **Renditja (Layout):** Përdorim Flexbox që elementet të shtrihen dhe të renditen në mënyrë të përshtatshme.
- **Tipografia dhe ngjyrat:** Zgjedhim fonte që lexohen lehtë dhe një paletë prej 2 deri në 3 ngjyrash të harmonizuara (për shembull, një temë moderne e errët me detaje blu me kontrast).
- **Hapësira:** Përdorim margin dhe padding për të shmangur grumbullimin e elementeve dhe që ato të mos jenë „të ngjitura“ me njëra-tjetrën.
- **Përshtatshmëria (Responsive):** Përmes media query-ve (@media) sigurohemi që dizajni të përshtatet nga tre kolona në kompjuter në një kolonë në telefon celular.

4. Integrimi i përmbajtjes (Mbushja e faqes me të dhëna reale)

Integrimi është hapi përfundimtar, ku teksti „i rremë“ ose gjenerik që na ka dhënë asistenti i AI-së zëvendësohet me informacione të vërteta dhe reale:

- E shkruajmë tekstin përfundimtar në paragrafët (<p>).
- I zëvendësojmë lidhjet e përkohshme me adresa të sakta drejt imazheve reale dhe faqeve të jashtme.
- I lidhim butonat me veprimet përkatëse.

Vetëm pasi të kenë përfunduar të 4 hapat sipas këtij rendi, mund të themi se kemi një faqe interneti të plotë, funksionale dhe të krijuar në mënyrë profesionale.

Për t'i përvetësuar më mirë fazat e përgatitjes, më së miri është të krijosh një faqe interneti personale (portofol), ku do të paraqesësh të gjitha ushtrimet, lojërat ose faqet e internetit që ke krijuar ose planifikon t'i krijosh në të ardhmen. Kjo është diçka që programuesit profesionistë e kanë gjithmonë.

Arkitektura dhe elementet që duhet të përmbajë faqja e internetit:

1. Struktura dhe etiketat semantike HTML

Faqja duhet të jetë e ndarë në pjesë të qarta dhe logjike:

- `<header>`: Në pjesën e sipërme, me emrin dhe një nëntitull të shkurtër (p.sh. „Zhvillues i ardhshëm Web“).
- `<main>`: Pjesa kryesore, ku do të vendoset përmbajtja.
- `<section>` me klasën `.proekti`: Këtu do të vendosen kartat për projektet e tij.
- `<footer>`: Në pjesën e poshtme, me tekstin për të drejtat e autorit.

2. Renditja (Layout) me Flexbox

Në seksionin e projekteve duhet të ketë **3 karta** (për shembull: „Faqja e internetit e shkollës“, „Loja ime e parë“, „Ushtrim me AI“).

- Këto 3 karta duhet të vendosen në një kontejner prindëror i cili do të ketë `display: flex;` dhe `flex-direction: row;`, në mënyrë që në kompjuter të renditen bukur në një rresht.

3. Dizajn përshtatës (@media query)

Faqja duhet të duket në mënyrë të përsosur edhe në telefon celular.

- Duhet të shtohet `@media (max-width: 768px)`, ku renditja me Flexbox do të ndryshojë në `flex-direction: column;`, në mënyrë që kartat të vendosen njëra poshtë tjetrës.
- Në pjesën `<head>` duhet patjetër të vendoset rreshti i saktë për viewport, në mënyrë që të mos shfaqet tekst i tepërt në ekran.

Si të përdoret asistenti i AI-së për këtë projekt? (Procesi i punës)

Nxënësit duhet ta realizojnë projektin përmes 3 hapave të mëposhtëm, duke përdorur AI-në në VS Code:

- **Prompti i parë (Gjenerimi i skeletit)**: Nxënësi i shkruan AI-së: „Ti je mentori im. Më shkruaj strukturën HTML5 dhe CSS-in fillestar për portofolin tim personal IT. Dua një sfond gri të errët, me header, pjesën kryesore me 3 karta për projekte të vendosura me Flexbox dhe footer. Përfshi edhe përshtatshmërinë për pajisjet mobile nën 768px.“
- **Testimi dhe gjetja e gabimeve (Analiza)**: Nxënësi e hap kodin në shfletues. Kontrollon nëse AI-ja ka harruar ndonjë pikëpresje (;), nëse teksti shtrembërohet ose nëse rastësisht është shfaqur ndonjë tekst i pjerrët (italik) që nuk e dëshiron.
- **Prompti i dytë (Përshtatja dhe përpunimi)**: Pasi ta analizojë kodin, nxënësi kërkon ndryshimin: „Kodi është super, por tani në kartën e dytë ndryshojë titullin në 'Projekti im Asistenti i IA-së', ndërsa ngjyrën e butonave në të gjitha kartat bëje të kaltër të çelët, me kënde të rrumbullakosura.“
- **Personale**: Nxënësi/ja prezanton veten dhe ndihet si një programues/e i/e vërtetë.

- **Përmban të gjitha standardet:** Kërkon të menduarit për strukturën (HTML), estetikën dhe përshtatshmërinë (CSS), si dhe komunikimin e duhur me teknologjinë (Prompt Engineering).
- **Kontrollohet lehtë:** Si mësues/e ose vlerësues/e, mund të shihni menjëherë nëse Flexbox-i funksionon kur zvogëloni dritaren e shfletuesit.



PYETJE PËR KONTROLLIN E NJOHURIVE:

1. Nxënësi/ja dëshiron të krijojë një faqe interneti për klubin e IT-së, por para se të fillojë me kodimin, hulumton se cilat informacione u nevojiten më shumë bashkëmoshatarëve dhe përgatit një listë të seksioneve. Në cilën fazë të realizimit të projektit ndodhet nxënësi/ja në atë moment?

- A) Dizajni
- B) Struktura
- C) Planifikimi
- D) Integrimi i përmbajtjes

2. Kur bëjmë një „skicë strukturore“ (Wireframe) ose një skicë të thjeshtë me drejtkëndësja për të përcaktuar se ku do të vendosen titujt, imazhet dhe kartat, ne në fakt përcaktojmë:

- A) Ngjyrën e sfondit të faqes
- B) Strukturën HTML të faqes
- C) Shpejtësinë e ngarkimit të serverit
- D) Lidhjen me dokumentet e jashtme PDF

3. Shqyrto situatën e mëposhtme: Nxënësi e ka përfunduar kodin HTML dhe tani dëshiron t'i rendisë kartat e projekteve me ndihmën e Flexbox, t'u rrumbullakosë këndet kutive dhe të shtojë media query për telefonat celularë. Shpjego cilën fazë të realizimit të projektit po e kryen nxënësi në këtë hap dhe cilën gjuhë po përdor.

4. Çfarë paraqet faza e „Integritit të përmbajtjes“ dhe çfarë duhet të bëjë saktësisht nxënësi me lidhjet që asistenti i AI-së i ka gjeneruar si href=»#», në mënyrë që portofoli i tij të bëhet funksional?

5. Nxënësi ka vendosur lidhjen e mëposhtme në kodin e tij për të shfaqur formularin e kontaktit: `<a href=»C:/Users/Ucenik/Desktop/IT_Klub/kontakt.html« class=»buton«>Hap</a>`. Analizo pse ky shteg është vendosur gabimisht (shteg absolut në vend të shtegut lokal) dhe si do të ndikojë kjo nëse dikush tjetër përpiqet ta hapë faqen në kompjuterin e tij.

PËR ATA QË DËSHIROJNË TË DINË MË SHUMË



Planifikimi në zhvillimin e uebit përfshin krijimin e një harte të detajuar të përdoruesit (User Flow), e cila parashikon me saktësi çdo lëvizje dhe klikim të vizitorit nëpër faqe. Në këtë fazë të avancuar, profesionistët përdorin mjete të specializuara digjitale, si Figma ose Adobe XD, për të krijuar një prototip interaktiv, i cili duket dhe funksionon si një ueb-faqe e vërtetë, ende pa u shkruar rreshti i parë i kodit HTML.

Gjithashtu, planifikohet edhe e ashtuquajtura strategji SEO (optimizimi për motorët e kërkimit), përmes së cilës përcaktohen paraprakisht fjalët kyçe që do të përdoren në tituj dhe në meta-etiketa, në mënyrë që faqja të jetë më e lehtë për t'u gjetur nga motorët e kërkimit.

