

Todor Dimov

BİLGİSAYAR-BİLİŞİM

I. (birinci) Sınıf

Lise Eğitimi

Ulusal Ders Kitapları Komisyonu tarafından onaylanan, lise eğitiminin I. (birinci) sınıfı Bilişim dersi ders kitabının onaylanması ve kullanılmasına ilişkin 07.04.2026 tarihli ve 26-420/3 sayılı Karar ile.

Üsküp, 2026

BİLGİSAYAR-BİLİŞİM

I. (birinci) Sınıf Lise Eğitimi

назив на оригинал:

Информатика

за I (прва) година гимназиско образование

Yayınevi:

ÜTHŞ «STUDENTSKİ SERVİS» Ltd. Şti. – Üsküp

Yazar:
Todor Dimov

Hakemler:

Prof. Dr. Gazmen Caferi

Prof. Dr. Atanas Hristov

Jaklina Miloševa

Editör:

Simona Ristovska

Dil Editörü:

Doç. Dr. Melek Nuredini

Türkçeye Çeviren:

Doç. Dr. Melek Nuredini

Tasarım ve Teknik Hazırlık:

ÜTHŞ «STUDENTSKİ SERVİS» Ltd. Şti. – Üsküp

Baskı:

ÜTHŞ «STUDENTSKİ SERVİS» Ltd. Şti.

Pero Nakov Caddesi No:112 – Üsküp

Baskı Adedi:

200 adet

Üsküp, 2026

CIP - Каталогизација во публикација

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

004(075.3)=512.161

DIMOV, Todor

Bilgisayar-Bilişim I. (birinci) sınıf lise eğitimi / Todor

Dimov ; türkçeye Çeviren Melek Nuredini. - Üsküp : Studentski servis,

2026. - 263 стр. : илустр. ; 26 см

Превод на делото: Информатика за I (прва) година гимназиско образование

ISBN 978-608-4777-48-9

COBISS.MK-ID 69405701

İÇİNDEKİLER

KONU 1: MODERN DİJİTAL TEKNOLOJİLERİ	7
1.1 Dijital Teknolojinin Günlük Hayattaki Rolü	10
1.2 Bilgisayar Sistemi ve Temel Bileşenleri	15
1.3 Von Neumann Mimarisi ve Bilgisayarın Bileşenleri	16
1.4 Giriş ve Çıkış Aygıtları – Sınıflandırma ve Karşılaştırma	21
1.5 Modern Dijital Teknolojiler: Dokunma, Holografi, 3D Projeksiyon	26
1.6 İşletim Sisteminin ve Uygulama Yazılımlarının Çalışma Ortamı	31
İşletim Sistemi	31
1.7 Dijital Teknolojinin Kullanımı için Etik Kurallar	36
1.8 Klasörlerin ve Dosyaların Hiyerarşik Yapıda Kullanılması	40
1.9 Yazılım Lisanslama ve Creative Commons	44
1.10 Yapay Zekâ (Genel ve Üretken)	48
KONU 1: GENEL TEKRARLAMA	55
KONU 2: ÇEVİRİM İÇİ YAŞAM	57
2.1 Bilgisayar Ağı Kavramı	59
Bilgisayar Ağlarının Amacı	59
İnternet	60
2.2 Bilgisayar Ağı Türleri	61
İnternet Hizmetleri	64
WWW	64
Arama Motorları (Search Engines)	64
Elektronik Posta	65
ftp	67
Görüntülü Görüşmeler	67
Elektronik Ticare	67
Elektronik Bankacılık	68
Etkileşimli İletişim	69
2.3 Web Teknolojilerinin Gelişimi	73
2.4 Bilgi Kaynağı Olarak Web	76
2.5 Dijital İz	79
2.6 İnternetin Tehlikeleri	83
2.7 Güvenli İnternet Kullanımı	87
KONU 2: GENEL TEKRARLAMA	90
KONU 3: C++ PROGRAMLAMA	91
3.1 Mantıksal ve Algoritmik Ödevler	92
Programlamanın Yeni Bir Düşünme Dili Olarak Kullanılması	93
3.2 Bilgi Kavramları	96
Veri Yapıları	97
3.3 İkili Sayılar: Gösterim ve Kullanım	100
3.4 Kodlama ve Şifreleme Temelleri	103
Kodlama	104
3.5 Programlama Nedir?	108
3.6 Programlama Dili ve Çevirici (Derleyici/Yorumlayıcı)	111
3.7 Programlama Dilleri Arasındaki Farklar: Scratch, C++, Java, Python, PHP, Lisp	114
3.8 Entegre Programlama Ortamı	117
3.9 Program Yazma, Çalıştırma ve Hata Ayıklama	120
3.10 Programlamada İfade (Statement)	123
3.11 Sözde Dil (Psödo Kod)	126
3.12 Önerme Dizisi	129
3.13 Değişkenler ve Sabitler	132
3.14 Değişken Türleri	135
3.15 Bir Değişkene Değer Atama	138

3.16 Değişken Değerinin Gösterimi	141
3.17 Aritmetik İşlemler ve İfadeler	145
3.18 Veri Girişi	149
3.19 Karşılaştırma İşlemleri	152
3.20 Mantıksal İfadeler	156
3.21 İki Seçenekli Karar Yapısı	160
3.22 if-else Yapısı İçin Alıştırma Soruları	164
3.23 Veri Girişi	165
3.24 İç içe if-else Yapısı için Alıştırmalar	168
3.25 Birden Fazla Seçenekten Seçim Yapısı	169
3.26 Birden Fazla Seçenekten Seçim Yapısı için Alıştırma Soruları	174
3.27 While Döngüsü	175
3.28 While Döngüsü Yapısı için Alıştırma Soruları	179
3.29 Koşullu Döngü – do while Yapısı	180
3.30 do while Döngüsü Yapısı için Alıştırma Soruları	184
3.31 for Döngüsü Yapısı	185
3.32 for Döngüsü Yapısından Alıştırma Görevleri	189
KONU 3: GENEL TEKRARLAMA	191

KONU 4: METİN ÇALIŞMASI 193

4.1 Biçim Stilleriyle Çalışma	195
Mevcut Stillerin Uygulanması	195
Mevcut Stillerin Düzenlenmesi	196
Kendi Stillerini Oluşturma	197
Başlık ve Alt Başlık	198
Neden Stiller Kullanılmalıdır	199
Sonuç	199
Pratik Alıştırma: Microsoft Word'de Stillerle Çalışma	199
4.2 İçindekiler ve Dizinler	203
İçindekiler Tablosu Nedir?	203
İçindekiler Tablosunu Özelleştirme	204
Dizin Nedir?	205
Pratik Etkinlik	206
4.3 Şablonlar ve Formlar	208
Şablon	208
Formlar	209
Belgelerin Korunması	213
Koruma Türleri	213
KONU 4: GENEL TEKRARLAMA	216

KONU 5: ELEKTRONİK TABLO HESAPLAMA PROGRAMI 217

5.1 Veri Tabanı Olarak Tablo	219
Çalışma Sayfalarıyla İşlemler	221
Tabloyu Düzenleme	222
Hücreleri Düzenleme	223
Otomatik Oldurma	224
5.2 Formüllerin ve İşlevlerin İleri Düzeyde Kullanımı	228
Elektronik Tablo Programındaki Kavramlar	228
Karmaşık Fonksiyonlar	231
5.3 Grafiklerle İleri Düzey Çalışma	238
Grafikğin Öğeleri	239
Grafikğin Ek Olarak Düzenlenmesi	240
5.4 Sıralama	244
5.5 Filtreleme	248
5.6 Rapor Oluşturma – Pivot Tablo	253
5.7 Veri Koruma	257
KONU 5: GENEL TEKRARLAMA	263

Ön Söz

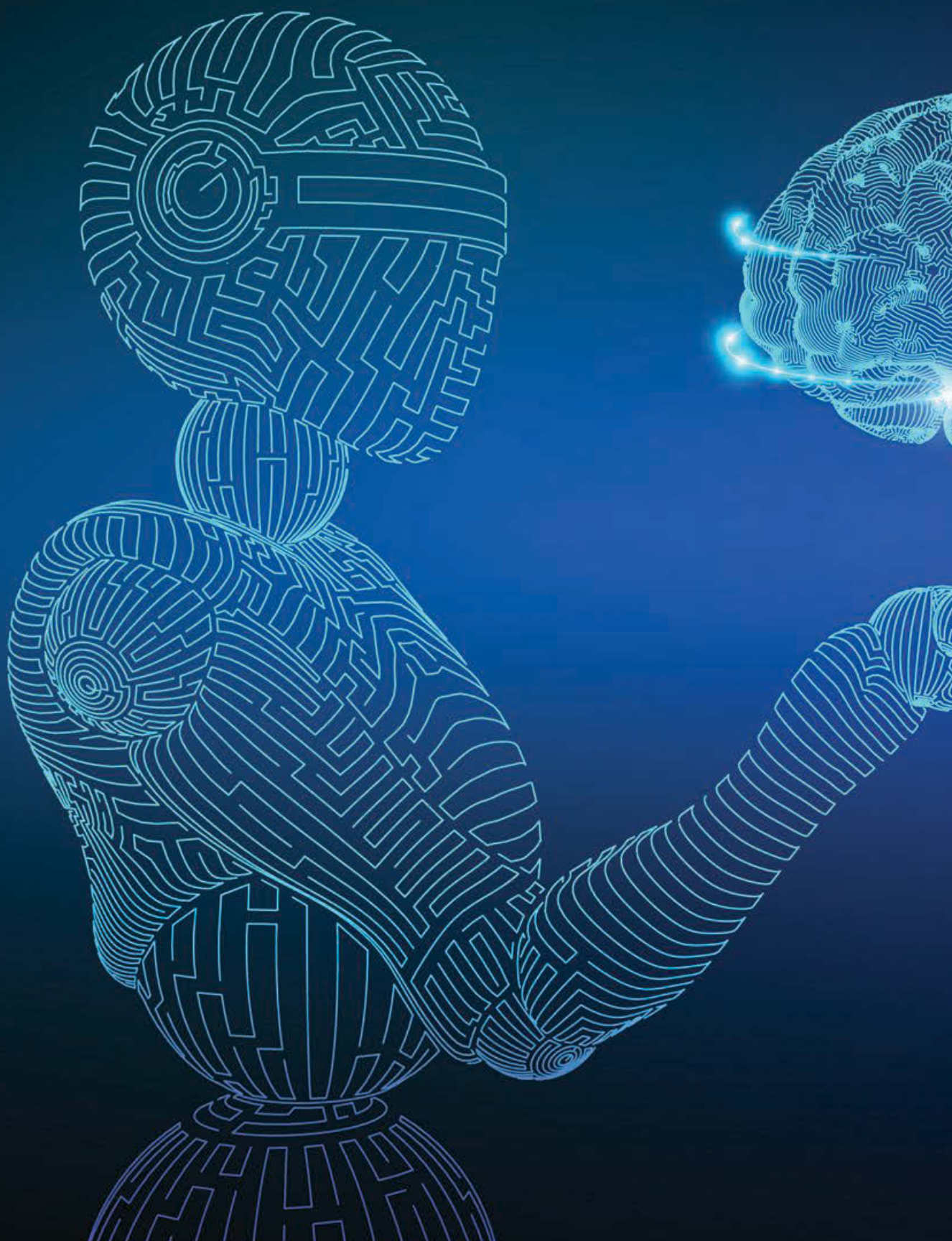
Bilişim dünyasına hoş geldiniz. Bu dünya yalnızca bizi çevreleyen bir alan değil, aynı zamanda onu anlamaya, şekillendirmeye, bilinçli ve sorumlu bir biçimde kullanmaya davet eden bir dünyadır. Bu ders kitabı yalnızca bilgi ve yönergeler içeren bir kitap değil, aynı zamanda geleceğe açılan bir rehberdir. Elinizdeki bu kitap, sizi dijital okuryazarlığın temelleri, eleştirel düşünme becerileri ve teknolojinin yaratıcı biçimde kullanılmasına ilişkin temel bilgilerle tanıştıracak bir kaynaktır. Bu kitap aracılığıyla modern toplumun temel unsurlarını, bilgisayarlar ve internetten programlamaya, veri analizinden dijital ortamdaki etik ilkelere kadar uzanan geniş bir yelpazede inceleme fırsatı bulacaksınız.

Her sayfa, yeni olanaklara açılan bir kapıdır. Modern dijital teknoloji, çevrim içi yaşam, C++ ile programlama, metin işleme ve elektronik tablo uygulamaları olmak üzere beş ünite boyunca mantıklı düşünmeyi, sorumluluk bilinciyle hareket etmeyi ve anlamlı ürünler ortaya koymayı sağlayacak çeşitli etkinlik ve uygulamalarla karşılaşacaksınız. Ekranın arkasındaki dünyanın nasıl işlediğini, verilerin ağlar üzerinden nasıl aktarıldığını, bir problemi çözen programın nasıl geliştirildiğini, bilgilerin nasıl düzenlendiğini ve görsel olarak nasıl sunulduğunu öğreneceksiniz. Daha da önemlisi, burada edineceğiniz her bilgi ve becerinin sizi kişisel ve mesleki gelişiminize bir adım daha yaklaştırdığını fark edeceksiniz.

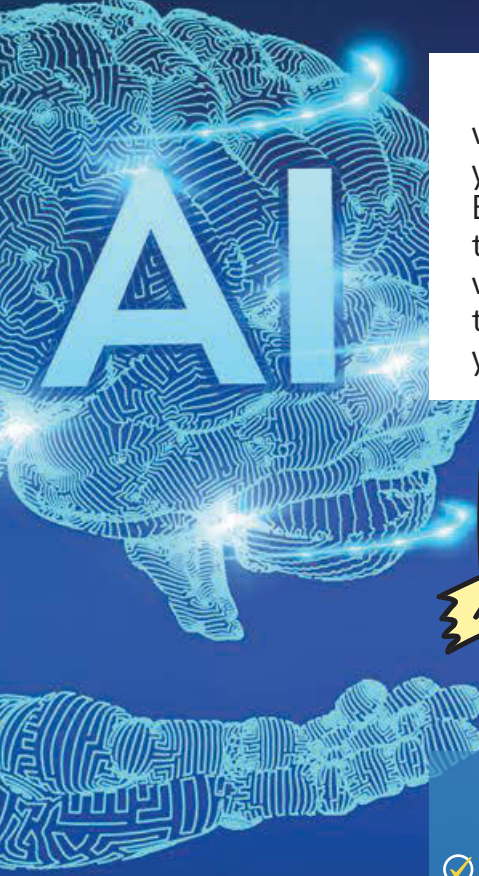
Bu ders kitabında yalnızca teknolojiler öğretilmemektedir aynı zamanda insan ile teknoloji arasındaki ilişki de ele alınmaktadır. Amaç, yalnızca dijital okuryazar bireyler değil, aynı zamanda dijital ortamda sorumluluk bilinciyle hareket eden bireyler yetiştirmektir. İnternetin sunduğu olanakların yanı sıra taşıdığı riskleri de inceleyecek, dijital ayak izi kavramının ne anlama geldiğini öğrenecek ve kendinizi olduğu kadar başkalarını da dijital ortamda korumanın yollarını keşfedeceksiniz. Ayrıca, yapay zekânın etik boyutunu, yeniliklerin günlük yaşamdaki rolünü ve teknolojinin seçimlerimizi, düşüncelerimizi ve değerlerimizi nasıl şekillendirdiğini değerlendirme fırsatı bulacaksınız.

Eğitimin yalnızca bilgi aktarmakla kalmayıp aynı zamanda ilham vermesi gerektiğine inanıyoruz. Bu nedenle bu ders kitabı, soru sormanızı, araştırmanızı, sorgulamanızı ve üretmenizi teşvik edecek şekilde hazırlanmıştır. Bilişim dersi yalnızca bir ders değildir, aynı zamanda hızla değişen dünyaya hazırlanmanız için önemli bir fırsattır. Teknoloji kendi başına bir amaç değil, daha iyi, daha adil ve daha yaratıcı bir dünya oluşturmak için yararlanabileceğimiz bir araçtır. Bu nedenle, bu öğretim yılı boyunca işleyeceğiniz her ders, gerçekleştireceğiniz her etkinlik ve hazırlayacağınız her proje, geleceğinizi oluşturan mozaikin önemli bir parçasını meydana getirecektir.

Bu ders kitabının size ilham vermesi, cesaret kazandırması ve öğrenme isteğinizi güçlendirmesi dileğiyle, bilişim yolculuğuna hoş geldiniz. Bu yolculuk değerli, ilgi çekici ve hepsinden önemlisi sizin için anlamlı bir öğrenme deneyimi olacaktır.



KONU 1: MODERN DİJİTAL TEKNOLOJİLERİ



Modern dijital teknoloji, bilgilerin işlenmesi, depolanması ve iletilmesi amacıyla elektronik sinyallerden ve verilerden yararlanan çok çeşitli araç ve sistemleri kapsamaktadır. Bu bilgiler, bireyler ve kurumlar açısından anlam ve değer taşımaktadır. Modern dijital teknoloji, iletişim, otomasyon ve yenilikçiliğin temelini oluşturmakta, bilgisayarlar ve akıllı telefonlar gibi basit araçlardan gelişmiş uygulamalara kadar yaşamın birçok alanında önemli bir rol oynamaktadır.

Yeni Kavramlar

- Creative Commons lisansları
- Genel yapay zekâ
- Üretken yapay zekâ.

Bu konuyu tamamladığında...

- ✓ Bilgisayar sisteminin temel bölümlerini sıralayabilir, açıklayabilir ve bu bölümlerin temel işlevlerini belirtebilirsiniz.
- ✓ Bilgisayarda belleğin ve işlemcilerin rolünü açıklayabilirsiniz.
- ✓ Farklı bellek türlerini sıralayabilirsiniz.
- ✓ Donanım aygıtlarının işlevlerini kendi ifadelerinizle açıklayabilirsiniz.
- ✓ Masaüstü bilgisayar, dizüstü bilgisayar, tablet ve akıllı telefon gibi farklı aygıtlar arasındaki farkları açıklayabilirsiniz.
- ✓ Virüs ve antivirüs programı kavramlarını açıklayabilirsiniz.
- ✓ Bilgisayarın etik kullanımına ilişkin temel kurallara uyabilirsiniz.



TEKRARLAMA SORULARI:

1. Bilgisayarınızı veya dizüstü bilgisayarınızı inceleyin ya da gerçek bir cihaza erişiminiz yoksa bir görsel kullanın! Daha sonra aşağıda gösterilen tabloya benzer bir tabloyu bir metin işleme programında oluşturun ve ilk satırdaki örneği izleyerek tabloyu doldurun!

Bilgisayarın bölümü	Açıklama (kendi ifadelerinizle)	İşlevi
CPU (işlemci)	Tüm komutları yerine getirir ve bilgisayarın ne yapması gerektiğini "düşünür"	Programlardan gelen tüm komutları yerine getirir ve diğer bileşenlerin çalışmasını yönetir.
RAM (bellek)		
Sabit disk (HDD/SSD)		
...		

2. Aynı tabloyu kullanın, AŞAĞIDAKİ örnekte gösterildiği gibi iki sütun daha ekleyin ve sütun başlıklarını değiştirerek farklı aygıtlar arasında karşılaştırma yapın! Tabloyu bildiğiniz diğer aygıtları da ekleyerek tamamlayın!

Aygıt	Kullanım alanı (nerede?)	Taşınabilirlik Evet/Hayır	Ekran boyutu	Pil (çalışma süresi/saat)
Masaüstü bilgisayar				
Dizüstü bilgisayar				
Tablet				
Akıllı telefon				
...				

3. Elektronik ortamda “Virüslerden nasıl korunabiliriz?” konulu bir poster hazırlayın. Posterde AŞAĞIDAKİ unsurlar yer almalıdır:

- Bilgisayar virüsünün ne olduğuna ilişkin kısa bir açıklama.
- Kendinizi korumak için üç öneri.
- Bir antivirüs programının adı.

4. Bilgisayar ve internetin etik kullanımıyla ilgili beş kural yazın! Kuralları “Yapılmamalıdır...” veya “Her zaman yapılmalıdır...” şeklinde ifade edebilirsiniz.



İLERİDEKİ KONULARDA ŞUNLARI ÖĞRENECEKSİN:

- Kullanım amacına bağlı olarak bir bilgisayar sistemi modelini belirleyebileceksin.
- Modern dijital teknolojileri sıralayabilecek ve açıklayabileceksin.
- Dijital teknolojilerin modern toplumdaki anlamını ve rolünü açıklayabileceksin.
- Farklı amaçlar için uygun yazılımları kullanabileceksin.
- İşletim sisteminin ayarlarını özelleştirebileceksin.
- Yapay zekâyı özellikleri üzerinden açıklayabileceksin.

Modern toplumda dijital teknoloji, yaşamın neredeyse her alanında bulunmaktadır. Otomasyonu, bağlantı kurmayı ve verilerin işlenmesini yakın zamana kadar hayal bile edilemeyen yöntemlerle mümkün kılmaktadır. Evde, iş yerinde, eğitimde ve sağlık alanında dijital teknoloji, günlük yaşamı kolaylaştıran ve yaşam kalitesini artıran araçlar sunmaktadır. Dijital teknolojinin yardımıyla bilgilere daha hızlı ulaşılır, iletişim gerçek zamanlı olarak gerçekleştirilir ve saatler süren işlemler artık birkaç saniye hatta milisaniye içinde tamamlanabilir. Dijital teknolojinin rolünü anlamak, dijital okuryazarlığın temel bir parçasıdır ve modern topluma etkin katılım için önemli bir ön koşuldur.



Resim 1 Dijital teknolojinin günlük yaşamdaki kullanımı

Evlerde dijital teknoloji, akıllı telefonlar, televizyonlar, sesli asistanlar, sıcaklık sensörleri ve güvenlik sistemleri gibi aygıtlar aracılığıyla kullanılmaktadır. Bu sayede kullanıcılar, uygulamalar yardımıyla aydınlatmayı, ısıtmayı veya ev eğlence sistemlerini kontrol edebilmektedir. Akıllı aygıtlar yalnızca konforu artırmakla kalmaz, aynı zamanda otomatik senaryolar ve uzaktan yönetim sayesinde enerji, zaman ve kaynak tasarrufu sağlar.

Eğitim alanında öğrenciler ve öğretmenler, e-öğrenme platformları, dijital tahtalar, simülasyonlar ve eğitim içeriklerini paylaşma araçlarını kullanmaktadır. Dijital teknoloji, bilgiye erişimi genişletmekte, evden öğrenmeyi mümkün kılmakta ve çoklu ortam içerikleri sayesinde dersleri daha dinamik ve etkileşimli hâle getirmektedir. Böylece öğrenciler öğrenme sürecine daha fazla katılım gösterir ve öğrenmeye daha iyi odaklanırlar. Dijital öğrenci bilgi sistemleri aracılığıyla veliler, çocuklarının öğrenme sürecindeki gelişimini takip edebilirler.



Resim 2 Dijital teknolojinin eğitimdeki rolü



Sağlık alanında sağlık kuruluşları, hastaların elektronik kayıtlarını, dijital tanı aygıtlarını ve uzaktan sağlık danışmanlığı hizmetlerini kullanmaktadır. Hastalar çevrim içi olarak randevu alabilir, sonuçlarına internet üzerinden ulaşabilir ve sağlık durumlarını takip etmek için mobil uygulamalardan yararlanabilirler. Yapay zekâ, tıbbi verilerin analizinde kullanılırken robotik sistemler cerrahi işlemlerde veya bakım hizmetlerinde kullanılmaktadır.

Neredeyse her iş yerinde dijital araçlar, iletişim, ekip çalışması, görevlerin düzenlenmesi ve süreçlerin otomatikleştirilmesini sağlamaktadır. Uzaktan çalışma, belgelerin bulut üzerinden paylaşılması ve video konferans sistemleri kurumlarda ve şirketlerde giderek daha yaygın kullanılmaktadır. Dijitalleşme verimliliği artırırken çalışanlardan siber güvenlik, veri analizi ve dijital iletişim gibi yeni beceriler de talep etmektedir.

Ulaşımında dijital teknoloji, navigasyon, trafik yönetimi, araç takip sistemleri ve otomatik ödeme sistemlerinde kullanılmaktadır. Toplu taşıma için geliştirilen akıllı uygulamalar, seferlerin gerçek zamanlı olarak takip edilmesini sağlarken navigasyon sistemleri güzergâhları optimize eder ve yolculuk süresini azaltır. Sensörler, kameralar ve yapay zekâ algoritmaları kullanan otonom araçlar günümüzde yollarda test edilmekte ve geleceğin ulaşım teknolojilerini oluşturmaktadır.

Ticarette internet üzerinden alışveriş ve ödeme işlemleri sürekli olarak artmaktadır. Elektronik ticaret, mobil bankacılık ve dijital cüzdanlar, her yerden ve her zaman nakitsiz işlem yapılmasını sağlamaktadır. Mağazalar, veri analizi sayesinde tüketici alışkanlıkları hakkında bilgi edinmekte ve ürün tekliflerini müşterilerin ilgi alanlarına göre düzenlemektedir. Aynı zamanda yapay zekâyâ dayalı öneri algoritmaları, müşterinin neyi ve nasıl satın aldığı üzerinde etkili olmakta, bu durum etik ve gizlilik konularında yeni sorular ortaya çıkarmaktadır.

Kamu yönetiminde hizmetler giderek daha fazla dijital ortamda sunulmaktadır. Elektronik randevu alma, faturaların çevrim içi ödenmesi ve belgelere internet üzerinden erişim, vatandaşlarla kurumlar arasındaki iletişimi daha hızlı ve verimli hâle getirmektedir. Dijital teknoloji, fiziksel olarak bulunma gereksinimini azaltmakta, bürokratik işlemleri kısaltmakta ve bilgilere daha şeffaf erişim sağlamaktadır. Bununla birlikte bu değişimler, vatandaşların dijital okuryazarlığının geliştirilmesini ve kişisel verilerin korunması için güvenlik önlemlerinin alınmasını gerektirmektedir.

Dijital teknolojinin olumsuz etkileri de bulunmaktadır ve bunlar göz ardı edilmemelidir. Ekranların aşırı kullanımı bağımlılığa yol açabilmekte, uyku düzeninin bozulmasına ve sosyal izolasyona neden olabilmektedir. Kişisel verilerin kötüye kullanılması ve siber saldırılar nedeniyle gizlilik ve güvenlik risk altına girmektedir. Ayrıca teknolojiye erişimi olmayan kişilerin bilgiye ve eğitime ulaşmada dezavantaj yaşadığı dijital eşitsizlik tehlikesi de bulunmaktadır. Bu nedenlerle dijital teknoloji bilinçli, sorumlu ve eleştirel bir yaklaşımla kullanılmalıdır.

Dijital teknoloji günlük yaşamın ayrılmaz bir parçası hâline gelmekte, daha hızlı iletişim, kaynakların daha akıllı yönetimi, hizmetlere daha kolay erişim ve yeni öğrenme ve çalışma biçimleri sağlamaktadır. Teknolojinin kullanımı önemli faydalar sunarken aynı zamanda etik, güvenlik ve sağlıkla ilgili yeni sorunları da beraberinde getirmektedir. Bu konuların anlaşılması, teknolojinin insan ve toplum yararına bilinçli, sorumlu ve etkili bir şekilde kullanılmasını sağlar.

ÖRNEK 1



Uygulamada E-öğrenme

Bir okulda öğretmen, öğrencilerin materyallere ulaştığı, testleri çözdüğü ve gerçek zamanlı geri bildirim aldığı bir dijital öğrenme platformu kullanmaktadır.

Bu platform sayesinde veliler de çocuklarının gelişimini takip edebilir, öğretmen ise sonuçlarla ilgili otomatik raporlar alabilir.

ÖRNEK 2



Akıllı Ev

Akıllı evde aydınlatma, sıcaklık ve güvenlik sistemi mobil uygulama aracılığıyla kontrol edilmektedir. Evde kimse olmadığında sistem otomatik olarak ısıtmayı azaltır, alarmı etkinleştirir ve olağan dışı hareketler algılandığında bildirim gönderir.

ÖRNEK 3



Tele-tıp

Kronik bir hastalığı olan hasta, kan basıncını ölçer ve sonucu doğrudan aile hekimine gönderen bir uygulama kullanmaktadır. Değerlerde izin verilen sınırların dışında değişiklik olduğunda sistem otomatik olarak doktoru uyarır ve çevrim içi görüşme randevusu oluşturur.

BİLGİ KONTROLÜ SORULARI VE ETKİNLİKLER



1. Dijital teknolojinin evde kullanıldığı üç alanı açıklayın!
2. Eğitimde dijital teknoloji kullanımına örnekler verin!
3. Dijital teknolojinin sağlık alanına nasıl katkı sağladığını açıklayın!
4. Tele-tıp nedir? Bir örnekle açıklayın!
5. Cep telefonu kullanımının olumlu ve olumsuz etkilerini yazın!
6. Dijital aygıtlara aşırı bağımlılığın hangi tehlikeleri ortaya çıkar?
7. Dijital eşitsizlik ne anlama gelir? Nasıl azaltılabilir?
8. Ülkemizde kamu yönetiminde dijital teknolojinin nasıl kullanıldığını araştırın!



UYGULAMALI ETKİNLİK

Bir okulda BT yöneticisi olduğunuzu düşün. AŞAĞIDAKİ durumların her biri için uygun bir bilgisayar sistemi seçme görevi sana verilmiştir. Her durum için bu sistemi neden seçtiğinizi açıkla.

Senaryo 1: Okul Yönetiminde Ofis Çalışmaları

- Belge işleme, e-posta kullanımı ve öğrencilerin verilerini yönetmek için bir sisteme ihtiyacın var.

Senaryo 2: Okulun Multimedya Kulübü için Video Düzenleme ve Grafik Tasarım

- Yüksek çözünürlüklü çoklu ortam içeriklerini işleyebilecek bir sisteme ihtiyacın var.

Senaryo 3: Programlama ve Yapay Zekânın Temellerinin Öğrenildiği Sınıf

- Yazılım geliştirmeyi, büyük verilerle çalışmayı ve makine öğrenmesi modellerini destekleyen bir sisteme ihtiyacın var.

Senaryo 4: Okul ziyaretçileri için bilgilendirme noktası

- Sürekli çalışan, kullanımı kolay ve klavye veya fare gerektirmeyen bir sisteme ihtiyacın var.

Her senaryo için AŞAĞIDAKİ bilgisayar sistemi türlerinden birini seç: Masaüstü bilgisayar, dizüstü bilgisayar, sunucu, iş istasyonu, gömülü sistem (embedded system), tablet/etkileşimli ekran.



SONUÇ

Dijital teknoloji, günlük yaşamın vazgeçilmez bir parçasıdır ve öğrenme, çalışma ve iletişim kurma biçimlerimizi önemli ölçüde şekillendirmektedir. Daha hızlı bilgi erişimi, kolay iletişim ve çeşitli etkinliklerin daha verimli gerçekleştirilmesini sağlamaktadır. Bununla birlikte, artan verimlilik, yaratıcılık ve öğrenmeyi kolaylaştırma gibi birçok avantajının yanında bazı dezavantajları da bulunmaktadır. Bu dezavantajlar genellikle gizlilik riskleri, teknolojinin aşırı kullanımı, yanlış bilgilere maruz kalma ve teknoloji bağımlılığı ile ilgilidir. Bu nedenle dijital teknolojinin rolü, sağladığı yararlar ve olası tehlikeleri hakkında eleştirel bir anlayış geliştirmek önemlidir. Bu bilinç, teknolojinin sorumlu, güvenli ve bilinçli bir şekilde kullanılmasına yardımcı olur. Böylece bireyler, çağdaş dijital toplumda yaşam ve çalışma için gerekli dijital becerilerini ve yeterliliklerini güçlendirebilirler.



DAHA FAZLA BİLGİ EDİNMEK İSTEYENLER İÇİN

Modern dijital teknolojiler arasında sanal gerçeklik (VR, Virtual Reality) ve artırılmış gerçeklik (AR, Augmented Reality) de yer almaktadır. Bu teknolojiler etkileşimli dijital deneyimler oluşturur, ancak farklı şekillerde çalışırlar. Amaçları, gerçek dünyayı bilgisayar tarafından oluşturulan bir ortamla tamamen veya kısmen değiştirmektir.

Sanal gerçeklik (VR), çeşitli etkinliklerin simülasyonudur ve çalışması için AŞAĞIDAKİ araçlara ihtiyaç vardır:

- **VR gözlüğü/kulaklığı (Headset)** – kullanıcının gözleri önünde 3B görüntüler gösterir.
- **Hareket sensörleri** – baş ve vücut hareketlerini takip eder.
- **Kontrol cihazları** – sanal dünyayla etkileşim kurulmasını sağlar.
- **Ses sistemi** – gerçekçi bir deneyim için 3B ses sağlar.

VR, gözlüğü taktığında hareketlerine tepki veren dijital bir dünyayı görmek şeklinde çalışır. Başını çevirdiğinde veya ellerini hareket ettirdiğinde bu hareketler takip edilir ve gerçek zamanlı olarak görüntülenir. Programlar, gerçekçi 3B ortamlar oluşturmak için grafik motorlarını (örneğin Unity veya Unreal Engine) kullanır.

Artırılmış gerçeklik (Augmented Reality), gerçek dünyayı dijital öğelerle (görüntüler, metinler, 3B nesneler) zenginleştirir. Gerçek dünyaya entegre edilen metin, görüntü, ses veya diğer sanal öğeler gibi gerçek zamanlı bilgileri kullanır ve bu yönüyle sanal gerçeklikten ayrılır. AR, simülasyonların aksine kullanıcının gerçek dünyayla etkileşimini destekler ve bu etkileşime değer katar. AR için kamera ve ekran (çoğunlukla cep telefonu, tablet veya AR gözlükleri), çevrenin ve konumun tanınması için sensörler ve GPS, dijital öğeleri gerçek dünyaya "yerleştirmek" için ise nesne/alan tanıma yazılımı gereklidir. Kamera gerçek dünyayı görüntüler, yazılım ortamı analiz eder ve dijital nesneleri ekler (örneğin, odadaki yeni bir çalışma masasının 3B modeli). Kullanıcı ekranda gerçek görüntü ile dijital görüntünün birleşimini görür.

Dijital teknolojinin "akıllı şehirlerde" nasıl kullanıldığını araştırın. Trafik yönetimi, kamu güvenliği, atık yönetimi ve enerji alanlarında hangi teknolojiler uygulanmaktadır? Büyük dünya şehirlerinin kentsel yaşamı iyileştirmek için yapay zekâyı nasıl kullandığını inceleyin. VR ve AR teknolojileri şehirlerde yaşam kalitesinin artırılmasına nasıl yardımcı olabilir? Bu teknolojiler kırsal bölgelerde de kullanım alanı bulabilir mi?

Dijital teknolojinin bilgisayar sistemiyle ilişkisi nedir?

Dijital teknoloji, bilgisayar sistemlerine dayanır. Yapay zekânın (YZ), dijital öğrenmenin, otomasyonun ve veri analizinin çalışabilmesi için yapay zekâ teknolojisini çalıştıran bilgisayar sistemlerine ihtiyaç vardır. Örneğin eğitimde yapay zekâ, sonuçları analiz etmek, örüntüleri tanımak ve öğretimi öğrencinin ihtiyaçlarına göre uyarlamak için algoritmalar kullanan bir bilgisayar sistemi aracılığıyla çalışır. Dijital teknoloji, bilgisayar sistemlerinin işlevlerini genişletir. Yapay zekâ ve makine öğrenmesi sayesinde bilgisayar sistemleri yalnızca verileri işlemekle kalmaz, aynı zamanda kendi kendine öğrenme, örüntü tanıma ve otomatik karar verme (örneğin bir eğitim kurumuna aday kabul etme) gibi yeni işlevler de kazanır. Bilgisayar sistemleri, dijital dönüşümün temel platformunu oluşturur. Eğitim sistemleri, yönetim, ilerlemenin izlenmesi, kişiselleştirilmiş öğretim ve benzeri uygulamalar için bilgisayar sistemlerini dijital teknolojinin altyapısı olarak kullanır.

Bilgisayar sistemi, donanım (hardware) ve yazılımdan (software) oluşur. Donanım, kasa, monitör, klavye, fare, sabit disk, kulaklık, mikrofon, yazıcı ve projektör gibi fiziksel bileşenleri içerir. Yazılım ise donanımı kontrol eden ve görevlerin gerçekleştirilmesini sağlayan programlar bütünüdür. Giriş aygıtları (örneğin klavye ve fare) veri girişini sağlarken, çıkış aygıtları (örneğin monitör ve yazıcı) sonuçları görüntüler. Tüm bu bileşenler farklı işlevleri yerine getirir ve farklı şekillerde çalışır.

Kasada, merkezi işlem birimi (CPU - Central Processing Unit) bulunur. **CPU**, kendi bileşenleri aracılığıyla komutları yerine getirir ve sistemdeki tüm işlemleri yönetir. Programlardaki komutları okuyarak, işleyerek ve uygulayarak çalışır, hesaplamalar yapar ve kararlar verir. CPU şu bileşenlerden oluşur:

- **ALU (Aritmetik Mantık Birimi):** Matematiksel ve mantıksal işlemleri gerçekleştirir.
- **CU (Kontrol Birimi):** Veri ve komut akışını yönetir.

Merkezi işlem biriminin (CPU) işlediği veriler, geçici olarak RAM (Random Access Memory) adı verilen çalışma belleğinde tutulur. RAM, o anda kullanılmakta olan verileri ve programları geçici olarak depolar. Çalışma şekli şöyledir: Bir uygulama açıldığında, ona daha hızlı erişilebilmesi için RAM'e yüklenir. Bilgisayar kapatıldığında RAM'deki tüm veriler silinir.

Veriler kalıcı olarak kalıcı bellek (HDD/SSD) üzerinde saklanır. Bu bellekte belgeler, programlar ve görseller gibi farklı türdeki veriler uzun süreli olarak depolanır. HDD ile SSD arasında fark vardır. HDD (Hard Disk Drive) mekanik yapıya sahip, daha yavaş ancak daha ucuz bir depolama birimidir. SSD (Solid State Drive) ise hareketli parçalara sahip değildir, daha hızlı çalışır ancak daha pahalıdır.

Bu yapı, 1945 yılında Von Neumann tarafından önerilen ve bir bilgisayarın nasıl çalıştığını açıklayan bilgisayar mimarisinin temelini oluşturur.

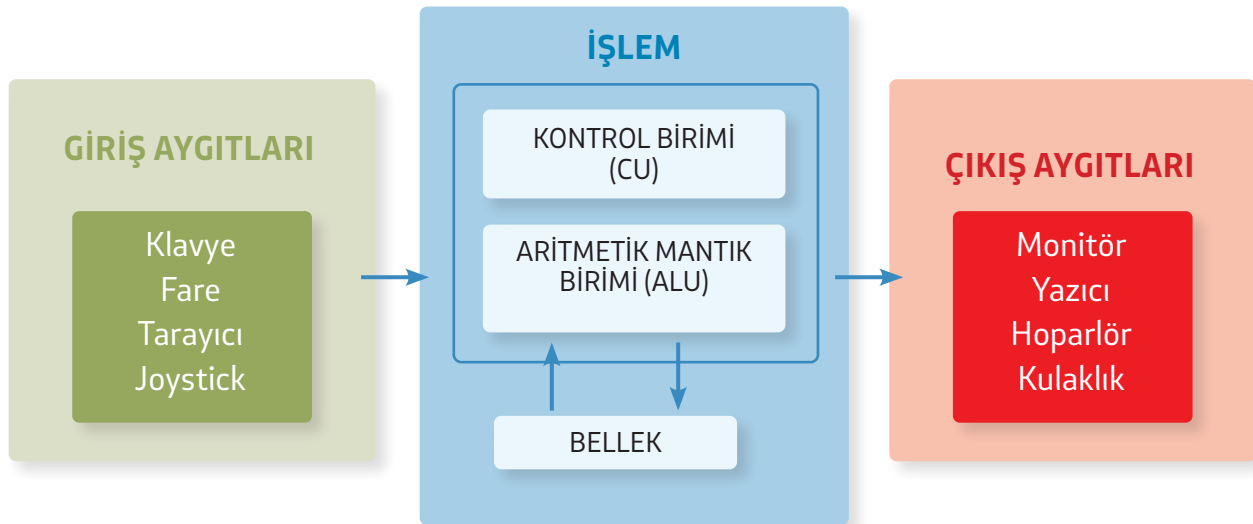
1.3

Von Neumann Mimarisi ve Bilgisayarın Bileşenleri

Von Neumann mimarisi, modern bilgisayar sistemlerinin temelini oluşturan bir yapıdır. Bu kavram, 20. yüzyılın ortalarında Macar asıllı Amerikalı matematikçi John von Neumann tarafından ortaya konulmuştur. Bu modele göre bilgisayar, belleğinde önceden depolanmış komutları yerine getiren bir aygıt olarak tanımlanır. Von Neumann mimarisi, işlemci, bellek, giriş ve çıkış aygıtları gibi bilgisayarın temel bileşenlerinin nasıl çalıştığını ve bunların iletişim kanalları aracılığıyla birbirleriyle nasıl etkileşim kurduğunu açıklar. Teknoloji büyük ölçüde gelişmiş olsa da bu modelde tanımlanan temel ilkeler günümüz bilgisayarlarında hâlâ uygulanmaktadır.

Von Neumann mimarisi birkaç temel bileşene dayanır: merkezi işlem birimi (CPU), ana bellek, giriş ve çıkış aygıtları ile veri yolu (bus) olarak adlandırılan iletişim yolları. İşlemci, tüm sistemin çalışmasını denetler ve iki temel alt birimden oluşur: Aritmetik Mantık Birimi (ALU) ve Kontrol Birimi (CU). Aritmetik Mantık Birimi hesaplama ve mantıksal işlemleri gerçekleştirirken, Kontrol Birimi komutları yorumlar ve sistemdeki tüm işlemleri koordine eder.

Ana bellek, o anda yürütülmekte olan işlemler için gerekli olan verileri ve komutları saklamak amacıyla kullanılır. Bu bellek geçicidir (uçucu bellek), yani bilgisayar kapatıldığında içeriği silinir. Çalıştırılmakta olan programlar ve hesaplama sonuçları da bu bellekte tutulur. Bellek, adreslere göre düzenlenmiştir ve her veri, kendine özgü bir adres yardımıyla bulunur.



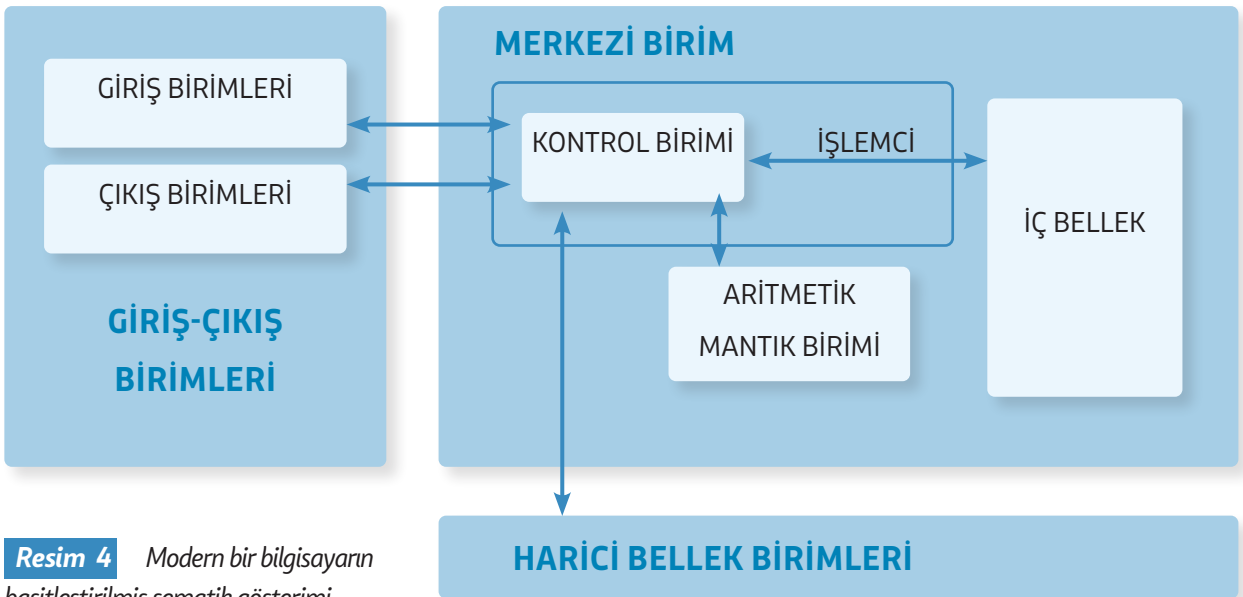
Resim 3

Von Neumann bilgisayar sistem modelinin basitleştirilmiş şematik gösterimi

Von Neumann mimarisinin temel ilkelerinden biri, hem verilerin hem de komutların aynı bellekte saklanmasıdır. Bu yaklaşım daha fazla esneklik sağlar, çünkü komutlar çalışma sırasında değiştirilebilir. Ancak aynı zamanda, kod ve veriler arasında açık bir ayrım olmadığında güvenlik açısından potansiyel bir risk oluşturabilir. Bu model, programlamayı mümkün hâle getirmiştir, çünkü programlar diğer veriler gibi yazılabilir, aktarılabilir, saklanabilir ve işlenebilir. Ancak komutların sırayla alınması ve yürütülmesi, işlemcinin aynı anda hem komut hem de veri alamadığı bir darboğaz oluşturur. Bu durum Von Neumann darboğazı olarak adlandırılır. Bu sorunu azaltmak için modern işlemciler, performansı artırmaya yönelik işlem hattı (pipeline) adı verilen bir teknik kullanır. Bu teknikte komutların yürütülmesi, alma, kod çözme, yürütme, belleğe erişim ve yazma gibi ardışık aşamalara ayrılır ve farklı komutlar aynı anda farklı aşamalarda işlenir. Böylece birden fazla komut aynı anda farklı işlem aşamalarında yürütülebilir ve bilgisayarın genel işlem hızı önemli ölçüde artırılır.

Giriş aygıtları, bilgisayara bilgi girilmesini sağlar. Bunlara klavye, fare, mikrofon ve tarayıcı örnek olarak verilebilir. Çıkış aygıtları ise işlenen sonuçların görüntülenmesi veya aktarılması için kullanılır, monitör, yazıcı ve hoparlör bunlara örnektir. Aynı zamanda hem giriş hem de çıkış işlevi gören birleşik aygıtlar da vardır. Örneğin dokunmatik ekranlar ve ağ bağdaştırıcıları bu tür aygıtlardır.

Veri yolları (bus), sistemin tüm bileşenlerini birbirine bağlayan elektronik iletişim yollarıdır. Üç temel veri yolu türü vardır: adres veri yolu (bellek konumlarını belirlemek için), veri veri yolu (verilerin aktarımı için) ve kontrol veri yolu (kontrol sinyallerinin iletimi için). Bu veri yolları, işlemci, bellek ve çevre birimleri arasında hızlı ve eş zamanlı iletişimi sağlayarak sistemin düzgün çalışmasını destekler.





Resim 5 Giriş birimleri

Merkezi işlem birimi (CPU), bilgisayarın beyni gibi çalışır ve hızı büyük ölçüde sistemin genel performansını belirler. Modern işlemciler milyarlarca transistör içerir ve bilgilerin paralel olarak işlenmesini sağlayan birden fazla çekirdek (core) kullanır. Ancak Von Neumann tarafından tanımlanan komutların sıralı olarak yürütülmesi temel mantığı günümüzde hâlâ geçerliliğini korumaktadır.

Ana bellek gigabayt (GB) cinsinden ölçülür. Daha hızlı işlem yapmak için işlemcilerin içinde bulunan önbellek (cache) belleği de kullanılır. Veriler, işlemci ile bellek arasında çok yüksek hızlarda hareket eder, ancak bu hız, modern bilgisayarların temel sınırlamalarından biri olan "Von Neumann darboğazı" (Von Neumann bottleneck) olarak bilinen duruma neden olur. Bu terim, bellek ile işlemci arasındaki iletişimin sınırlı aktarım kapasitesini ifade eder ve sistemin toplam hızını yavaşlatır.

Von Neumann mimarisi, tüm modern dijital bilgisayarların temelini oluşturur. Teknoloji büyük ölçüde gelişmiş olsa da temel ilkeler değişmemiştir: Komutlar ve veriler aynı bellekte saklanır, işlemci bunları adım adım yürütür ve tüm bileşenler iletişim yolları (veri yolları) aracılığıyla birlikte çalışır. Bu mimariyi anlamak, bilgisayarların nasıl çalıştığını daha iyi kavramayı sağlar ve bilişim alanında daha ileri öğrenmeler için temel oluşturur.



SONUÇ

Von Neumann mimarisine göre bilgisayar sisteminin nasıl çalıştığını anlamak, bilgisayarın verileri nasıl aldığını, işlediğini ve sakladığını kavramanı sağlar. İşlemcinin, belleğin ve giriş-çıkış aygıtlarının görevlerini açıklayarak temel ancak önemli bir dijital okuryazarlık geliştirirsin.

Aynı zamanda farklı bilgisayar sistemlerini karşılaştırmak ve giriş-çıkış aygıtlarını doğru şekilde sınıflandırmak, bilgisayarın farklı bölümlerinin nasıl birlikte çalıştığını daha iyi anlamana yardımcı olur. Böylece uygun donanımı seçebilir, aygıtlar arasındaki farkları tanıyabilir ve teknolojiyi günlük görevlerde daha verimli kullanabilirsin. Bu bilgiler yalnızca bilişim alanında daha ileri öğrenmelere hazırlık sağlamakla kalmaz, aynı zamanda seni modern teknolojinin bilinçli ve yetkin bir kullanıcısı olmaya teşvik eder.

ÖRNEK 1



İşlemci komutu nasıl yürütür?

İki sayının toplama işlemi yapılırken, komut önce bellekten okunur. Daha sonra işlemcinin kontrol birimi tarafından yorumlanır ve ardından Aritmetik Mantık Birimi (ALU) tarafından gerçekleştirilir. ALU hesaplama işlemi yaparak sonucu oluşturur.

ÖRNEK 2



Aygıtların Veri Yolları Aracılığıyla Birlikte Çalışması

Kullanıcı klavyede bir tuşa bastığında, sinyal giriş aygıtı üzerinden gönderilir ve veri yolu aracılığıyla işlemciye iletilir. İşlemci bu bilgiyi işler ve ardından sonuç bir çıkış aygıtına gönderilebilir, örneğin ekranda görüntülenebilir.

ÖRNEK 3



Önbellek (cache) belleği neden vardır?

İşlemci, ana bellekten daha hızlı çalışır. İşlemcinin beklemesini önlemek için sık kullanılan verileri saklayan önbellek (cache) belleği kullanılır. Önbellek daha hızlıdır ancak kapasitesi çok daha küçüktür ve komutların daha verimli çalışmasını sağlayarak bilgisayarın performansını artırır.

BİLGİ KONTROLÜ SORULARI VE ETKİNLİKLER



1. Bir bilgisayar sisteminde CPU'nun rolünü açıklayınız!
2. ALU nedir ve görevi nedir?
3. RAM ile önbellek (cache) belleği arasındaki farkı açıklayınız!
4. Üç veri yolu (bus) türü nelerdir ve ne işe yararlar?
5. Von Neumann mimarisine göre bir komutun nasıl yürütüldüğünü adım adım yazınız!
6. "Von Neumann darboğazı" teriminin ne anlama geldiğini araştırınız!
7. Verilerin ve komutların bellekte saklanması neden önemlidir?
8. Giriş ve çıkış aygıtlarını günlük hayattan örnekler vererek karşılaştırınız!



DAHA FAZLA BİLGİ EDİNMEK İSTEYENLER İÇİN

Von Neumann mimarisine alternatiflerin olup olmadığını ve bazı araştırmacıların neden yeni modeller önerdiğini araştır! "Paralel işlemeyi" ne ifade ettiğini ve modern işlemcilerde neden birden fazla çekirdeğin kullanıldığını incele!

Grafik kartının (GPU – Graphics Processing Unit) rolü nedir?

Anakart (Motherboard) nedir? Bu cihazlar nasıl veri alır?



1.4

Giriş ve Çıkış Aygıtları – Sınıflandırma ve Karşılaştırma

Her bilgisayar sisteminde giriş ve çıkış aygıtları, kullanıcı ile bilgisayar arasında bilgi alışverişinde önemli bir rol oynar. Giriş aygıtları aracılığıyla veriler sisteme girilirken, çıkış aygıtları işleme sonuçlarını kullanıcıya sunar. Bu aygıtlar, bilgisayarla etkileşim kurulmasını sağlar ve metin ve resimlerle çalışmaktan iletişim kurmaya ve oyun oynamaya kadar günlük görevlerin yerine getirilmesi için gereklidir. Rollerinin doğru anlaşılabilmesi için bu aygıtların işlevlerine, işledikleri veri türüne ve farklı alanlardaki kullanım amaçlarına göre sınıflandırılması gerekir.

Giriş aygıtları, bilgisayara veri girişi yapmak için kullanılır. En yaygın kullanılan giriş aygıtları klavye, fare, tarayıcı, mikrofon ve kameradır. Klavye alfasayısal verilerin girilmesini sağlar, fare işaretleme ve seçim yapmaya olanak tanır, mikrofon ise daha sonra işlenecek ses sinyallerini bilgisayara aktarır. Tarayıcılar resim veya belgelerin bilgisayara aktarılmasını sağlarken, dijital kameralar görsel bilgileri bilgisayarın işleyebileceği verilere dönüştürür.



Resim 6 Giriş ve çıkış aygıtları

Çıkış aygıtları, işlemenin sonuçlarını kullanıcıya sunmak için kullanılır. En yaygın çıkış aygıtları arasında monitörler, yazıcılar, hoparlörler ve projektörler yer alır. Monitörler verileri ekranda görsel olarak görüntüler, yazıcılar dijital bilgileri fiziksel kopyalara dönüştürür, hoparlörler ise ses çıkışı sağlar. Her çıkış aygıtı belirli bir bilgi türü için tasarlanmıştır ve kalite, hız ile doğruluk özellikleri kullanıcıların ve uygulamaların ihtiyaçlarına göre uyarlanır.

Bazı aygıtlar aynı anda hem giriş hem de çıkış aygıtı olarak çalışabilir. Bu tür aygıtlara giriş-çıkış (I/O) aygıtları denir. Örneğin, dokunmatik ekran, dokunma yoluyla veri girişine olanak sağlarken aynı zamanda bilgileri ekranda görüntüler. Modemler ve ağ kartları ise bilgisayar ağları üzerinden veri alışverişi için kullanılır ve bunlar da I/O aygıtları olarak kabul edilir.



Resim 7 Giriş ve çıkış aygıtları

Giriş ve çıkış aygıtları farklı ölçütlere göre sınıflandırılabilir: işledikleri veri türüne (metin, görüntü, ses), kullanım şekline (manuel, otomatik), giriş/çıkış hızına ve doğruluk düzeyine göre. Endüstriyel ve tıbbi uygulamalarda aygıtlardan yüksek hassasiyet ve dayanıklılık beklenirken, ev ortamında kullanım kolaylığı ve kullanıcı konforu daha önemli özelliklerdir.

Modern giriş aygıtları, daha yüksek etkileşim ve hassasiyet sağlamak için gelişmiş teknolojiler kullanmaktadır. Örneğin, grafik tabletler sanatsal çalışmalar için basınca duyarlı dijital kalem kullanırken, parmak izi tarayıcıları veya yüz tanıma sistemleri gibi biyometrik aygıtlar güvenli kimlik doğrulaması sağlar. Sanal gerçeklikte ise eldivenler ve başlıklar gibi giriş aygıtları hareket ve el hareketlerine ilişkin verileri toplar, bu veriler daha sonra dijital dünyada gezinme ve etkileşim amacıyla kullanılır.

Çıkış aygıtlarında ise gelişmeler daha yüksek çözünürlük, daha fazla hız ve daha kaliteli çıktı elde edilmesi yönündedir. Modern monitörler yüksek çözünürlük ve ultra geniş ekran desteği sunarken, 3D yazıcılar karmaşık nesnelerin fiziksel olarak üretilmesini sağlar. Bu teknolojiler yalnızca sanayide değil, tıp, mimarlık ve sanatta da kullanılmaktadır, çünkü bu alanlarda görsel ve fiziksel sunum büyük önem taşır.

Bazı sistemlerde, sensör aygıtları ayrı bir kategori oluşturur. Bu aygıtlar ışık, ses, hareket ve sıcaklık gibi fiziksel olayları algılar ve bunları dijital verilere dönüştürür. Sensörler akıllı evlerde, güvenlik sistemlerinde ve endüstriyel otomasyonda kullanılır. Bu sistemlerde çıkış aygıtları sesli alarmlar, aydınlatma sistemleri veya mobil uygulama üzerinden gönderilen bildirimler olabilir. Böylece etkin kontrol ve uzaktan izleme sağlanır.



Resim 8 Sensör aygıtları

Donanım özelliklerinin yanı sıra, bu aygıtları yöneten sürücü yazılımları (driver) da büyük önem taşır. Sürücüler, işletim sisteminin aygıtın işlevini tanımasını ve onunla iletişim kurmasını sağlar. Uygun yazılım olmadan donanım düzgün çalışmaz. Bu nedenle aygıtlar ile onları kullanan programlar arasında soyutlama katmanı (abstraction layer) adı verilen bir katman bulunur.

Giriş ve çıkış aygıtları, bilgisayar sistemiyle etkileşim kurulmasını sağlayan temel bileşenlerdir. Bu aygıtlar, verilerin farklı biçimlerde girilmesini, işlenmesini ve görüntülenmesini mümkün kılar. Türlerinin, işlevlerinin ve sınıflandırma ölçütlerinin anlaşılması, bilişim alanındaki sonraki öğrenmeler için temel oluşturur. Teknolojinin gelişmesiyle birlikte bu aygıtlar da sürekli gelişmektedir, günlük ve mesleki yaşamda yeni iletişim, otomasyon ve görselleştirme olanakları sunmaktadır.



SONUÇ

Giriş aygıtları, verilerin ve komutların bilgisayara girilmesini sağlar, çıkış aygıtları ise işlemin sonuçlarını kullanıcıya sunar.

Giriş Aygıtları (Input Devices)

Bilgisayara veri girilmesini sağlayan aygıtlar:

- **Klavye** – metin ve komut girişi,
- **Fare** – işaretleme ve seçenekleri seçme,
- **Tarayıcı** – kâğıt üzerindeki belgeleri dijital ortama aktarma,
- **Mikrofon** – ses girişi,
- **Web kamerası** – video girişi,
- **Grafik tablet (graphics tablet)** – dijital çizim,
- **Dokunmatik ekran (giriş aygıtı olarak)** – dokunarak veri girişi,
- **Joystick/Gamepad** – oyunlarda kontrol sağlama,
- **Barkod okuyucu** – barkodları okuma,
- **Parmak izi tarayıcısı (Fingerprint scanner)** – biyometrik kimlik doğrulama.





Çıkış Aygıtları (Output Devices)

Bilgisayardan verileri görüntüleyen veya ileten aygıtlar:

- **Monitör** – metin ve görüntülerin görsel olarak gösterilmesi,
- **Yazıcı** – belge ve fotoğrafların yazdırılması,
- **Hoparlör** – ses çıkışı,
- **Kulaklık** – ses çıkışı,
- **Projeksiyon cihazı** – görüntünün duvara veya bir yüzeye yansıtılması,
- **Plotter** – teknik çizimlerin basılması,
- **Dokunmatik ekran** (çıkış aygıtı olarak) – görüntünün gösterilmesi,
- **LED göstergeler** – ışıklı uyarı sinyalleri.

Bu şekilde yapılan sınıflandırma, bilgisayar sistemlerinin çalışma yapısını daha iyi anlamayı sağlar, belirli bir görev için uygun donanımın bilinçli olarak seçilmesine yardımcı olur ve bireyin dijital teknolojiyi daha becerikli, güvenli ve bağımsız bir şekilde kullanmasına katkı sağlar.

ÖRNEK 1



Belge Tarama

Bir belge tarayıcıya yerleştirildiğinde, aygıt metni veya görüntüyü tarar ve dijital biçime dönüştürür. Daha sonra bu dijital veri ekranda görüntülenir. Bu işlem, bir giriş aygıtının nasıl çalıştığını gösterir.

ÖRNEK 2



Rapor Yazdırma

Tablolu hesaplama programında veri analizi yapıldıktan sonra, kullanıcı raporu yazıcıya gönderir. Çıkış aygıtı, dijital belgeyi fiziksel bir kâğıt kopyaya dönüştürür.

ÖRNEK 3



Dokunmatik Ekran Kullanımı

ATM’de kullanıcı, dokunmatik ekran aracılığıyla veri girer ve aynı zamanda görsel ve metinsel geri bildirimler alır. Bu aygıt aynı anda hem giriş hem de çıkış aygıtı olarak çalışır.

BİLGİ KONTROLÜ SORULARI VE ETKİNLİKLER

1. Günlük hayatta kullanılan üç giriş aygıtı belirtiniz ve işlevlerini açıklayınız!
2. En yaygın çıkış aygıtları nelerdir ve ne amaçla kullanılırlar?
3. Aynı anda hem giriş hem de çıkış aygıtı olan bir aygıtın nasıl çalıştığını açıklayınız!
4. Üç giriş ve üç çıkış aygıtını, veri türü, doğruluk ve hız açısından karşılaştıracığınız bir tablo hazırlayınız!
5. Grafik tabletin ne olduğunu ve hangi amaçlarla kullanıldığını araştırınız!
6. Aygıtların doğru çalışması için sürücülerin (driver) neden önemli olduğunu açıklayınız!
7. Biyometrik aygıtın ne olduğunu tanımlayınız ve örnekler veriniz!
8. Sensör aygıtlarının akıllı evlerde nasıl çalıştığını ve hangi bilgileri toplayabildiklerini yazınız!



DAHA FAZLA BİLGİ EDİNMEK İSTEYENLER İÇİN

Giriş ve çıkış aygıtlarının tıp, mimarlık veya oyun gibi özel meslek alanlarında nasıl kullanıldığını araştırınız! Sanal gerçeklik ile çalışmada kullanılan aygıtların neler olduğunu ve bu aygıtların dijital ortamda gerçek etkileşimi nasıl oluşturduğunu inceleyiniz.



Resim 9

Giriş ve çıkış cihazları.

1.5

Modern Dijital Teknolojiler: Dokunma, Holografi, 3D Projeksiyon

Modern dijital teknoloji sürekli gelişerek yeni etkileşim, görselleştirme ve bilgi yönetimi yöntemlerini ortaya çıkarmaktadır. En dikkat çekici yenilikler arasında dokunma, holografi ve üç boyutlu projeksiyon teknolojilerine dayanan teknolojiler yer almaktadır. Bu teknolojiler günümüzde günlük kullanım için kullanılan cihazlarda kullanılmakta, ancak potansiyelleri tıp, eğitim, mimarlık ve eğlence endüstrisi gibi alanlara da yayılmaktadır. Bu teknolojiler, insanların cihazlarla ve dijital içeriklerle iletişim kurma biçimini değiştirerek daha gerçekçi, daha dinamik ve daha doğal kullanıcı deneyimleri sunmaktadır.

Dokunmatik teknoloji, kullanıcının genellikle bir ekran olan dijital yüzeyle fiziksel temas yoluyla doğrudan iletişim kurabilme olanağına dayanır. En yaygın olarak, parmak veya kalem dokunuşuna tepki veren kapasitif ve rezistif ekranlar kullanılır. Günümüzde dokunmatik ekranlar akıllı telefonlara, tabletlere, ATM'lere, bilgi kiosklarına ve okullardaki etkileşimli tahtalara entegre edilmiştir. Kullanım alanları otomotiv endüstrisi, sağlık sektörü ve endüstriyel kontrol sistemlerine de yayılmaktadır. Bu alanlarda doğrudan ve sezgisel etkileşim, verimliliği önemli ölçüde artırmaktadır.



Resim 10 Dokunmatik Teknolojisi

Dokunmatik ekranlar aynı anda birden fazla dokunma noktasını algılayabilir, buna "multi-touch" teknolojisi adı verilir. Bu teknoloji, iki veya daha fazla parmak kullanılarak yakınlaştırma, döndürme ve nesneleri hareket ettirme gibi işlemlerin yapılmasını sağlar. Bu özellik, görsel işlemlerin önemli olduğu tasarım uygulamaları, oyunlar ve navigasyon gibi kullanıcı arayüzleri için özellikle önemlidir. Eğitimde ise dokunmatik etkileşimli tahtalar, öğrencilerin derse katılımını artırır ve karmaşık bilgilere görsel erişim olanağı sağlar.

Holografi, özel gözlüklere ihtiyaç duyulmadan görülebilen üç boyutlu görüntülerin oluşturulmasını sağlayan bir teknolojidir. Hologram, ışık ışınlarının girişimiyle oluşturulur ve nesnenin derinliği, şekli ve rengi hakkında her açıdan bilgi içerir. Bu görüntüler havada "asılı duruyormuş" gibi görünür ve benzersiz bir görsel deneyim sunar. Holografik projeksiyonlar,

pazarlama, sanat, eğitim ve sunum alanlarında bilgilerin etkileyici bir şekilde görsel olarak aktarılması için kullanılır.

Tıp alanında holografi, anatomik yapıların görselleştirilmesi için kullanılır ve cerrahi işlemlerin daha iyi anlaşılmasına ve planlanmasına olanak sağlar. Müzelerde hologramlar, eserlerin veya tarihî olayların yeniden oluşturulması için kullanılırken, müzik endüstrisinde ise vefat etmiş sanatçıların sanal konserlerinin gerçekleştirilmesinde kullanılmaktadır. Bu teknoloji gelişmeye devam etmekte ve dijital içeriklerle mekân içinde yeni etkileşim biçimleri sunmayı vaat etmektedir.



Resim 11 Alman sirkî hayvanlar için hologram kullanmaktadır

3D projeksiyon, özel projektörler kullanılarak ve bazı durumlarda gözlüklere ihtiyaç duyularak, üç boyutlu nesnelerin düz bir yüzey üzerinde görüntülenmesini sağlar. Sinema endüstrisinde, sanal turlarda, mimari sunumlarda ve simülasyonlarda kullanılır. Eğitimde ise 3D projeksiyonlar, öğrencilerin atomların yapısını, vücudun anatomisini veya yapıların tarihini, klasik yöntemlerle elde edilemeyen görsel derinlikle incelemelerine olanak sağlar.

“Heliodisplay” teknolojisi, fiziksel bir yüzeye ihtiyaç duymadan yapılan projeksiyonun en devrim niteliğindeki örneklerinden biridir. Görüntüler, ortam olarak sis veya buhar kullanılarak havaya yansıtılır. Kullanıcılar projeksiyonlara “dokunabilir” ve onları değiştirebilir, bu durum, bu teknolojiyi bilim kurgudaki holografik ekranlara yaklaştırmaktadır. Bu cihazlar, yüksek düzeyde görsel etki gerektiren sergilerde, sunumlarda ve lüks tanıtımlarda kullanılmaktadır.

Modern dijital teknolojiler olan dokunma, holografi ve 3D projeksiyon, etkileşim ve görselleştirme alanında yeni boyutlar sunmaktadır. Bu teknolojiler kullanıcı deneyimini geliştirir, öğrenmeyi kolaylaştırır ve bilgilerin etkileyici bir şekilde sunulmasını sağlar. Bu teknolojilerin çalışma prensiplerini ve kullanım alanlarını anlamak, bu tür teknolojilerin daha yaygın ve gelişmiş hâle geleceği bir dünyada yaşayacak ve çalışacak genç nesiller için bir gerekliliktir.



SONUÇ

Modern dijital teknolojiler hızla gelişmekte ve yeni iletişim, etkileşim ve görselleştirme yöntemleri oluşturmaktadır. Bu teknolojileri birbirinden ayırt etmek, kullanım alanlarını anlamaya ve piyasada ortaya çıkan yenilikleri takip etmeye yardımcı olur. Bu durum, farklı arayüz türlerini, bilgi giriş ve görüntüleme yöntemlerini ve günlük yaşamın giderek daha fazla parçası hâline gelen gelişmiş görsel teknolojileri ayırt etmeyi sağlar.

Dokunmatik teknoloji, cihazların basit **bir dokunuşla** kontrol edilmesini sağlar. Eski model akıllı telefonlarda, ATM'lerde ve etkileşimli kiosk cihazlarında kullanılmıştır. **Çoklu dokunma teknolojisi (Multi-touch)**, aynı anda birden fazla dokunuşu destekler ve yakınlaştırma, döndürme, kaydırma gibi hareketlerin yapılmasını sağlar. Günümüzde modern akıllı telefonlarda (iPhone, Android), tabletlerde ve etkileşimli tahtalarda kullanılmaktadır. **Dokunmasız teknoloji (Touchless/Gesture-based technology)**, fiziksel temas olmadan, hareketler, jestler veya sensörler kullanılarak cihazların kontrol edilmesini sağlar. Xbox Kinect'te (hareketle kontrol), sensörlü temassız anahtarlarda, sensörlü otomatik kapılarda vb. kullanılmaktadır.

3D görüntü teknolojisi, derinlik içeren ve gerçekçi görünen veya ekrandan "çıkiyormuş" gibi algılanan görüntüler oluşturan teknolojidir. 3D televizyonlar ve filmler, 3D gözlükler (anaglif, polarize, aktif perdeleme vb.), 3D projektörler ve oyunlardaki 3D modelleme alanlarında uygulanmaktadır.

Holografi (Holography), gözlüğe ihtiyaç duyulmadan, farklı açılardan görülebilen üç boyutlu görüntüler oluşturan teknolojidir. Hologram konserleri (Tupac, Whitney Houston), havaalanları ve müzelerdeki hologram asistanları, 3D hologram reklamları gibi alanlarda kullanılmaktadır.

Heliodisplay teknolojisi, görüntüyü havaya yansıtarak "havada asılı duruyormuş" gibi gösteren bir teknolojidir. Kullanıcı, ekran veya fiziksel temas olmadan jestlerle menüler arasında hareket edebilir. Sergilerde, etkileşimli sunumlarda, havada görüntülerin oluşturulduğu pazarlama uygulamalarında vb. kullanılmaktadır.

Modern dijital teknolojilerin birbirinden ayrılması, teknolojik etkileşimlerin gelişimini anlamayı sağlar, basit dokunuştan dokunmasız etkileşime, görsel 3D ve holografik görüntülere kadar uzanan ilerlemeyi gösterir. Böylece teknolojik eğilimleri takip etme, her teknolojinin uygulanabilirliğini değerlendirme ve yeni dijital yeniliklere hazır bir kullanıcı olma becerileri geliştirilir.

ÖRNEK 1



Etkileşimli Sınıf

Bir sınıfta, öğrencilerin doğrudan parmaklarıyla soruları çözmelerine, nesneleri sürüklemelerine ve dijital kalemle yazı yazmalarına olanak sağlayan dokunmatik ekranlı etkileşimli bir tahta kullanılmaktadır.

ÖRNEK 2



Hologram Sergisi

Bir teknoloji sergisinde ziyaretçiler, dönen ve her açıdan görüntülenebilen tarihi bir nesnenin hologramını izler. Özel gözlüklere ihtiyaç duyulmaz ve etki, nesnenin havada "asılı duruyormuş" gibi görünmesini sağlar.

ÖRNEK 3



Sanal Mimari

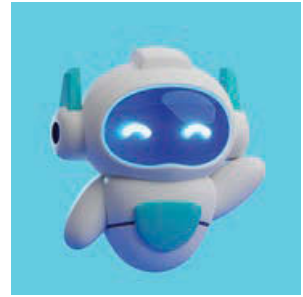
Mimar, bir binanın üç boyutlu modelini göstermek için 3D projektör kullanır. Müşteri, binanın iç ve dış görünümünü gerçekçi bir perspektifle inceleyebilir.



Resim 12 Heliodisplay



Resim 13 3D görüntüler



Resim 14 3D ve hologram





BİLGİ KONTROLÜ SORULARI VE ETKİNLİKLER

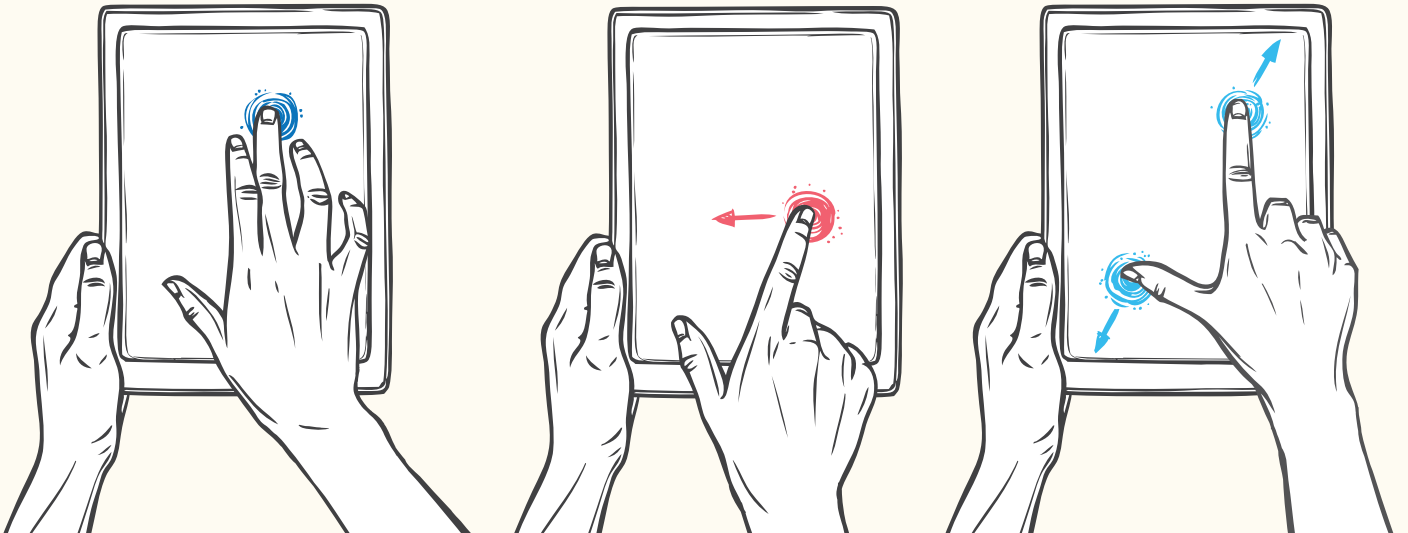


1. Dokunmatik ekranın nasıl çalıştığını ve en yaygın olarak nerelerde kullanıldığını açıklayınız!
2. "Multi-touch" (çoklu dokunma) nedir ve kullanıldığı bir örnek veriniz!
3. Holografi kavramını tanımlayınız ve hologramın nasıl oluşturulduğunu açıklayınız!
4. Holografinin uygulandığı üç alan belirtiniz ve neden kullanıldığını açıklayınız!
5. 3D projeksiyon nedir ve klasik bir monitörden nasıl farklıdır?
6. "Heliodyplay" teknolojisinin ne olduğunu ve nasıl çalıştığını araştırınız!
7. Modern teknolojilerin öğrenmeyi ve sunumları nasıl geliştirdiğini açıklayınız!
8. Dokunmatik ve holografik teknolojilerin geleceği hakkında kısa bir metin yazınız!



DAHA FAZLA BİLGİ EDİNMEK İSTEYENLER İÇİN

Hologramların ve 3D projeksiyonların tıp alanında ve bilimsel görselleştirmede kullanımını araştırınız. Fiziksel temas gerektirmeyen yeni nesil arayüzleri, örneğin hareketlerle (jestlerle) kontrol ve sesli komutlarla yönetim sistemlerini inceleyiniz.



Bir bilgisayar sisteminin çalışma ortamı, dijital teknolojilerin bütünleştirilmesi ve uygulanması için temel oluşturur. Yapay zekâ tabanlı uygulamalar, üretkenlik araçları ve ağ hizmetleriyle etkileşim kurarak süreçlerin yönetilmesini, analiz edilmesini ve otomatikleştirilmesini sağlar, böylece iş süreçlerinin verimliliği, güvenliği ve esnekliği artırılır. İnsan, donanım, yazılım ve dijital araçlar arasındaki etkileşimler, arayüz ve onu destekleyen altyapı, yani bilgisayar sisteminin çalışma ortamı aracılığıyla gerçekleştirilir. Bu çalışma ortamı AŞAĞIDAKİ bileşenlerden oluşur:

- İşletim sistemi (Windows, Linux, macOS).
- Grafiksel kullanıcı arayüzü (GUI).
- Araçlar ve uygulamalar (kullanıcı programları, veritabanları, düzenleyiciler, yapay zekâ platformları).
- Ağ ve güvenlik ayarları.
- Kullanıcı profilleri, izinler ve erişim hakları.

Modern bilgisayar sistemleri, bilgisayarın çalışmasını yöneten bir işletim sistemine ve kullanıcı görevlerinin yerine getirilmesini sağlayan uygulama yazılımlarına dayanır. İşletim sisteminin çalışma ortamı, kullanıcının donanım ve programlarla iletişim kurmasını sağlayan arayüzü oluşturur. Çalışma ortamının yapısını ve işlevlerini, ayrıca uygulama yazılımlarının rolünü anlamak, bilgisayarın etkili ve güvenli bir şekilde kullanılabilmesi açısından büyük önem taşır. Bu ortam, işletim sistemi içinde kolay gezinmeyi, verilerin düzenlenmesini, programların çalıştırılmasını ve farklı yazılım türleriyle verimli bir şekilde çalışılmasını sağlar.

1.6.1 İşletim Sistemi

İşletim sistemi, işlemci, bellek, giriş ve çıkış aygıtları ile uygulamalar dâhil olmak üzere bilgisayarın tüm kaynaklarını koordine eden temel sistem yazılımıdır. En yaygın kullanılan işletim sistemleri şunlardır:

- Windows ,
- Linux ,
- macOS  и
- Android 

Bilgisayar açıldığında, işletim sistemi belleğe yüklenir ve kullanıcı için bir çalışma ortamı hazırlar. Bu çalışma ortamı, masaüstünü (desktop), menüleri, simgeleri ve program pencerelerini içerir.

Oturum açma (login) ve oturumu kapatma (logout), birden fazla kullanıcının bulunduğu bilgisayarlarda gerçekleştirilen temel işlemlerdir. Oturum açıldığında her kullanıcı, kendine ait ayarları, belgeleri ve uygulamaları içeren kendi kullanıcı profiline erişir. Masaüstünde uygulamalara, dosyalara ve sistem işlevlerine hızlı erişim sağlayan simgeler bulunabilir. Görev çubuğu (taskbar) aracılığıyla açık pencereler yönetilir, programlar başlatılır ve sistemin durumu izlenir. Bazı ayarlar kullanıcı ihtiyaçlarına göre özelleştirilebilir. Örneğin, arayüz dili ana dile çevrilebilir, klavye için dil desteği eklenebilir, dil değiştirme klavye kısayolu değiştirilebilir, ekranın belirli bir süre sonra otomatik olarak kilitlenmesi sağlanabilir veya her oturum açılışında parola girilmesi zorunlu hâle getirilebilir.

Farklı parolalara sahip birden fazla kullanıcıyı aynı bilgisayara eklemeyi deneyiniz. Cihazınızı kablolu bir ağa bağlayınız. Bunun için hangi bilgileri bilmeniz gerekir?

Uygulama yazılımları, metin yazma, hesaplama, grafik tasarımı ve internette gezinme gibi belirli görevlerin yerine getirilmesini sağlar. Bu yazılımlar, donanımı doğrudan yönetmez, çalışabilmek için işletim sistemine dayanır. En yaygın uygulama türleri, kelime işlemciler (örneğin Microsoft Word, Writer), elektronik tablo programları (örneğin Excel, Calc), web tarayıcıları (örneğin Chrome, Edge), medya oynatıcılar ve eğitim uygulamalarıdır.

Dosya sistemi, çalışma ortamının en önemli bileşenlerinden biridir. Dosyaların saklanması, düzenlenmesini ve erişilmesini sağlar. Dosyalar, hiyerarşik olarak düzenlenebilen klasörler (dizinler) içinde saklanır. Her dosyanın bir adı ve içerik türünü belirten bir uzantısı (örneğin *.docx, *.jpg, *.mp3) vardır. İşletim sisteminin grafiksel kullanıcı arayüzü sayesinde kullanıcı, dosya ve klasör oluşturabilir, yeniden adlandırabilir, kopyalayabilir, taşıyabilir veya silebilir.

Farklı işletim sistemleri benzer işlevlere sahip olmakla birlikte, görünüşleri ve kullanım biçimleri farklılık gösterebilir. Örneğin Windows, Başlat menüsü, simgeler ve Dosya Gezgini (File Explorer) kullanırken, Linux ortamları (GNOME ve KDE gibi) farklı paneller ve dosya yöneticileri sunar. Mobil cihazlarda ise Android ve iOS, dosyalara uygulamalar ve bulut hizmetleri üzerinden erişim sağlar, diğer cihazlarla senkronizasyon otomatik olarak gerçekleştirilir.

Uygulamalar farklı kaynaklardan yüklenebilir: çevrim içi mağazalardan (örneğin Microsoft Store, Google Play), kurulum dosyalarından (*.exe, *.msi) veya ağ üzerinden. Kullanıcının, yazılımın kaynağına ve lisansına dikkat etmesi önemlidir. Lisans, uygulamanın ücretsiz, açık kaynaklı, ticari veya belirli bir süreyle sınırlı deneme sürümü olup olmadığını belirler. Creative Commons lisansları ise belgeler, görseller ve eğitim materyalleri gibi içerikler için kullanılır ve içerik sahiplerinin kullanım haklarını farklı düzeylerde tanımlamalarına olanak tanır.

İşletim sistemi ayrıca güvenlik ve bakım araçlarını da içerir. Bunlar, **güvenlik yazılımları** (antivirüs programları), sistem güncellemeleri, kullanıcı erişim denetimi ve yedekleme gibi işlevleri kapsar. Bu özellikler, kötü amaçlı yazılımlara karşı koruma sağlamak ve bir arıza durumunda verilerin korunmasını güvence altına almak açısından büyük önem taşır. Kullanıcıların, yetkisiz erişim veya veri kaybı riskini en aza indirmek için cihazlarını düzenli olarak güncellemeleri ve güçlü parolalar kullanmaları gerekir. Ayrıca, işletim sisteminin sunduğu güvenlik araçlarına ek olarak bilgisayara ilave koruma yazılımları da yüklenebilir.

Dijital teknolojiler ile bilgisayar sisteminin çalışma ortamı arasındaki ilişkiyi daha açık bir şekilde göstermek amacıyla, bu bağlantı AŞAĞIDAKİ tabloda sunulmuştur.

Dijital Teknoloji	Çalışma Ortamıyla İlişkisi
Yapay zekâ (AI)	Otomasyon için işletim sistemlerine entegrasyon (sesli asistanlar, akıllı öneriler, güvenlik taraması).
Bulut teknolojileri (Cloud)	Çalışma ortamı, yerel bilgisayardan çevrim içi çalışma ortamlarına doğru genişlemektedir (Google Workspace, Microsoft 365).
Uzaktan çalışma araçları	Çalışma ortamı mobil hâle gelmektedir: Kaynaklara her yerden erişim, VPN, Remote Desktop, Teams, Zoom aracılığıyla sağlanır.
Siber güvenlik	Antivirüsler, güvenlik duvarları (firewall), şifreleme gibi güvenlik çözümleri çalışma ortamının bir parçası olarak çalışır.
Verimlilik ve otomasyon araçları	Microsoft Copilot, Notion AI, Grammarly gibi araçlar, çalışma kalitesini artıran yazılım çalışma ortamının bileşenleridir.
Etkileşimli kullanıcı arayüzleri	Dijital tahtalar, dokunmatik ekranlar, jestlerle kontrol ve VR/AR girişleri, çalışma ortamındaki etkileşimli teknolojilerdir.

Tablo 1 Dijital teknolojinin bilgisayar sistemiyle bağlantısı

Dijital teknoloji (Yapay zekâ, bulut teknolojileri, otomasyon) bilgisayar sisteminin çalışma ortamı, içerisinde çalışarak daha hızlı, daha verimli ve daha güvenli bir çalışma imkânı sağlar.



SONUÇ

İşletim sistemi, bilgisayarın tüm çalışma işleyişini kontrol eden ve yöneten temel yazılımdır. Donanım bileşenleri ile kullanıcı arasında bağlantı kurar, programların çalıştırılmasını, dosyaların, belleğin, aygıtların ve güvenlik süreçlerinin yönetilmesini sağlar. Bu işlevleri sayesinde işletim sistemi, kullanıcıların bilgisayarı kullanmasını kolaylaştıran menüler, masaüstü, ayarlar, pencereler ve diğer grafiksel öğeler aracılığıyla çalışabilecekleri kullanıcı ortamını oluşturur.

İşletim sistemi ile uygulama yazılımları arasındaki farkı anlamak, temel dijital kavramlardan biridir. İşletim sistemi, bilgisayarın çalışabilmesi için gerekli olan temel yapıyı oluştururken, uygulama yazılımları kullanıcının belirli görevlerini yerine getirir. Bu görevler arasında metin işleme, çizim, iletişim, oyun oynama ve internette gezinme yer alır. Kullanım amacına göre uygulamalar şu şekilde sınıflandırılabilir: eğitim yazılımları, iş uygulamaları (ofis paketleri), multimedya yazılımları, iletişim programları, oyunlar, grafik araçları ve güvenlik yazılımları. Yazılım kullanmanın yanı sıra sistemi korumak da önemlidir. Bu nedenle kullanıcıların, antivirüs ve güvenlik programlarını değerlendirirken dikkate alınması gereken kriterleri bilmesi gerekir. Bu kriterler arasında, virüslere ve kötü amaçlı yazılımlara karşı koruma düzeyi, çalışma hızı, güncellemeler, güvenlik duvarı (firewall) seçenekleri, kullanıcı desteği, bilgisayar performansına etkisi ve ek özellikler (ortalama karşıtı koruma, fidyeye yazılımı koruması vb.) bulunur. Bu kriterlerin uygulanmasıyla, dijital cihazlarla çalışırken yüksek düzeyde güvenlik sağlanır.

ÖRNEK 1



Belgeleri Klasörlerde Düzenleme

Kullanıcı, masaüstünde “Ödevler” adlı bir klasör oluşturur. Hazırladığı tüm Word belgelerini, derslere göre ayrılmış şekilde bu klasörde saklar. Böylece dosyalar daha düzenli hâle gelir ve gerekli belgelere kolayca erişim sağlanır.

ÖRNEK 2



Görev Çubuğu Üzerinden Uygulama Başlatma

Ekranın alt kısmında öğrenci, görev çubuğundaki (taskbar) web tarayıcısı (Chrome) simgesine tıklar. Böylece menüler arasında arama yapmaya gerek kalmadan uygulama hemen açılır.

ÖRNEK 3



Lisanslı Yazılım Yükleme

Öğretmen, matematik öğrenmek için bir eğitim programı yükler. Kurulum sırasında Creative Commons lisansına sahip ücretsiz sürümü seçer, böylece programı yasal olarak ve kısıtlama olmadan kullanabilir.

Dijital Teknoloji ile Doğrudan Bağlantı

Öğretmen, AŞAĞIDAKİ bileşenleri içeren bir bilgisayar çalışma ortamında çalışır:

- LMS (Learning Management System – Öğrenme Yönetim Sistemi),
- Öğrencilerin gelişimini analiz eden bir yapay zekâ aracı,
- Bulut ile otomatik senkronizasyon,
- Şifreleme ve iki faktörlü kimlik doğrulama ile sağlanan güvenlik koruması.

BİLGİ KONTROLÜ SORULARI VE ETKİNLİKLER

1. Bilgisayarda işletim sisteminin rolünü açıklayınız.
2. “Çalışma ortamı” kavramı ile ne ifade edilmektedir?
3. İşletim sistemi ile uygulama yazılımları arasındaki farklar nelerdir?
4. Masaüstü (desktop) üzerinden gerçekleştirilebilecek bazı işlemleri belirtiniz.
5. Yazılım lisansının ne olduğunu ve neden önemli olduğunu açıklayınız.
6. En sık kullandığınız uygulamaları ve bunları hangi amaçlarla kullandığınızı yazınız.
7. Açık kaynaklı yazılım ile ticari yazılım arasındaki farkı araştırınız.



DAHA FAZLA BİLGİ EDİNMEK İSTEYENLER İÇİN



Mobil cihazlarda kullanılan işletim sistemlerini, çalışma ortamlarının masaüstü bilgisayarlardan nasıl farklılaştığını ve avantajlarını araştırınız. Sanal makinelerin ne olduğunu ve farklı işletim sistemlerinin test edilmesi ve simülasyonunda nasıl kullanıldıklarını inceleyiniz!



Modern dijital teknolojileri, dokunmatik ve çoklu dokunmatik ekran teknolojileri, temassız teknolojiler, 3D görüntüler, holografi ve heliodisplay gibi teknolojileri ayırt etmek ve anlamak, bu teknolojilerin olanaklarını ve sınırlamalarını tanımayı, aynı zamanda teknolojiyi güvenli, yaratıcı ve bilinçli bir şekilde kullanmaya hazırlıklı olmayı sağlar. Bununla birlikte, dijital okuryazarlığın önemli bir bölümü de dijital teknolojinin kullanımına ilişkin etik **kuralların** bilinmesidir. Gizliliğe ve kişisel verilere saygı göstermek, saldırgan iletişimden ve siber zorbalıktan kaçınmak, telif haklarına saygı göstermek ve paylaşmadan önce bilgilerin doğruluğunu kontrol etmek, teknolojinin kötüye kullanım, hile veya zarar verme olmadan sorumlu bir şekilde kullanılmasını sağlar. Dijital teknolojinin etik kullanımı, çağdaş dijital dünyada güvenli, sorumlu ve bilinçli bir şekilde hareket etmenin temelini oluşturur. Teknoloji geliştikçe, özellikle genç kullanıcıların çevrim içi ortamda doğru davranmayı, kişisel verilerini korumayı ve sanal ortamda diğer insanlara saygı göstermeyi öğrenme ihtiyacı da artmaktadır.

Teknolojik okuryazarlık ile birlikte güçlü bir **dijital etik** anlayışı da geliştirilmelidir, çünkü gelişmiş teknolojiler beraberinde sorumlulukları da getirir. Somut durumlar üzerinden, güvenli ve doğru davranış biçimleri öğrenilmeli, yani sorumlu bir dijital vatandaş olmanın yolları kavranmalıdır.

Her şeyden önce, teknolojinin kullanımında etik anlayış **gizliliğe saygı göstermekle** başlar. Bu, başkalarına ait fotoğrafları, videoları veya kişisel bilgileri izin almadan paylaşmamamız gerektiği anlamına gelir. Bize güvenli veya bir “şaka” gibi görünse bile bu tür paylaşımlar yapılmamalıdır. Kişisel bilgiler her zaman korunmalıdır, parolalar güçlü olmalı, başka kişilerle paylaşılmamalı ve şüpheli internet sitelerine veya uygulamalara girilmemelidir.

Örnek: Öğrenciler, öğretmenin izni olmadan ders sırasında veya okul dışında katılımcıların izni olmadan cep telefonu ile kayıt yapmamalıdır.

Teknolojinin etik kullanımında bir diğer önemli konu **kişisel verilerin korunmasıdır**. **Örnek:** Güvenilir olmayan dokunmatik veya temassız terminallere kişisel parmak izi ya da biyometrik veriler bırakılmamalıdır. Bir diğer önemli konu ise **telif haklarına saygı gösterilmesidir**. Dijital dünyada birçok içerik, görseller, müzikler, videolar ve metinler kolayca erişilebilir durumdadır. Ancak bu durum, bu içeriklerin izinsiz kullanılabileceği anlamına gelmez. Her zaman kaynağın belirtilmesi, izin verilen materyallerin veya eğitim amaçlı kullanıma açık ücretsiz kaynakların tercih edilmesi gerekir. Böylece saygı ve dürüstlük kültürü geliştirilir. **Örnek:** 3D modeller kullanılarak hazırlanan bir sunumda, kullanılan modelin alındığı kaynaklar belirtilmelidir.

Aynı zamanda, internet üzerindeki iletişime de dikkat edilmelidir. Çevrim içi ortam, küçümseme, alay etme veya olumsuzluk yayma alanı olmamalıdır. Her sözün bir sonucu vardır, ekranın arkasından yazılmış olsa bile. Kibar bir şekilde yazmayı bilmek, kırıcı yorumlar kullanmamak ve bir kişiyi incitebilecek toplu saldırılara veya şakalara katılmaktan kaçınmak gerekir. Bireysel veya toplu saldırılar yasal düzenlemelere tabidir ve cezalandırılabilir. Bu gibi durumlarda sessiz kalmak çözüm değildir, bildirimde bulunmak cesur ve doğru bir seçimdir. Aynı şekilde, siber zorbalık da dijital toplumdaki en ciddi tehditlerden biridir. Teknolojinin etik kullanımı, bu tür davranış biçimlerinden aktif olarak kaçınmayı, aynı zamanda bu davranışlar fark edildiğinde tanımayı ve bildirmeyi gerektirir.

Günümüzde bilgilerin çok hızlı yayıldığı bir dönemde, dijital okuryazarlığın önemli bir bölümü de bilgilerin doğruluğunun kontrol edilmesidir. Bu, herhangi bir haber, video veya fotoğrafı paylaşmadan önce kaynağın güvenilir olup olmadığının kontrol edilmesi anlamına gelir. Bu şekilde yanlış bilgilerin, manipölasyonların ve paniğin yayılması önlenir.

Teknolojinin etik kullanımı, cihazların sorumlu bir şekilde kullanılmasını da ifade eder.

Teknoloji hiçbir zaman dolandırıcılık, kopya çekme, bilgisayar korsanlığı, sahte kimlik kullanımı veya zarar verme amacıyla kullanılmamalıdır. Bu ilke okul ortamı için de geçerlidir, cihazlar sınavların fotoğrafını çekmek, kopya çekmek veya öğrencileri ve öğretmenleri izinleri olmadan kaydetmek amacıyla kötüye kullanılmamalıdır.

eknolojinin öz denetimle ve dengeli kullanımı da daha az önemli değildir. Cep telefonlarının, sosyal ağların veya oyunların aşırı kullanımı, sağlık, öğrenme süreci ve kişiler arası ilişkiler üzerinde olumsuz etkiler oluşturabilir. Etik alışkanlıkların bir parçası da kişinin kendi sınırlarını fark etmesi, ölçülü kullanım sağlaması ve bilinçli seçimler yapmasıdır.

ÖRNEK 1



İçeriği Paylaşmadan Önce Doğruluk Kontrolü

Öğrenci, Instagram, TikTok veya Facebook'ta bir video ya da haber paylaşmadan önce kaynağın güvenilir olup olmadığını kontrol eder ve doğru kaynağı belirtir. Bu şekilde yanlış haberlerin ve manipölasyonların yayılması önlenir.

ÖRNEK 2



Okul Cihazlarının ve Internetin Etik Kullanımı

Bilgisayar laboratuvarında çalışırken bir öğrenci, önceki öğrencinin Google Classroom hesabında oturumunun açık kaldığını fark eder. Öğrenci, profili incelemeyi, bunun yerine hesabın çıkışını yapar ve kendi hesabıyla giriş yapar. Okul bilgisayarı veya tableti kullanan bir öğrenci, başkalarının hesaplarına giriş yapmaz, izin almadan ayarları değiştirmez ve yasa dışı içerikler indirmeyi. Bunun yerine cihazları yalnızca okul etkinlikleri için kullanır ve okul kurallarına uyar.

ÖRNEK 3



Dijital Medyaların Etik Şekilde Kaydedilmesi ve Kullanılması

Okul için bir proje hazırlayan öğrenci (örneğin: video sunumu veya röportaj), öncelikle kayıta yer alan tüm kişilerden izin alır ve videonun hangi amaçla kullanılacağını açıkça açıklar. Tarih dersi projesi için bir grup öğrenci bir öğretmenle röportaj yapar. Kayıt başlamadan önce öğretmenden izin isterler, kaydın yalnızca sınıf içinde gösterileceği konusunda bilgi verirler ve yayınlamadan önce son hâlini onaylaması için kendisine gönderirler.



Resim 15

Akıllı telefonun bilinçli kullanımı

BİLGİ KONTROLÜ SORULARI VE ETKİNLİKLER



1. Dijital teknolojinin etik kullanımı ne ifade eder? İki kural belirtiniz!
2. Sana zararsız görünse bile, başkasına ait bir fotoğrafın izin almadan paylaşılmasının neden önemli olduğunu açıklayınız!
3. Bir sınıf arkadaşınıza hakaret eden bir mesaj aldığınızı düşününüz. Etik dijital davranış kurallarına göre ne yapardınız?
4. AŞAĞIDAKİ örneği okuyunuz ve etik açıdan doğru ve yanlış olan davranışları belirtiniz:
5. Bir öğrenci, cep telefonu ile dersin bir bölümünü kaydeder ve öğretmenden izin almadan videoyu özel bir grupta paylaşır. Nedenlerini açıklayınız.
6. Şu durumu değerlendiriniz: Kaynağı kontrol edilmeden, şok edici görünen bir haberin paylaşılması etik midir? Kararınızı açıklayınız!
7. Öğrenciler arasındaki dijital kültürü geliştirmek için "Okul Dijital Kodu"na ekleyeceğiniz kısa bir kural oluşturunuz!
8. Bir grup sohbetinde (chat) bir kişinin korku veya paniğe neden olabilecek yanlış bir bilgi paylaşması durumunda nasıl davranacağınızı açıklayınız. Durumu etik bir şekilde çözmek için hangi adımları atardınız?



DAHA FAZLA BİLGİ EDİNMEK İSTEYENLER İÇİN

Dünyada kaç çocuğun internet tehditlerine ve internetin etik olmayan kullanımına maruz kaldığını araştırınız. Avrupa Komisyonu'nun Dijital Hizmetler Yasası'nın (Digital Services Act) son sürümünü AŞAĞIDAKİ adresten inceleyiniz:

<https://digital-strategy.ec.europa.eu/en/policies/digital-services-act-package>.



Her bilgisayar sisteminde bilgiler dosyalar şeklinde saklanır ve bunların düzenlenmesi klasörler aracılığıyla gerçekleştirilir. Hiyerarşik yapı olarak bilinen bu düzenleme yöntemi, verilerin mantıksal olarak gruplandırılmasını, kolay erişilmesini ve verimli bir şekilde yönetilmesini sağlar. Doğru bir düzenleme yapılmadığında, özellikle bilgi miktarı fazla olduğunda dosyaları bulma, açma ve kaydetme işlemlerinde zorluklar ortaya çıkabilir. Klasörlerin (folder) hiyerarşisini ve dosya adlandırma kurallarını anlamak, dijital okuryazarlığın ve bilgisayarların güvenli kullanımının temelini oluşturur.

Klasörler (veya dizinler), dosyaların ya da diğer klasörlerin saklandığı alanlardır. Hiyerarşik sistem, klasörlerin alt klasörler içerebilmesi anlamına gelir ve böylece ağaç yapısına benzer bir düzen oluşturulur. Bu yapının en üstünde kök klasör (root) bulunur ve buradan diğer tüm alt dizinler ve dosyalar dallanarak oluşur.

Her dosyanın kendine özgü bir adı ve veri türünü gösteren bir uzantısı vardır. Örneğin, .txt metin dosyasını, .jpg görüntü dosyasını, .mp3 ise ses kaydını ifade eder. Önerilen dosya adlandırma yöntemi, açıklayıcı isimler kullanmayı (örneğin: "Matematik_Odevi_mayis.docx") ve özel karakterlerden kaçınmayı içerir. Bu yöntem, özellikle dosya sayısının fazla olduğu durumlarda dosyaların tanımlanmasını ve aranmasını kolaylaştırır.

Dosya sistemleri, Windows işletim sistemindeki File Explorer veya macOS işletim sistemindeki Finder gibi grafiksel dosya yöneticileri aracılığıyla görüntülenebilir. Kullanıcı yeni bir klasör oluşturabilir, dosyaları kopyalayabilir veya taşıyabilir, yeniden adlandırabilir ya da silebilir. Ayrıca dosyaları ad, tür, boyut veya son değiştirilme tarihine göre sıralama işlevleri de bulunmaktadır. Gelişmiş durumlarda düzenleme işlemleri, etiketler, işaretler veya kategoriler kullanılarak da yapılabilir ve böylece ek filtreleme imkânı sağlanır.

Hiyerarşik yapı, dosyaların projelere, derslere, zamana veya kullanım amacına göre mantıklı bir şekilde gruplandırılmasını sağlar. Örneğin, "Okul" adlı ana klasör içerisinde her ders için (örneğin "Matematik", "Kimya") alt klasörler oluşturulabilir ve her birinde belgeler, sunumlar ve ödevler saklanabilir. Bu düzenleme, öğrencilerin ve öğretmenlerin uzun süreli arama yapmaya gerek kalmadan veriler arasında hızlı ve kolay bir şekilde hareket etmelerini sağlar.

Dosyalar ve klasörler yıl, ay veya haftaya göre de düzenlenebilir. Örneğin, belirli bir etkinliğe ait fotoğraflar şu yapı içerisinde saklanabilir: "2025 → Mart → Gezi". Bu yaklaşım, arşivleme ve dönemlere göre ilerlemenin takip edilmesi açısından oldukça yararlıdır. Ayrıca ağ veya bulut üzerinden diğer kullanıcılarla paylaşımı kolaylaştırır.

Google Drive, Dropbox ve OneDrive gibi bulut hizmetleri, aynı hiyerarşik düzenleme ilkesini sunar, ancak herhangi bir cihazdan erişim ve gerçek zamanlı paylaşım imkânı sağlar. Kullanıcılar klasör oluşturabilir, belgeleri sürükleyip bırakabilir, erişim izinleri belirleyebilir (yalnızca görüntüleme, düzenleme) ve yapılan değişiklikleri takip edebilir. Bu özellikler özellikle ekip çalışmaları, okul projeleri ve dijital portfolyolar için oldukça kullanışlıdır.

Düzenlemede iyi uygulamalar, gereksiz dosyaların düzenli olarak temizlenmesini, yedek kopyaların oluşturulmasını ve bilgisayar kullanımına başlandığı andan itibaren uygun bir yapı kurulmasını içerir. Yanlış düzenleme, zaman kaybına, verilerin çoğalmasına ve önemli bilgilerin kaybolma riskine yol açar.

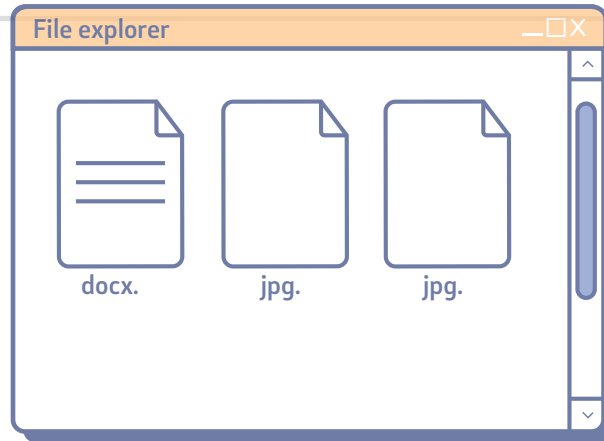
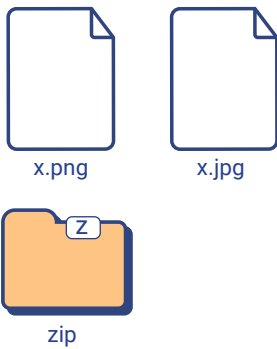
Dosya ve klasörlerin hiyerarşik bir yapı içerisinde düzenlenmesi, her bilgisayar kullanıcısı için temel bir beceridir. Uygun bir düzenleme, verimliliği artırır, veri kaybı riskini azaltır ve paylaşımı kolaylaştırır. Dijital okuryazarlık, tam olarak bu ilkelerle başlar ve kullanılan cihaz, işletim sistemi veya yazılımdan bağımsız olarak geçerliliğini ve önemini korur.



SONUÇ

Farklı türlerde belge formatları bulunmaktadır. Örneğin, .docx ve .pdf metin belgeleri için kullanılan formatlardır. .jpg ve .png görüntüler için, .mp4 video dosyaları için, .mp3 ses dosyaları için, .zip ve .rar ise sıkıştırılmış dosyalar için kullanılan formatlardır. Bu formatların tanınması sayesinde veriler daha kolay düzenlenir ve dosyalar, onları açan uygun programlarla ilişkilendirilir. Örneğin, .docx dosyaları için Word, .pptx dosyaları için PowerPoint, .pdf dosyaları için Adobe Reader, .jpg/.png dosyaları için görüntü görüntüleyiciler ve ses ile video formatları için medya oynatıcılar kullanılır.

Dosyaların hiyerarşik olarak düzenlenmesi her kullanıcı için önemlidir. Bu düzenleme, klasörlerin ve alt klasörlerin mantıklı bir sıraya göre oluşturulmasını ifade eder. Böyle bir yapı, verilerin bulunmasını, düzenlenmesini ve saklanmasını kolaylaştırır, düzensizliğin ve materyal kaybının oluşmasını önler. Formatların tanınması ve organizasyonun yanı sıra, dosya türü (format/file type) ile dosyayı açan program arasındaki farkın bilinmesi de önemlidir. En önemli konulardan biri de sıkıştırmanın ve bunun dosyaların boyutu ile kalitesi üzerindeki etkisinin analiz edilmesidir. Ses veya video dosyalarında sıkıştırma, sesin netliğini ve görüntü kalitesini etkileyebilir. Arşiv dosyalarında (.zip, .rar) ise dosya boyutu, özgün kalite kaybedilmeden azaltılır, ancak dosyanın kullanılabilmesi için önce sıkıştırılmış hâlden çıkarılması gerekir.



Resim 16 Klasör ve dosyaların organizasyonu

ÖRNEK 1



Dersler için Dosyaların Düzenlenmesi

Öğrenciler, "Okul" adlı ana bir klasör oluşturur ve bunun içerisinde "Matematik", "Bilişim" ve "Tarih" adlı üç alt klasör düzenler. Bu alt klasörlerin her birinde, ev ödevlerini içeren Word belgeleri, PDF sunumları ve tahtadan çekilmiş görüntüler saklanır.

ÖRNEK 2



Tarihlere Göre Yapılandırma

"Fotoğraflar" klasörü içerisinde aylara göre alt klasörler oluşturulur: "2025_Ocak", "2025_Şubat" ve "2025_Mart". Bu klasörlerde, her etkinliğe ait fotoğraflar kronolojik sıraya göre saklanır.

ÖRNEK 3



Bulut Organizasyonu

Bir öğrenci, tüm projelerini saklamak için Google Drive kullanır. Öğrenci, derslere göre alt klasörlere ayrılmış "Projeler" adlı bir klasöre sahiptir. Her klasördeki belgeler, içerik ve tarihe göre adlandırılır (örneğin: "Biyoloji_Araştırması_2025_03.docx").

BİLGİ KONTROLÜ SORULARI VE ETKİNLİKLER



1. Dosya hiyerarşik yapısının ne olduğunu açıklayınız!
2. Dosyaların doğru şekilde adlandırılmasının faydaları nelerdir?
3. Bir yarıyıl boyunca tüm derslere ait belgeleri nasıl düzenleyeceğinize dair bir şema oluşturunuz!
4. Dosya uzantısı nedir? Beş farklı örnek veriniz ve açıklayınız!
5. Yeni bir klasörün nasıl oluşturulduğunu ve bir dosyanın bu klasöre nasıl taşındığını açıklayınız!
6. File Explorer veya Finder programlarında hangi işlevler bulunmaktadır?
7. Dosyaların yerel olarak ve bulut ortamında saklanması arasındaki fark nedir?
8. Yedek kopyaların ne olduğunu ve neden önemli olduklarını araştırınız!!



UYGULAMALI ETKİNLİK

AŞAĞIDAKİ bilgisayar koruma programlarını kriterlere uygun olarak değerlendiriniz! Programın adını da yazınız!

Kriter	Açıklama	Programın adı	
Tehditleri tespit etme etkinliği	Programın virüsleri, kötü amaçlı yazılımları, casus yazılımları ve diğer tehditleri tespit etme yeteneği.		
Gerçek zamanlı koruma	İnternette gezinme, dosya indirme ve dosya açma sırasında gerçek zamanlı koruma sağlayıp sağlamadığı.		
Hız ve performans	Programın sistem hızını ne kadar etkilediği. Arka planda sessiz çalışıp çalışmadığı.		
Kullanılabilirlik ve arayüz	Kullanımının kolay olup olmadığı, sezgisel bir arayüze ve anlaşılır seçeneklere sahip olup olmadığı.		
Güncelleme sıklığı	Yeni tehditlere karşı veri tabanının ne sıklıkla güncellendiği.		
Ek özellikler	Güvenlik duvarı (Firewall), ebeveyn kontrolü, VPN, web kamerası koruması vb.		
Destek ve dokümantasyon	Teknik destek, kullanım kılavuzları ve SSS (Sıkça Sorulan Sorular) bölümünün bulunması.		
Fiyat ve lisanslama	Ücretsiz, deneme sürümü veya ücretli olup olmadığı. Fiyatının uygun olup olmadığı.		
Uyumluluk	Farklı işletim sistemleri (Windows, macOS, Linux) için destek sağlaması.		
Kullanıcı yorumları ve itibar	Kullanıcıların ve uzmanların görüşleri, ödüller ve değerlendirmeler.		



DAHA FAZLA BİLGİ EDİNMEK İSTEYENLER İÇİN

Dosya sistemi yapısı kavramının ne anlama geldiğini ve hangi dosya sistemi türlerinin bulunduğunu (örneğin: FAT32, NTFS, ext4) araştırınız. Çok sayıda dosyanın düzenlenmesi için iyi uygulamaları ve adlandırma ile sınıflandırma işlemlerini otomatikleştirmek için kullanılan araçları inceleyiniz.



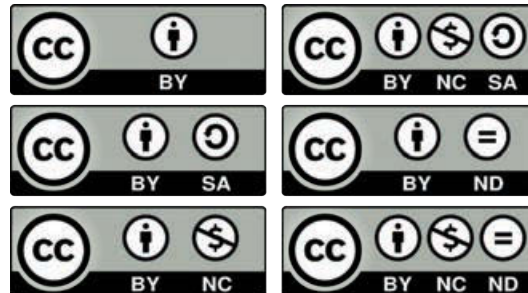
Öğretmenler ve öğrenciler eğitim sürecinde farklı dijital araçlar kullandıklarında, bu araçların hangi lisanslarla çalıştığını kontrol etmelidirler. Bu lisansların kapalı kaynak kodlu olup olmadığı, uyarlanıp uyarlanamayacağı ve telif haklarını ihlal etmeden kullanılmasına izin verilip verilmediği incelenmelidir. Ayrıca, dijital araçların, özellikle yapay zekâ (örneğin otomatik metin oluşturma) kullanılarak eğitim içerikleri oluşturulduğunda, elde edilen sonuçların paylaşıp paylaşılamayacağı veya başka lisanslarla korunup korunmadığına dikkat edilmelidir.

Bilişim dünyasında yazılımlar ve dijital içerikler, kullanımları üzerinde belirli haklara sahip olan kişiler tarafından oluşturulur. Başkalarının bu ürünleri hangi koşullarda kullanabileceğini düzenlemek amacıyla lisanslar kullanılır. Lisanslar, yazılımın veya içeriğin kurulumu, kopyalanması, değiştirilmesi ve dağıtılması ile ilgili kullanım haklarını belirler. Creative Commons, eser sahiplerinin kendi çalışmalarını açıkça belirlenmiş koşullar altında paylaşmalarına olanak sağlayan modern bir lisanslama yaklaşımıdır. Bu kavramların anlaşılması, yazılımların, çoklu ortam içeriklerinin ve eğitim materyallerinin etik bir şekilde kullanılması açısından temel öneme sahiptir.

Yazılım lisansları, yazılımın sahibi veya üreticisi ile son kullanıcı arasında yapılan yasal sözleşmelerdir. Her biri farklı kullanım düzeylerine izin veren çeşitli lisans türleri bulunmaktadır. Ticari yazılımlar genellikle yalnızca tek bir kopyanın kullanılmasını, değiştirilmemesini ve paylaşılmamasını sınırlayan lisanslarla sunulur. Microsoft Office, Adobe Photoshop ve diğer profesyonel yazılım paketleri bu tür yazılımlara örnektir. Kullanıcı, yazılımı kullanmadan önce genellikle EULA (End User License Agreement – Son Kullanıcı Lisans Sözleşmesi) biçimindeki lisans koşullarını kabul etmek zorundadır.

Bunun yanında, ücretsiz kullanıma izin veren Freeware gibi lisans türleri de bulunmaktadır. Bu tür lisanslar genellikle herhangi bir ücret ödemediği kullanıma izin verir, ancak çoğunlukla değiştirme hakkı sunmaz. Shareware lisansları, belirli bir deneme süresi kullanımına izin verir ve bu sürenin sonunda kayıt veya ödeme yapılmasını gerektirir. Open Source (Açık Kaynak) lisansları ise kullanıcıya yazılımı kullanma, inceleme, değiştirme ve dağıtma hakkı verir. En bilinen örnekleri arasında GNU General Public License (GPL), MIT License ve Apache License bulunmaktadır.

Creative Commons (CC), en çok resim, metin, sunum, video ve müzik gibi içeriklerin lisanslanmasında kullanılan bir lisanslama türüdür. CC lisansları, tamamen yasaklamak yerine, eser sahibinin tercihiyle bağlı olarak farklı kullanım düzeylerine izin verir. Altı temel CC lisans türü bulunmaktadır ve bunlar, eser sahibinin belirtilmesi (BY), ticari olmayan kullanım (NC), değişiklik yapılamaması (ND) ve aynı koşullarla paylaşma (SA) gibi şartların farklı kombinasyonlarından oluşur).



Resim 12 CC Lisans türleri

Örneğin, CC BY lisansı, eser sahibinin belirtilmesi koşuluyla eserin kullanılmasına, paylaşılmasına ve değiştirilmesine izin verir. CC BY-NC lisansı aynı hakları sağlar, ancak ticari amaçlarla kullanıma izin vermez. CC BY-ND lisansı ise eserin paylaşılmasına izin verir, ancak değiştirilmesine izin vermez. Bu kombinasyonlar, eser sahibinin haklarına saygı göstererek kaynakların esnek bir şekilde kullanılmasına olanak sağlar.

Eğitimde CC lisansları, öğretim materyallerinin paylaşılması için giderek daha fazla kullanılmaktadır. Birçok üniversite, kütüphane ve araştırma kuruluşu, içeriklerini bu lisanslarla yayımlayarak küresel ölçekte erişilebilir hâle getirmektedir. Bu durum açık bilginin yaygınlaşmasına, daha iyi iş birliğine ve dijital dayanışmaya katkıda bulunmaktadır.

Her kullanıcının kullandığı yazılımın veya içeriğin lisansına dikkat etmesi önemlidir. Yasa dışı kopyalama, dağıtım veya değiştirme, telif haklarının ihlali anlamına gelir ve yasalarla cezalandırılabilir. Hukuki sonuçların yanı sıra, bu tür davranışlar dijital vatandaşlık etiğini zedeler ve içerik üreticilerinin çalışmalarına zarar verir.

Ayrıca, içeriğin kullanımına ilişkin herhangi bir kısıtlamanın bulunmadığı kamu malı (Public Domain) alanları da vardır. Bu kaynaklar, telif haklarının süresi dolduğu veya yazar haklarından açıkça feragat ettiği için lisans gerektirmeden özgürce kullanılabilir. Bununla birlikte, bu tür kaynakları kullanırken bile özgün yaratıcıya saygı göstermek amacıyla kaynağın belirtilmesi tavsiye edilir.

Lisanslama, yazılım ve dijital içeriğin kullanımını düzenleyen temel mekanizmadır. Farklı lisans türlerini bilen kullanıcılar, kendilerine sunulan araçları ve bilgileri sorumlu bir şekilde kullanabilirler. Creative Commons lisansları, açık paylaşımı özellikle destekler, ancak bunun için açık koşullar da belirler. Bu kurallara uyularak iş birliği, saygı ve adalet temeline dayanan bir dijital kültür oluşturulur.



SONUÇ

- Yazılım ve dijital içeriklerin lisanslanması, kullanıcıların programları, belgeleri, görselleri veya videoları nasıl kullanabilecekleri, paylaşabilecekleri ve değiştirebilecekleri konusunda büyük bir role sahiptir. Ticari, ücretsiz, özgür veya açık kaynaklı yazılım lisansları, kullanıcının yazılımı değiştirip değiştiremeyeceğini, dağıtıp dağıtamayacağını veya yalnızca belirli koşullar altında kullanıp kullanamayacağını belirler. Dijital içerikler kapsamında **Creative Commons** (CC) lisansları büyük önem taşımaktadır. Bu lisanslar, eser sahibinin diğer kişilerin kendi içeriğini nasıl kullanabileceğini belirlemesine yardımcı olmak amacıyla oluşturulmuştur. Tüm CC lisansları, izinler ve kısıtlamaların birleşimine dayanır. Bu sayede eser sahibi belirli haklarını korurken, içeriğin belirli koşullar altında serbestçe kullanılmasına da izin verebilir. En yaygın kullanılan Creative Commons lisansları şunlardır:
- **CC BY (Atıf – Yazarı Belirt)** – İçeriğin kullanılmasına, paylaşılmasına ve değiştirilmesine izin verir, ancak özgün eserin yazarı belirtilmelidir.
- **CC BY-SA (Atıf – Aynı Koşullarla Paylaş)** – İçeriğin değiştirilmesine izin verir, ancak oluşturulan yeni eser aynı lisans koşullarıyla paylaşılmalıdır. Lisanslama, yazılım ve dijital içeriğin kullanımını düzenleyen temel mekanizmadır. Farklı lisans türlerini bilen kullanıcılar, kendilerine sunulan araçları ve bilgileri sorumlu

bir şekilde kullanabilirler. Creative Commons lisansları, açık paylaşımı özellikle destekler, ancak bunun için açık koşullar da belirler. Bu kurallara uyularak iş birliği, saygı ve adalet temeline dayanan bir dijital kültür oluşturulur.

- **CC BY-ND (Atıf – Değiştirme)** – İçeriğin kullanılmasına izin verir, ancak değiştirilmesine izin vermez.
- **CC BY-NC (Atıf – Ticari Olmayan)** – İçeriğin kullanılmasına ve değiştirilmesine izin verir, ancak ticari amaçlarla kullanılmasına izin vermez.
- **CC BY-NC-SA / CC BY-NC-ND** – Ticari olmayan kullanım, değiştirmeme ve aynı koşullarla paylaşma şartlarının farklı birleşimlerinden oluşan lisanslardır.
- **CC0** – Eser sahibi tüm haklarından feragat eder ve içerik kamu malı (Public Domain) hâline gelir.

CC lisanslarını ayırt edebilmek, dijital kaynakların doğru şekilde kullanılmasını, kaynakların belirtilmesini ve içerik oluştururken veya indirirken etik ve yasal kurallara uygun davranılmasını sağlar. Lisansların bilinmesi sayesinde kullanıcılar, kendi çalışmalarını nasıl koruyacaklarına ve başkalarının çalışmalarına nasıl saygı göstereceklerine karar verebilirler.

ÖRNEK 1



Ücretsiz Ancak Sınırlı Yazılım

Bir öğrenci Freeware türünde bir metin düzenleyici kullanmaktadır. Her ne kadar bu yazılım için ücret ödemese de, lisans koşulları bunu yasakladığı için kaynak kodunu değiştiremez veya başkalarına dağıtamaz.

ÖRNEK 2



GPL Lisanslı Açık Kaynak Kodlu Yazılım

Bir bilişim öğretmeni, GPL lisansına sahip bir Linux dağıtımı kullanmaktadır. Yazılımı değiştirebilir ve başkalarıyla paylaşabilir, ancak yeni sürümü aynı lisans koşulları altında dağıtmak zorundadır.

ÖRNEK 3



Creative Commons ile Sunum

Bir öğretmen, internetten CC BY-NC lisansı ile alınmış görseller ve müzikler kullanarak bir PowerPoint sunumu oluşturur. Yazarların kaynaklarını belirtir ve içeriği yalnızca eğitim amaçlı, ticari olmayan kullanımlar için kullanır.

BİLGİ KONTROLÜ SORULARI VE ETKİNLİKLER



1. Yazılım ve dijital içerikler için neden lisanslar vardır?
2. Ticari yazılım ile açık kaynak kodlu yazılım arasındaki farklar nelerdir?
3. 3 çeşit Creative Commons lisansı belirtiniz ve açıklayınız!
4. Lisans türleri arasındaki farkları açıklayınız (CC BY, CC BY-SA, CC BY-ND, CC BY-NC vb.)!
5. Bir yazılımın kamu malı (Public Domain) olması ne anlama gelir?
6. Lisanssız yazılım kullanmanın sonuçları nelerdir?
7. MIT lisansının ne olduğunu ve hangi amaçlarla kullanıldığını araştırınız!
8. Tüm ücretsiz programlar yasal olarak kullanılabilir mi? Açıklayınız!
9. CC lisansına sahip bir yazılım bulunuz ve nasıl kullanıldığını açıklayınız!



UYGULAMALI ETKİNLİK

Orijinal içerik içeren bir belge oluşturunuz. Belge, metin (makale, blog yazısı, şiir) ve görsel/resim içermelidir. İçerik eğitici veya yaratıcı olmalıdır. İçeriğiniz için uygun bir CC lisansı seçiniz. Bu lisansı neden seçtiğinizi açıklayınız! Bu lisans neye izin verir, neyi kısıtlar?



DAHA FAZLA BİLGİ EDİNMEK İSTEYENLER İÇİN

Mobil uygulamaların nasıl lisanslandığını ve Google Play veya App Store gibi mağazalara göre kullanım koşullarının neler olduğunu araştırınız! Yazılım korsanlığının (software piracy) ne olduğunu ve bunun yasal ve etik sonuçlarını inceleyiniz!



1.10

Yapay Zekâ (Genel ve Üretken)

Yapay zekâ (YZ), modern dijital dünyadaki en önemli ve en hızlı gelişen teknolojilerden biridir. Yapay zekâ sayesinde makineler, geleneksel olarak insan zekâsı gerektiren öğrenme, çıkarım yapma, iletişim kurma ve problem çözme gibi yeteneklere sahip olabilmektedir. Günümüzde akıllı telefonlardan ve navigasyon sistemlerinden tıbbi teşhis araçlarına ve otonom araçlara kadar birçok sistem, uygulama ve cihaz bir tür yapay zekâ teknolojisine dayanmaktadır. Bu dersin amacı, yapay zekâ kavramını, sınıflandırılmasını ve kullanım alanlarını açıklamak, özellikle yapay zekânın iki çağdaş yönü olan genel yapay zekâ ve üretken yapay zekâ üzerinde durmaktır.

Yapay zekâ, bir bilgisayar sisteminin konuşma tanıma, problem çözme, deneyimlerden öğrenme ve karar verme gibi akıllı davranış gerektiren görevleri yerine getirme yeteneği olarak tanımlanır. Zekâ düzeyi ve yeteneklerine bağlı olarak yapay zekâ sistemleri iki ana kategoriye ayrılır: genel yapay zekâ (AGI – Artificial General Intelligence) ve belirli amaçlara yönelik uzmanlaşmış veya üretken yapay zekâ.

Genel yapay zekânın amacı, insan gibi düşünebilen, öğrenebilen ve farklı durumlara uyum sağlayabilen bir sistem oluşturmaktır. Bu yapay zekâ türü hâlâ teorik aşamada ve günümüzde pratik olarak kullanılmamaktadır. Genel yapay zekânın, mantık, duygular, bağlam bilgisi ve etik değerlendirmeleri çalışma sürecine dâhil etmesi beklenmektedir. Bu alandaki araştırmalar, görev veya ortamdan bağımsız olarak çalışabilecek evrensel algoritmaların ve öğrenme modellerinin geliştirilmesine odaklanmaktadır.

Genel yapay zekâ henüz gerekli gelişmişlik düzeyine ulaşmamış olsa da akademik ve teknolojik araştırmaların aktif konularından biri olmaya devam etmektedir. Geliştirilen modeller, insan düşünme biçimini taklit etmek, makine ile akıl yürütme, bağlamı anlama ve yeni durumlara uyum sağlama yeteneklerini geliştirmek amacıyla incelenmektedir. Amaç, yalnızca tek bir görevi yerine getirmekle sınırlı olmayan, insan benzeri şekilde farklı alanlarda “düşünebilen” bir sistem oluşturmaktır.

Yapay Zekâ / Bilgisayar Adı	Proje için Oluşturuldu	Ne Yapar?
HAL 9000	2001: Bir Uzay Destanı	Uzay gemisini kontrol eder.
Data (Android)	Star Trek	İnsan gibi düşünme ve duygu yeteneğine sahip bir makine.
JARVIS / FRIDAY	Demir Adam / Yenilmezler	Duygusal zekâyâ sahip kişisel asistan.
Skynet	Terminatör	Kendi bilincini kazanan ve tehlikeli hâle gelen askerî yapay zekâ sistemi.

Tablo 2 Genel yapay zekâ örnekleri

Buna karşılık, üretken yapay zekâ, günümüzde farklı alanlarda kullanılan pratik bir yapay zekâ türüdür. Bu yapay zekâ, yeni içerikler oluşturma konusunda uzmanlaşmıştır, metin, görsel, müzik, video veya hatta programlama kodu üretebilir. Derin öğrenme (deep learning) temelli üretken modeller, çok büyük miktardaki verilerle eğitilir ve daha sonra insan yaratıcılığına benzeyen yeni, özgün ifadeler oluşturmak için kullanılır.

Üretken yapay zekâ günümüzde günlük yaşamın birçok alanında yer almaktadır. Örneğin, mobil telefonlarda cümlelerin otomatik tamamlanmasını sağlayan, metinleri çeviren veya hatta tamamen yeni makaleler yazabilen dil modelleri bulunmaktadır. DALL•E veya Midjourney gibi görsel üretim sistemleri, metin açıklamalarına dayanarak özgün görseller oluşturmak için kullanılmaktadır. Müzik endüstrisinde algoritmalar besteler oluştururken, film sektöründe sanal sesler veya karakterler geliştirilmektedir. Tüm bu uygulamalar, insan düşünme biçimini andıran yaratıcı sonuçlar elde etmek için verileri ve istatistiksel yöntemleri birleştirir.

Özellik	Genel Yapay Zekâ (AGI)	Üretken Yapay Zekâ (GenAI)
Uygulama Alanı	Evrensel (insan gibi)	Sınırlı, özel amaçlı
Öğrenme Yeteneği	Kendi kendine öğrenme, evrensel	Büyük veri kümelerinden öğrenir
Bilinç / Anlama	Potansiyel (teorik olarak)	Hayır, yalnızca istatistiksel bir modeldir
Örnek	Bilim kurgudan kurgusal bilgisayar	ChatGPT, DALL•E, Copilot
Durum	Teori aşamasında, henüz mevcut değil	Gerçek ve günümüzde kullanılan teknoloji

Tablo 3 Genel ve üretken yapay zekâ karşılaştırması

Üretken yapay zekâ algoritmaları genellikle sinir ağları adı verilen mimarileri kullanır. Özellikle üretken karşıt ağlar (GAN'lar) ve transformatörler bu alanda yaygın olarak kullanılan yöntemlerdir. Bu sistemler, milyonlarca örneği işleyerek içeriklerin temel örüntülerini, yapılarını ve stillerini öğrenir. Daha sonra bu "öğrenilmiş" örüntüleri kullanarak orijinale benzeyen ancak yeni ve özgün olan içerikler oluştururlar. Örneğin, daha önce hiç var olmamış ancak gerçekçi görünen "Ay üzerinde bir kale" görseli oluşturulabilir.

Bununla birlikte, üretken yapay zekânın gelişimine rağmen birçok zorluk bulunmaktadır. Yanlış bilgiler, olayların veya süreçlerin hatalı anlaşılmasına yol açabilir. Sahte görseller (deepfake), istenmeyen önyargılar ve telif hakları gibi konular özel dikkat gerektirmektedir. Özellikle bu sistemlerin eğitim, gazetecilik veya sosyal medya alanlarında kullanılması durumunda, üretilen içeriklerin tespiti ve kontrolü için etik kurallara, yasal düzenlemelere ve denetim araçlarına ihtiyaç vardır.

Yapay zekâ, genel ve üretken biçimleriyle, yaşamın farklı alanlarını geliştirme konusunda büyük potansiyele sahip bir araç olarak ortaya çıkmaktadır. Genel yapay zekâ hâlâ geleceğe yönelik bir vizyon olarak geliştirilmeye devam ederken, üretken yapay zekâ günümüzde birçok alanda aktif olarak kullanılmaktadır. Bu teknolojilerin bilinmesi, modern dijital dünyayı anlamak ve yapay zekâyı sorumlu bir şekilde kullanmak açısından gereklidir. Uygun kullanım ve düzenlemeler sayesinde yapay zekâ, eğitim, bilim, sanat ve günlük yaşam için güçlü bir yardımcı hâline gelebilir.

Bunun yanında yapay zekâ, YouTube, Netflix ve Amazon gibi platformlarda film, müzik veya ürünlere yönelik kişiselleştirilmiş içerik önerilerinde de kullanılmaktadır. Algoritmalar kullanıcı davranışlarını takip eder, verileri analiz eder ve seçimleri kolaylaştıran, kullanıcı memnuniyetini artıran öneriler oluşturur. Bu sistemler giderek daha gelişmiş hâle gelmekte, yalnızca bireysel kullanıcıdan öğrenmekle kalmayıp binlerce farklı kullanıcının tercihlerini karşılaştırarak en uygun içerikleri sunmaktadır.

Sağlık alanında üretken yapay zekâ, tıbbi verilerin analiz edilmesi, raporların oluşturulması, yeni ilaçlar için molekül modellerinin geliştirilmesi ve belirli tedavilerin etkilerinin simüle edilmesi amacıyla kullanılmaktadır. Bu sayede araştırma süreçleri hızlandırılmakta ve teşhis yöntemleri geliştirilmektedir. Bazı yapay zekâ sistemleri, hastalık belirtilerini insan uzmanlara benzer veya daha yüksek doğruluk oranlarıyla tanımlama yeteneği göstermektedir.

Yapay zekâ kullanılırken küresel düzeydeki önerilere, Avrupa Birliği'nin yasal düzenlemelerine ve ulusal mevzuatlara uyulmalıdır. Sahip olduğu güçlü imkânlar kötüye kullanılmamalıdır. Yapay zekâ kullanırken AŞAĞIDAKİ önerilere uyulmalıdır:

1. Gizlilik ve Kişisel Verilerin Korunması

- Yapay zekâ, izin olmadan kişisel bilgileri toplamamalı veya işlememelidir.
- Veriler kötüye kullanım, hırsızlık veya yetkisiz erişime karşı korunmalıdır.
- Her kullanıcı, hangi verilerin toplandığını ve neden toplandığını bilmelidir.

2. Şeffaflık (açıklık ve anlaşılabilirlik)

- Yapay zekâ sistemleri çalışma biçimleri konusunda açık ve anlaşılır olmalıdır.
- Kullanıcı, bir insanla değil, bir yapay zekâ sistemiyle iletişim kurduğunu bilmelidir.
- Yapay zekânın belirli bir kararı nasıl verdiğini açıklayabilmesi önemlidir (örneğin, bir öğrenciye neden belirli bir not verdiğini açıklaması).

3. Adil Davranış ve Ayrımcılığın Olmaması

- Yapay zekânın eğitildiği veriler çeşitli ve dengeli olmalıdır.
- Yapay zekâ, ırk, cinsiyet, yaş, din veya diğer özelliklere göre önyargılı davranmamalıdır.

İnsan Kontrolü ve Sorumluluk

- Kullanıcılar, yapay zekâ tarafından verilen kararların nihai sorumluluğunu her zaman taşımalıdır.
- Yapay zekâ, özellikle sağlık veya eğitim gibi hassas alanlarda, insan müdahalesi olmadan karar vermemelidir).

Güvenlik ve Güvenilirlik

- Yapay zekâ sistemleri siber saldırılara, kötüye kullanıma veya hatalara karşı korunmalıdır.
- Güvenli çalışmaları için düzenli olarak test edilmeli ve güncellenmelidir.

Amaca Uygun Etik Kullanım

- Yapay zekâ, insanların yararı için kullanılmalıdır, eğitim, sağlık, çevrenin korunması gibi alanlarda insanlara katkı sağlamalıdır.
- Sahte haberler, manipülasyon, izinsiz gözetim veya sivillere karşı askerî amaçlar için kullanılmamalıdır.

İnsan Onuruna ve Haklarına Saygı

- Yapay zekâ, ifade özgürlüğü, gizlilik, düşünce özgürlüğü gibi temel insan haklarına saygı göstermelidir.
- Yapay zekâ, insanları değersizleştirecek veya onların önemini azaltacak şekilde kullanılmamalıdır.

Yapay zekânın etik kullanımı, sorumluluk, şeffaflık, insan odaklılık, adil davranış ve güvenlik ilkelerini ifade eder. Teknoloji insanlara hizmet etmelidir, insanlar teknolojiye değil.



Resim 13 Yapay zekâ tarafından oluşturulan görseller



SONUÇ

Yapay zekâ (YZ), bilgisayar sistemlerinin normalde insan zekâsı gerektiren görevleri gerçekleştirebilme yeteneğidir. Bu görevler arasında öğrenme, görüntü ve konuşma tanıma, karar verme ve problem çözme yer alır. Günlük yaşamda YZ, video ve müzik önerileri, navigasyon, Siri ve Google Asistan gibi akıllı yardımcılar, spam filtreleri, yüz tanıma sistemleri, otomatik çevirmenler ve sohbet botları gibi birçok alanda kullanılmaktadır. Genel yapay zekâ, insan gibi düşünebilen, öğrenebilen ve birçok farklı durumda uyum sağlayabilen bir yapay zekâ türü olarak tasarlanmaktadır. Ancak günümüzde henüz gerçek anlamda mevcut değildir ve yalnızca teorik bir fikirdir.

Araştırmacılar, bağlamı anlayabilen, yeni koşullara uyum sağlayabilen ve yalnızca belirli bir görevi değil, farklı görevleri çözebilen sistemler geliştirmeye çalışmaktadır. Amaç, insan zihni gibi birçok alanda çalışabilecek bir zekâ geliştirmektir.

Üretken yapay zekâ ise eğitildiği verilere dayanarak yeni metinler, görseller, müzikler, videolar veya kodlar oluşturabilme yeteneğine sahiptir. Bir makale yazabilir, bir problemi çözebilir, fotoğraf oluşturabilir veya yeni fikirler ve içerikler geliştirebilir.

- **Genel yapay zekâ örnekleri:** Otomobillerdeki otopilot sistemleri, hava durumu tahmin sistemleri, tıbbi analiz sistemleri, spam filtreleri, film öneri sistemleri.
- **Üretken yapay zekâ örnekleri:** ChatGPT, DALL•E, Midjourney, Bard ve müzik veya 3D model oluşturma sistemleri.

Yapay zekânın kullanım alanları arttıkça, etik kullanım kuralları da büyük önem kazanmaktadır. Yapay zekâ sistemleriyle kişisel verilerin kontrollü bir şekilde paylaşılması ve yapay zekâ tarafından üretilen bilgilerin doğruluğunun kontrol edilmesi gerekmektedir, çünkü bu bilgiler hatalı olabilir. Eser sahipliği konusu da giderek daha önemli hâle gelmektedir. Başkalarına ait fikirlerin izinsiz kopyalanması kesin olarak düzenlenmiştir. Yapay zekâ öğrenme amacıyla kullanılmalı, kopya çekmek amacıyla kullanılmamalıdır.

Genel ve üretken yapay zekâ ile ilgili etik ikilemler giderek daha fazla önem kazanmaktadır. Bu konular arasında, bazı sistemlerin çok fazla veri toplaması nedeniyle ortaya çıkan gizlilik sorunları bulunmaktadır. Yapay zekânın hatalar veya yanlış bilgiler oluşturabilmesi nedeniyle doğruluk sorunu da önemli bir tartışma konusudur. Ayrıca yapay zekâyı bağımlı hâle gelme ve bağımsız düşünmenin azalması, izin verilmediği hâde okul ödevlerinde yapay zekâ araçlarının kullanılmasının adil olup olmadığı gibi konular da etik tartışmalar arasında yer almaktadır. Diğer etik ikilemler arasında, üretken yapay zekânın yanlış bilgiler, sahte fotoğraflar veya sahte ses kayıtları oluşturmak amacıyla kötüye kullanılmasının nasıl önleneceği ve makinelerin iş yerlerinde insanların yerini almasının etik olup olmadığı bulunmaktadır.

ÖRNEK 1



Metin Oluşturma

Çevrim içi bir platform, uzun makalelerin otomatik olarak özetlenmesi için üretken yapay zekâ kullanmaktadır. Kullanıcı metni girer, yapay zekâ metni işler ve kısa, öz bir sürümünü oluşturur.

ÖRNEK 2



Tıpta Yapay Zekâ

Röntgen görüntülerini analiz eden bir sistem, akciğerlerdeki olası hastalıkları tespit etmek için yapay sinir ağı kullanmaktadır. Tanı daha sonra, yapay zekâyı destekleyici bir araç olarak kullanan doktor tarafından doğrulanır.

ÖRNEK 3



Sanal Sanatçı

Bir öğrenci, şu açıklamaya dayalı dijital bir poster oluşturmak için üretken bir model kullanır: "Mor dağlara ve iki güneşli bir gökyüzüne sahip fantastik bir manzara." Yapay zekâ birkaç saniye içinde bir görsel oluşturur.

BİLGİ KONTROLÜ SORULARI VE ETKİNLİKLER



1. Yapay zekânın ne olduğunu kendi sözlerinle açıkla!
2. Genel yapay zekâ ile üretken yapay zekâ arasındaki farklar nelerdir?
3. Üretken yapay zekânın kullanıldığı üç gerçek örnek ver!
4. "Yapay sinir ağı" kavramı neyi ifade eder ve işlevi nedir?
5. Günümüzde hangi üretken yapay zekâ uygulamalarının bulunduğunu ve ne gibi olanaklar sağladıklarını araştır!
6. Üretken yapay zekâ ile ilgili etik sorunlar nelerdir?
7. "Deepfake" videolar nedir ve neden tehlikeli olabilir?
8. Yapay zekânın gelecekte sınıfta nasıl kullanılacağını düşün. Bir örnek açıkla!



DAHA FAZLA BİLGİ EDİNMEK İSTEYENLER İÇİN

GPT gibi üretken dil modellerinin nasıl eğitildiğini ve eğitim süreçlerinde hangi verilerin kullanıldığını araştır! Etik yapay zekâ ve yapay zekâ şeffaflığı kavramlarının ne anlama geldiğini ve teknolojinin gelecekteki gelişimi açısından neden önemli olduklarını incele!



UYGULAMALI ETKİNLİK

Hayal et, okulun AŞAĞIDAKİ amaçlarla bir yapay zekâ (YZ) sistemi kullanmak istiyor:

- Okula giriş için otomatik yüz tanıma,
- Güvenlik amacıyla teneffüslerde öğrencilerin kameralarla izlenmesi,
- Metin analizi ile yazılı ödevlerin otomatik değerlendirilmesi.

Etik bir değerlendirme yap.

AŞAĞIDAKİ soruları yanıtlayın:

1. Bu YZ sisteminde hangi etik sorunlar ortaya çıkabilir? (ör. gizlilik, önyargı, yanlış sonuçlar...)
2. Kim zarar görebilir veya ayrımcılığa uğrayabilir ve neden?
3. Hangi etik kural etkilenmektedir?
4. Hangi ikilemlerle karşılaşacaksınız?
5. Okul, YZ'nin sorumlu bir şekilde kullanılmasını nasıl sağlayabilir?
6. Sizce bu sistemi okulunuzda uygulamak iyi mi yoksa kötü mü olur? Neden?
7. Sorunu çözmek için ne yapardınız?

Doğru sonuca ulaşmak için AŞAĞIDAKİ görseldeki tabloyu kullanın.

No.	Durumun / ikilemin açıklaması	Etkilenen etik kurallar	Sorun nedir? (Çatışmanın açıklaması)	Olası çözümler (etik kurallara göre)
1				

Tablo 4 İkilemin çözülmesi

KONU 1: GENEL TEKRARLAMA

1. Günlük yaşamında kullandığın üç dijital teknoloji örneği ver!
2. Klasörlerin ve dosyaların ne olduğunu ve bilgisayarda verilerin düzenlenmesi açısından neden önemli olduklarını açıkla!
3. Masaüstünde “Fotoğraflarım” adlı yeni bir klasör oluştur ve içine bir fotoğraf ekle! Daha sonra klasörü belgeler klasörüne taşı ve istediğin bir isimle yeniden adlandır!
4. İki modern dijital teknolojiyi (örneğin: akıllı telefon ve akıllı saat) karşılaştır ve kullanım amaçlarındaki farklılıkları açıkla!
5. Metin yazma, tablo oluşturma ve sunum hazırlama için en uygun uygulamaları değerlendir! Neden uygun olduklarını açıkla.
6. İşletim sisteminin temel ayarlarını düzenlemek için kısa kullanıcı yönergeleri oluştur (güncellemeler – en yeni sistem düzeltmelerini kontrol etme ve yükleme, kullanıcı hesapları ve şifreler, yerel kullanıcı için 2FA ile güçlü bir şifre oluşturma, Wi-Fi ağına bağlanma, güvenlik duvarını (Firewall) etkinleştirme, yalnızca gerekli uygulamaların kamera/mikrofon/konum erişimine sahip olması için izinleri oluşturma, antivirüs programını başlatma ve sistem kontrolü yapma, öğrenme için ekranı optimize etme, netlik ve kontrast ayarları, uyku modu/ekran kapatma ayarlarını düzenleme ve gereksiz program ve uygulamaları kaldırma).
7. Genel yapay zekâ (General AI) ile üretken yapay zekâ arasındaki iki farkı belirt!
8. Dijital teknolojinin günümüz toplumunda eğitim, iletişim veya çalışma hayatını nasıl etkilediğini açıkla!
9. Farklı işletim sistemlerinin (Windows, Android, Linux) görünüm, çalışma ortamı ve özellikler açısından nasıl farklılaştığını analiz et!
10. Okulda üretken yapay zekâ uygulamak zorunda olduğunu düşün. Kullanılabileceği üç yöntemi öner, nasıl çalıştığını açıkla ve olası yararları ile risklerini belirt!



Handwritten notes on sticky notes:

- Yellow sticky note:** A diagram showing a floor plan with a central area labeled "Meeting Room" and surrounding areas labeled "Reception", "Kitchen", and "Bathroom".
- Pink sticky note:**
- 1st floor plan -
- 2nd floor plan -
- 3rd floor plan -
- 4th floor plan -
- 5th floor plan -



KONU 2: ÇEVİRİM İÇİ YAŞAM

Günümüzde başka bir ülkede yaşayan bir arkadaşınla iletişim kurmak, çevrim içi film izlemek, bir uygulama üzerinden doktor randevusu almak veya dünyanın her yerinden insanlarla sanal bir oyun oynamak oldukça kolaydır – tüm bunlar bilgisayar ağları ve internet sayesinde gerçekleşmektedir.

Günümüzde dünya hiç olmadığı kadar bağlantılı hâle gelmiştir. İster cep telefonu, ister dizüstü bilgisayar, ister akıllı saat veya endüstriyel robot olsun – tüm bu cihazlar ağlar aracılığıyla birbirleriyle “iletişim kurmaktadır”. Bu nedenle bilgisayar ağları ve internet hakkında bilgi sahibi olmak yalnızca yararlı bir beceri değil, aynı zamanda çevremizdeki dünyayı anlamaya yönelik en önemli adımlardan biridir.

Yeni Kavramlar

bilgisayar ağı, sunucu, istemci, LAN, WAN, internet, web, “internette gezinme”, internet hizmetleri, www, ftp, arama motorları (search engines), elektronik posta, görüntülü görüşmeler, e-ticaret, elektronik bankacılık, etkileşimli iletişim, dijital ayak izi, şiddet, nefret, uyuşturucu kullanımı veya kendine zarar verme davranışlarını teşvik eden içerikler, dezenformasyon, sahte haberler, güvenlik ve gizlilik ayarları, hukuki boyut, bildirim mekanizmaları, iyi olma hâli.

Bu konuyu tamamladığında...

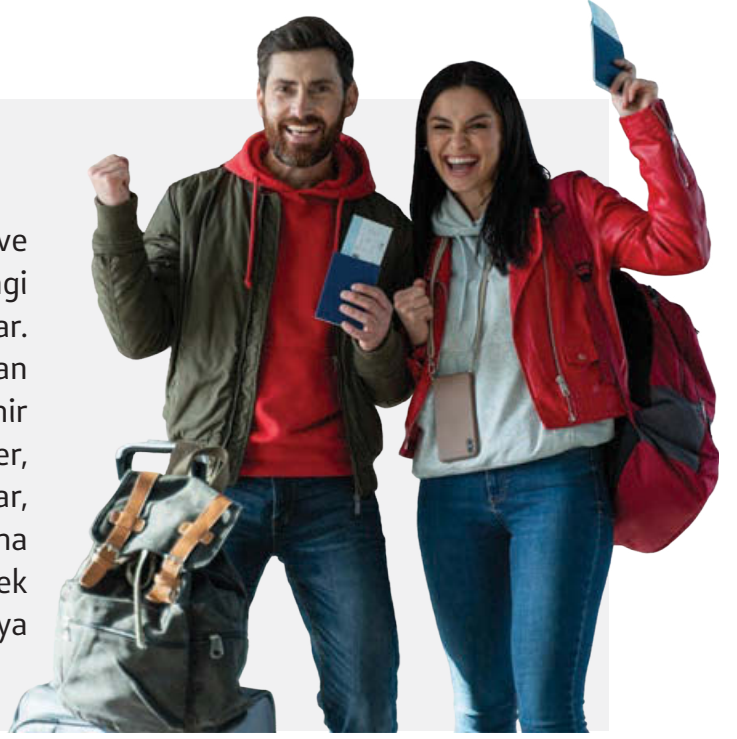
- ✓ Web arama motoru ile web tarayıcısı arasındaki farkı ayırt edebiliyorsun.
- ✓ İnternet kaynaklarına erişebiliyor ve gerekli bilgileri arayabiliyorsun.
- ✓ Web üzerindeki geçerli bilgi kaynaklarını tanıyabiliyorsun.
- ✓ “Dijital ayak izi”nin olumlu ve olumsuz yönlerini anlayabiliyorsun.
- ✓ İnternet üzerinden iletişim kurabiliyorsun.



TEKRARLAMA SORULARI:

Seyahat Macerası

Havaalanındasın. Valizin hazır ve elinde seni dünyadaki istediğin herhangi bir yere götürebilecek sihirli bir bilet var. Nereye gitmeyi seçerdin? Her zaman ziyaret etmek istediğin bir ülke ya da şehir seç. Oranın kültürü nasıldır? (Dil, gelenekler, yemekler). Oraya nasıl gidebilirsin? (Uçuşlar, fiyatlar vb.). Orada 3 günlük bir konaklama sana yaklaşık ne kadara mal olur? (Gerçek örnekler ver.). Orada mutlaka görmek veya yapmak istediğin üç şey nedir?



AŞAĞIDAKİ KONULARDA ŞUNLARI ÖĞRENECEKSİN:

- Bilgisayar ağı ve internet kavramlarını açıklayabileceksin.
- İnterneti, bilgi edinme ve bilgi paylaşma aracı olarak tanımlayabileceksin.
- Farklı internet hizmetlerini (servislerini) ayırt edebileceksin.
- Web üzerinde güvenilir (geçerli) bilgi kaynaklarını kullanabileceksin.
- İnternette bıraktığın dijital izin (dijital ayak izi)
- olumlu ve olumsuz yönlerini analiz edebileceksin.
- Bulduğun bilgilerden yararlanarak dijital
- içerik oluşturabileceksin.



Bilgisayar ağı, birbirleriyle iletişim kurabilmeleri ve kaynak paylaşımı yapabilmeleri için en az iki bilgisayarın birbirine bağlanması anlamına gelir. Kaynaklar, donanım bileşenlerini (bilgisayar, sunucu, tarayıcı, yazıcı vb.) ve yazılım bileşenlerini (oyunlar, veri tabanları, uygulamalar vb.) kapsar. Bilgisayarlar, verimliliği artırmak ve maliyetleri azaltmak (zamandan ve paradan tasarruf etmek) amacıyla bir ağ üzerinden birbirine bağlanır. Bu cihazlar, altyapıya ve kullanıcının ihtiyaçlarına bağlı olarak fiziksel kablolar aracılığıyla veya kablolu iletişim yoluyla birbirine bağlanabilir.

2.1.1 Bilgisayar Ağlarının Amacı

Bilgisayar ağları aracılığıyla kullanıcılar veri, yazılım ve donanımı paylaşır, ayrıca birbirleriyle iletişim kurarlar.

Veri ve bilgi paylaşımı. Ağ ortamında, ağdaki diğer bilgisayarlarda bulunan verilere erişmek mümkündür. Örneğin, bir şirkette bilgisayar ağı sayesinde tüm çalışanlar ortak toplantı takvimine veya müşterilere ilişkin bilgilerin yer aldığı bir veri tabanına erişebilir. Ev ortamında kurulan bir bilgisayar ağına ise aile bireylerinden biri kendi bilgisayarındaki bir filmi veya belgeyi paylaşabilir, diğer aile bireyleri tablet ya da başka bir bilgisayar kullanarak bu içeriklere doğrudan erişebilir. Böylece bu dosyaların kendi cihazlarına kopyalanmasına gerek kalmaz.

Yazılım paylaşımı. Bilgisayar ağına, başka bilgisayarlarda çalışan yazılımlar kullanılabilir. Örneğin, bir okulda grafik tasarım programı tek bir bilgisayara (sunucuya) kurulabilir. Programın her öğrencinin bilgisayarına ayrı ayrı kurulması yerine, öğrenciler okul ağı üzerinden bu programa erişebilir ve kendi projeleri üzerinde çalışabilirler.

Donanım paylaşımı. Ağ ortamında birden fazla kullanıcı donanım kaynaklarını (örneğin sunucu, yazıcı ve tarayıcı) ortaklaşa kullanabilir. Şirketlerde veya evlerde, daha yüksek kaliteli tek bir yazıcı çoğu zaman ağ üzerinden birden fazla bilgisayara bağlanır ve bu bilgisayarların her birinden aynı yazıcıya çıktı alınabilir.

İletişim. Bilgisayar ağları, insanların iletişim kurma biçimini değiştirmiştir. Günümüzde insanlar en çok e-posta, sosyal ağlar, anlık mesajlaşma hizmetleri, internet telefonu ve benzeri araçlar üzerinden iletişim kurmaktadır. Bu ağlar, iletişimin fiziksel uzaklıktan bağımsız olarak daha hızlı, ekonomik ve gerçek zamanlı gerçekleşmesini sağlar. Ağlar sayesinde yurt dışındaki arkadaşlarla iletişim kurulabilir, çevrim içi derslere katılınabilir, dünyanın farklı yerlerindeki kişilerle ortak projeler yürütülebilir ve fotoğraf ile videolar birkaç saniye içinde paylaşılabilir.

2.1.2 İnternet

İnternet, standartlaştırılmış iletişim protokollerini kullanan, birbirine bağlı ağlardan oluşan, çeşitli bilgi ve iletişim hizmetleri sağlayan küresel bir bilgisayar ağıdır. **İnternet, istemci-sunucu (client-server)** mimarisini kullanır.

Sunucu (server), istemcilerden gelen istekleri kabul eden ve ağ üzerinden diğer bilgisayarlara hizmet, kaynak veya bilgi sağlayan özel olarak yapılandırılmış bir bilgisayardır. Sunucular sürekli çalışır durumdadır. Yüksek işlem gücüne ve geniş depolama kapasitesine sahip olan bu güçlü sistemler, güvenilirlik ve kesintisiz çalışma amacıyla tasarlanmış olup çoğunlukla veri merkezlerinde bulunur. Sunucular, sanal makineler, bulut hizmetleri veya yazılım bileşenleri şeklinde de olabilir.

Sunucuların sağlayabileceği hizmetlerden bazıları şunlardır:

- Dosyaların saklanması ve paylaşılması (dosya sunucusu),
- Web sayfalarının sunulması (web sunucusu),
- E-posta gönderme ve alma hizmetlerinin sağlanması (e-posta sunucusu),
- Verilerin yönetilmesi (veritabanı sunucusu).

İstemci (client), sunucudan bilgi veya hizmet almak amacıyla iletişimi başlatan cihazdır (dizüstü bilgisayar, telefon, tablet, web tarayıcısı vb.). Örneğin, bir kullanıcı bir web sayfasını ziyaret etmek için internet tarayıcısını kullandığında, kullanıcının bilgisayarı istemci, web sayfasını barındıran bilgisayar ise sunucudur. Bilgisayar ağlarında istemci ile sunucu arasında her zaman çift yönlü iletişim bulunur.

İstemci (örneğin web tarayıcısı), sunucuya bir istek gönderir. Sunucu ise istenen içeriği bulur ve istemciye geri gönderir. Bu iletişim, HTTP, HTTPS ve TCP/IP gibi protokoller aracılığıyla gerçekleştirilir. İnternete bağlı her cihazın, kimliklendirme ve verilerin yönlendirilmesi amacıyla kullanılan benzersiz bir IP adresi vardır.

Bunun yanı sıra, ağlar ihtiyaçlara göre farklı biçimlerde yapılandırılır. Örneğin, etkin cihazların sayısının izlenmesi, erişim denetimi, ağ trafiğinin önceliklendirilmesi ve etkinliklerin izlenmesi gibi ayarlamalar yapılabilir. Eğitim kurumlarında bilgisayar ağları, merkezi yönetim, internet erişiminin denetlenmesi, kaynakların paylaşılması ve öğrenci ile öğretmenler arasında iş birliğinin sağlanmasına olanak tanır. Küresel ölçekte internet, birbirleriyle ortak bir dil olan TCP/IP protokolünü kullanan yüz binlerce birbirine bağlı ağdan oluşur.

DNS (Domain Name System – Alan Adı Sistemi), insanların kolayca okuyabildiği web sitesi adlarını (örneğin www.wikipedia.org) makinelerin anlayabileceği IP adreslerine dönüştürür. Bu sayede internet daha erişilebilir ve kullanımı daha kolay hâle gelir. Veri paketleri ağ üzerinden farklı yollar izleyerek iletilir ve bu süreçte her adım, verilerin verimli biçimde aktarılmasını sağlayan yönlendiriciler (router) ve anahtarlar (switch) tarafından kontrol edilir.

Yönlendiriciler (router), verilerin internet üzerinden yönlendirilmesinde merkezi bir rol oynar. Gönderilen her veri paketi için en uygun yolu belirleyerek verinin hedefe hızlı ve güvenli bir şekilde ulaşmasını sağlarlar. Belirli yolların aşırı yüklenmesi veya kesintiye uğraması, veri iletiminde gecikmelere ya da veri kaybına neden olabilir. Bu nedenle iletişim protokolleri, hata denetimi, verilerin yeniden gönderilmesi ve hata düzeltme mekanizmalarını içerir.

2.1.3 Bilgisayar Ağı Türleri

Bilgisayar ağlarının yapısını ve kullanım alanlarını anlayabilmek için ağlar, büyüklüklerine ve kapsama alanlarına göre sınıflandırılır. En yaygın olarak iki ana kategoriye ayrılır:

- **LAN (Local Area Network)** – Yerel Alan Ağı,
- **WAN (Wide Area Network)** – Geniş Alan Ağı (farklı mahalleler, aynı şehirdeki farklı bölgeler, farklı şehirler vb. gibi geniş coğrafi alanlara yayılan ağ)

Yerel Alan Ağı – LAN

LAN, evler, sınıflar, okullar, ofisler veya bir bina gibi sınırlı bir coğrafi alan içerisindeki farklı cihazların (bilgisayarlar ve çevre birimleri) birbirine bağlanmasını sağlayan yerel bilgisayar ağıdır.

LAN ağı basit bir yapıda olabilir (iki bilgisayardan oluşan bir ağ gibi) veya karmaşık bir yapıya sahip olabilir (büyük bir şirkette yüzlerce bilgisayar ve çevre biriminden oluşan ağ gibi)

WAN Bilgisayar Ağı

WAN, geniş bir coğrafi alana yayılan bilgisayar ağıdır. Uzak konumlarda bulunan bilgisayarların birbirine bağlanması amacıyla kullanılır.

WAN bilgisayar ağı, daha küçük ağları birleştirerek birbirine bağlar. Böylece bir ağdaki kullanıcılar başka bir ağdaki kullanıcılarla iletişim kurabilir veya uzak bir konumdaki kaynakları kullanabilir.

Özellik	LAN	WAN
Kapsama alanı	Küçük alan	Büyük mesafeler
Hız	Hızlı (Gbps)	LAN'dan daha yavaş (yeterince hızlı)
Fiyat	Düşük	Yüksek (hat kiralama, özel altyapı)
Mülkiyet	Özel (kuruluş, ev vb.)	Kamuya açık veya özel (internet sağlayıcısının altyapısı)
Ekipman	Switchler, routerlar	Fiber optik hatlar, uydular, telekomünikasyon ağları
Yönetim	Basit	Karmaşık, sağlayıcıların desteğiyle
Örnek	Ev ağı	Birden fazla uzak şubesi olan şirket, internet ağı

Tablo 1 Yerel ve küresel LAN ve WAN ağları arasındaki farklar



NOT:

Bilgisayar ağıları internet ile aynı şey olarak değerlendirilmemelidir. Örneğin, bir okul laboratuvarında bilgisayarlar birbirine bağlanarak veri alışverişi yapabilir, ancak bu ağ internete bağlı olmayabilir.



SONUÇ

Bilgisayar ağıları, bilgi, kaynak ve hizmetlerin paylaşılması amacıyla birden fazla bilgisayarın ve cihazın birbirine bağlanmasını sağlar. Sunucu, hizmetleri, verileri veya kaynakları sağlar, istemci ise bu hizmetleri istek göndererek kullanır. Yerel ve küresel ağlar arasındaki fark, cihazların ne kadar geniş bir alanda birbirine bağlanabileceğini gösterir – küçük bir sınıftan tüm dünyaya kadar.

ÖRNEK 1



Okulda LAN Ağı

Bir okulda, tüm sınıflar internet erişimi, merkezi bir yazıcı ve öğretim materyallerinin bulunduğu ortak bir veri tabanına erişim sağlayan yerel bir ağ (LAN) üzerinden birbirine bağlanmıştır.

ÖRNEK 2



DNS ve Web Sitesi Arama

Bir öğrenci web tarayıcısına «www.wikipedia.org» yazdığı anda, DNS sunucusu bu adresi bir IP numarasına dönüştürür ve isteği, içeriğin bulunduğu uygun web sunucusuna yönlendirir.

ÖRNEK 3



Evde Wi-Fi Kullanımı

Ev ortamında cihazlar, Wi-Fi yönlendirici (router) aracılığıyla kablosuz olarak birbirine bağlanır. Bu cihaz, cep telefonu, dizüstü bilgisayar ve dış internet bağlantısı arasındaki internet trafiğini yönlendirir.

BİLGİ KONTROLÜ SORULARI VE ETKİNLİKLER



1. Bilgisayar ağı, bilgisayar sunucusu ve istemci kavramlarını açıklayınız!
2. LAN ve WAN ağı arasındaki fark nedir?
3. Günlük hayatta kullanılan sunuculara iki örnek veriniz!
4. Bir web sayfasını açmak için internet tarayıcısı kullanıyorsunuz. Bu durumda hangi cihaz istemcidir, hangisi sunucudur?
5. Sunucuların sürekli aktif olması, istemcilerin ise sürekli aktif olmasının gerekmemesi neden önemlidir?
6. Her bilgisayar sunucu olabilir mi? Görüşünüzü açıklayınız!
7. Bir okul için basit bir ağ tasarlayınız! Sunucuları ve istemcileri nasıl düzenleyeceğinizi açıklayınız!



DAHA FAZLA BİLGİ EDİNMEK İSTEYENLER İÇİN

Bir web adresini tarayıcıya girdiğinde ve Enter tuşuna bastığında arka planda neler olduğunu araştır. Cihazından sunucuya ve geri dönüş yolunda veri aktarım sürecinin bir haritasını oluşturmaya çalış. Ek olarak, ağlarla ilgili alanlarda oluşturulabilecek kariyerleri incele – ağ yöneticisi, siber güvenlik mühendisi veya BT destek uzmanı gibi.



İnternet, günlük yaşamı kolaylaştıran çok geniş bir hizmet yelpazesine erişim sağlar. Bu internet hizmetleri, iletişim, bilgi edinme, eğitim, alışveriş, bankacılık ve daha birçok alanda kullanılır. Bu hizmetlerin yapısını, amacını ve çalışma biçimini anlamak, doğru, güvenli ve etkili kullanım için önemlidir. Bu derste en önemli internet hizmetleri, nasıl çalıştıkları ve avantajları ile riskleri açıklanacaktır.

WWW

En yaygın kullanılan internet hizmeti World Wide Web (WWW)'dir. Bu hizmet, metin, resim, video ve etkileşimli içeriklere sahip web sayfalarının görüntülenmesini sağlar. Web sayfalarına web tarayıcıları (örneğin Chrome, Firefox, Edge) aracılığıyla erişilir ve içerikler HTTP ve HTTPS protokolleri kullanılarak aktarılır.

Web sayfaları sunucularda (web sunucularında) bulunur ve birbirine bağlı web sayfaları grubunu oluşturur. Web üzerindeki her konumun, URL (Uniform Resource Locator) adı verilen benzersiz bir adresi vardır. WWW, öğrenme, araştırma, eğlence ve forumlar, portallar ile sosyal ağlar aracılığıyla iletişim kurmak için kullanışlıdır.

Bu internet hizmetleri hızlı bir şekilde gelişmekte, kullanıcı deneyimini iyileştirmek amacıyla sürekli olarak yeni özellikler eklenmektedir. Kullanıcılar, özellikle kişisel veya finansal bilgilerin girilmesinin gerektiği durumlarda, yeni olanaklar ve güvenli kullanım yöntemleri hakkında sürekli bilgi sahibi olmalıdır.

Önemli: WWW, internet ile aynı şey değildir – internet üzerinden çalışan bir hizmettir. İnternet, birbirine bağlı bilgisayarlardan oluşan bir ağ iken, WWW bu ağ aracılığıyla eriştiğimiz uygulamalar veya dijital içerikler bütünüdür.

Arama Motorları (Search Engines)

Günümüzde internette milyonlarca web sayfası bulunmaktadır. Bu durum, bu sayfaların bulunmasını bazen karmaşık ve zaman alıcı hâle getirir. İnternet, devasa, sınırsız ve neredeyse tükenmeyen bir bilgi kaynağı hâline gelmiştir. İçerikleri daha kolay bulmak için web dizinleri (sayfaları konu alanlarına göre gruplandıran sistemler) veya daha yaygın olarak arama motorları (search engines) kullanılabilir.

Bir web arama motoruna (Google, Edge, Yahoo, Bing), anahtar kelimelerden oluşan bir arama isteği girildiğinde, internette arama yapan bir algoritma etkinleşir. Bu isteğe yanıt olarak arama motoru, girilen anahtar kelimeleri içeren web sayfalarının listesini görüntüler. Adresleri ve kısa açıklamaları arama sonuçlarında gösterilen bu web sayfalarından herhangi birini, arama motorunun sunduğu sonuçlarda yer alan sayfa adına tıklayarak ziyaret edebilirsiniz.

Elektronik Posta

Elektronik posta, internetin en eski ve en yaygın kullanılan hizmetlerinden biridir. Bilgisayar ağları aracılığıyla kullanıcılar arasında metin mesajlarının ve farklı türde dosyaların alışverişini sağlar. Geleneksel postanın dijital biçimini temsil eder ancak çok daha hızlı, kullanışlı ve erişilebilirdir.

Elektronik postanın kullanılabilmesi için her kullanıcının kendi elektronik posta adresine sahip olması gerekir. E-posta adresi kullanıcı tarafından seçilir ve e-posta sunucusu tarafından onaylanır.

Her elektronik posta adresi için, kullanıcının elektronik mesajlarının saklandığı bir elektronik posta kutusu oluşturulur. E-posta sunucusu, internete bağlı olan ve elektronik posta kullanıcılarının mesajlarını alan ve gönderen bir bilgisayardır.

E-posta adresi iki bölümden oluşur: kullanıcı adı ve e-posta sunucusunun adı. Bu iki bölüm @ işareti ("at" olarak okunur, ayrıca halk arasında "maymun işareti" olarak da adlandırılır) ile ayrılır. Kullanıcı adı, e-posta adresi oluşturulurken kullanıcı tarafından seçilir, ancak bu adın daha önce aynı e-posta sunucusunun başka bir kullanıcısı tarafından alınmamış olması gerekir. @ işaretinden sonraki bölüm ise kullanıcının elektronik posta hizmetinin gerçekleştiği e-posta sunucusunun adıdır. Örneğin, maja_matevska@t-home.mk e-posta adresinde kullanıcı adı maja_matevska, sunucu adı ise t-home.mk'dir. Elektronik posta kullanılırken küçük ve büyük harflerle yazılan e-posta adresleri aynıdır.

Örneğin: biljana@gmail.com=BILJANA@GMAIL.COM.

Elektronik posta mesajının yapısı:

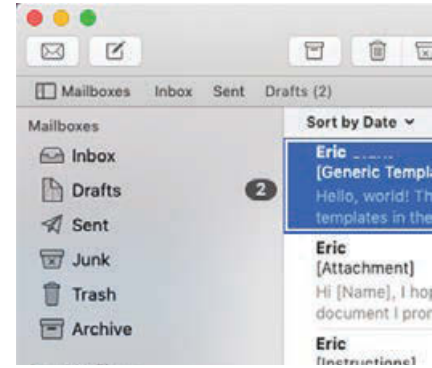
- Gönderen adresi (**From**): Otomatik olarak doldurulur.
- Alıcı adresi (**To**): Adres bölümüne yazılır.
- **Carbon Copy (cc)**: Mesajın bir kopyasını alacak başka bir kişinin e-posta adresi girilir. Alıcı, mesajın başka bir kişiye de gönderildiğini görür.
- **Blind Carbon Copy (bcc)**: Mesajın alıcısı, bcc alanında adresi belirtilen kişilere de mesajın bir kopyasının gönderildiğini bilmez.
- Mesaj başlığı – konu (**Subject**): İçeriğin kısa açıklamasıdır. Mesajın alıcısı, gönderenin adı (adres) ve tarih bilgisinin yanı sıra mesajın konusunu da görür. Alıcı, bu bilgilere dayanarak mesajı okuyup okumayacağına ve ne zaman okuyacağına karar verir.
- Mesaj metni: Mesajın yazıldığı bölümdür. Mesajın içeriği, kime gönderildiğine bağlı olarak değişir. Ek (**Attachment**): Mesaja eklenen dosyadır. Belge, resim, sunum, program vb. olabilir.
- **İmza (Signature)**.



Resim 1 Elektronik Mesaj

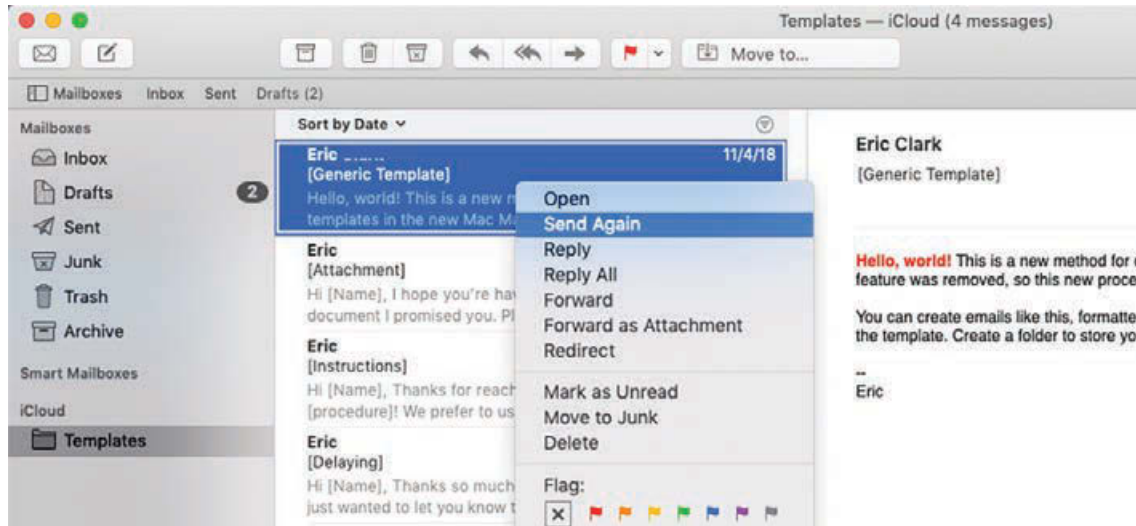
Elektronik posta kutuları genellikle AŞAĞIDAKİ gibi standart klasörleri içerir:

- **Inbox (Gelen Kutusu):** Alınan mesajların bulunduğu klasör.
- **Sent (Gönderilenler):** Gönderilen mesajların bulunduğu klasör.
- **Trash (veya Deleted – Çöp Kutusu):** Silinen mesajların bulunduğu klasör.
- **Drafts (Taslaklar):** Gönderilmemiş ve düzenleme aşamasındaki mesajların bulunduğu klasör).



Resim 2 Elektronik Posta Kutusunun Yapısı

Tüm e-posta programları, e-postaların düzenlenmesine olanak sağlar. Kullanıcılar yeni klasörler oluşturabilir, mesajları bir klasörden başka bir klasöre taşıyabilir veya kopyalayabilir ve mesajların alındıktan hemen sonra belirli klasörlere ayrılmasını sağlayacak kurallar belirleyebilir.



Resim 3 Elektronik posta kutusunun tam yapısı

Elektronik posta programları AŞAĞIDAKİ işlemlere sahiptir:

- Elektronik posta oluşturma – Yeni mesaj (New).
- Mesaj gönderme – Gönder (Send).
- Alınan mesaja cevap verme – Yanıtla (Reply).
- Alınan mesajı başka bir kişiye gönderme – İlet (Forward).
- Mesajı silme – Sil (Delete).
- Mesajı yazdırma – Yazdır (Print).
- Mesaja başka belgeler ekleme – Ekle (Attach).
- Adres defterini kullanma.

ftp

FTP, File Transfer Protocol ifadesinin kısaltmasıdır ve dosya aktarım protokolü anlamına gelir. İnternet üzerindeki en eski hizmetlerden biridir. İki bilgisayar arasında internet üzerinden dosya gönderilmesini ve alınmasını sağlar. Bu işlem genellikle istemciden (senin bilgisayarından) sunucuya ve tam tersi yönde gerçekleşir.

Görüntülü Görüşmeler

Görüntülü görüşmeler, internet üzerinden başka bir kişiyle canlı olarak görüntülü ve sesli iletişim kurmayı sağlayan bir iletişim yöntemidir. Bunun için AŞAĞIDAKİler gereklidir:

- Cihaz – bilgisayar, dizüstü bilgisayar, tablet veya cep telefonu.
- Kamera (yerleşik veya harici) – görüntü için.
- Mikrofon ve hoparlörler (veya kulaklık) – ses için.
- İnternet bağlantısı – daha iyi bağlantı daha kaliteli görüşme sağlar.
- Görüntülü görüşme uygulaması veya platformu.

Giderek daha fazla kullanılan diğer hizmetler ise, verilerin internet sunucularında saklanması ve her yerden erişilmesini sağlayan bulut hizmetleridir (Cloud Services). Bunlara Google Drive, OneDrive, Dropbox örnek verilebilir. Bunun yanında, internet üzerinden video izleme, müzik dinleme veya canlı yayın yapmayı sağlayan akış hizmetleri (Streaming Services) de bulunmaktadır. En bilinenleri YouTube, Netflix, Spotify ve Twitch'tir

Elektronik Ticaret

Günümüzün dijital dünyasında, internetin ve modern teknolojilerin gelişmesi sayesinde giderek daha fazla insan, cihazları üzerinden sadece birkaç tıklamayla ürün ve hizmet satın almayı tercih etmektedir. Bu ticaret biçimine elektronik ticaret (veya e-ticaret) adı verilir.

Elektronik ticaret, fiziksel uzaklığa bakılmaksızın internet üzerinden alıcılar ve satıcılar arasında iletişim kurulmasını ve ürün alışverişi yapılmasını sağlar. Güvenli web siteleri, çevrim içi mağazalar, mobil uygulamalar ve elektronik ödemeler, bu ticareti hızlı, pratik ve erişilebilir hâle getiren araçlardan sadece birkaçıdır. Bu ticaret türü şunları kapsar:

- Şirketlerin bireysel müşterilere mal ve hizmet satışı – B2C (business to customer),
- Şirketlerin diğer şirketlere mal ve hizmet satışı – B2B (business to business),
- Satıcıların kendi mal ve hizmetlerini bireysel müşterilere sattığı satış türü – C2C (customer to customer).

Günümüzde birçok web sitesi çeşitli ürünlerin alım ve satımına olanak sağlamaktadır. Satın alma işlemi, alıcının mal ve hizmetlerin ücretini kredi kartı, teslimat sırasında nakit ödeme veya standart ödeme belgesi ile ödemesiyle gerçekleştirilir. Satıcı, ürünü posta veya kurye hizmeti aracılığıyla teslim eder. Dünyadaki birçok hizmet sağlayıcısı elektronik ticaret olanağı sunmaktadır. Örneğin, günümüzde bir seyahat tamamen internet üzerinden organize edilebilir. Uçak, otobüs veya tren biletleri en kolay şekilde ulaşım şirketlerinin web siteleri üzerinden satın alınabilir. Kullanıcı, hem elektronik hem de basılı biçimde kullanılabilen (basılması önerilmez) bir elektronik bilet alır ve pasaportuyla uçak, otobüs veya tren yolculuğu yapabilir. Oteller ve hosteller, kendi web siteleri üzerinden veya bu tür hizmetlerde uzmanlaşmış, güvenilir ve tanınmış web siteleri aracılığıyla daha güvenli bir şekilde rezerve edilebilir. E-ticaret hizmetleri kullanılırken özellikle güvenliğe dikkat edilmelidir. Gerçek para harcamayı gerektirdiği için dikkatsizlik birçok olumsuz sonuca yol açabilir.

Elektronik Bankacılık

Günümüzde bankalar elektronik bankacılık hizmetleri sunmaktadır. Elektronik bankacılık (veya e-bankacılık), bankacılık hizmetlerine erişmek için internetin ve dijital cihazların kullanılmasıdır. Bu hizmet, kullanıcıların fiziksel olarak banka şubesine gitmesine gerek kalmadan bilgisayar, cep telefonu veya ATM aracılığıyla finansal işlemlerini yönetmesine olanak sağlar. Sisteme giriş yaptıktan sonra hesap bakiyeni kontrol edebilir, para yatırma, para çekme işlemleri yapabilir ve bir hesaptan başka bir hesaba para aktarabilirsiniz. Elektronik fatura ödeme hizmeti oldukça kullanışlıdır, çünkü her şeyi evden ödeyebilir ve postanede veya bankada sıra beklemek zorunda kalmazsınız. Güvenlik önemli bir konudur. Bu nedenle bankalar, kullanıcı verilerini korumak amacıyla şifreleme, iki faktörlü kimlik doğrulama ve mobil token gibi güvenlik sistemleri kullanır.

Etkileşimli İletişim

İnternet hizmetleri etkileşimli iletişimi mümkün kılar. Canlı sohbetler (Chat), görüntülü görüşmeler ve video konferansların yanı sıra internet iletişimi, forumları, çevrim içi tartışmaları ve sosyal ağları da kapsar.

Forumlar, kullanıcıların tartışma yapmasına olanak sağlayan web siteleridir (genellikle yalnızca forum kurallarında belirlenen konular hakkında). Tartışmalar konular şeklinde düzenlenir. Bir kişi belirli bir konu hakkında ilk mesajı yayınlayarak bir konu başlığı başlatır ve bu konu hakkındaki sonraki tartışmalar, o konu başlığındaki mesajlara verilen yanıtlarla devam eder.

Bloglar (web LOG – blog), kullanıcıların belirli bir konu hakkındaki düşüncelerini yayınladıkları ve kamuoyu ile paylaştıkları bir tür kişisel günlüklerdir. Kullanıcılar, blog yazarları tarafından yayınlanan yazıları okur ve bunlar hakkında herkese açık şekilde yorum yapabilir ve tartışabilir.

Sosyal ağlar, 21. yüzyılın ilk on yılının ortalarında oluşturulmuş ve kısa sürede bir tür sosyolojik fenomen hâline gelmiştir. On yıldan daha kısa bir süre içerisinde aktif kullanıcı sayısı birkaç milyara ulaşmıştır. Sosyal ağlar, kullanıcıların kişisel arkadaşlarının ve tanıdıklarının hesaplarıyla veya kamuya açık alandaki kişilerin hesaplarıyla bağlantı kurmasına ve iletişim oluşturmaya olanak sağlar. Bireysel sosyal ağların kendilerine özgü özellikleri vardır:

 Facebook – en büyük sosyal ağıdır, durum paylaşımları, fotoğraflar, videolar paylaşmayı ve arkadaşlar, gruplar ve sayfalar ile bağlantı kurmayı sağlar.	 Instagram – fotoğraf ve kısa video paylaşma platformudur. Gençler arasında popülerdir. Ayrıca influencerlar ve işletmeler tarafından da kullanılır.	 TikTok – müzik, efektler ve meydan okumalar içeren kısa, yaratıcı videolar için kullanılan bir ağıdır.
 Twitter (X) – kısa metinli paylaşımlar ("tweetler") için kullanılan bir platformdur. Haberler, görüşler ve hızlı iletişim için idealdir.	 Snapchat kısa metinli paylaşımlar ("tweetler") için kullanılan bir platformdur. Haberler, görüşler ve hızlı iletişim için idealdir.	 LinkedIn – profesyonel bağlantı kurma, iş arama ve iş iletişimi için kullanılan profesyonel bir ağıdır.



YouTube

YouTube – video platformu olmasına rağmen yorumlar, paylaşımlar ve topluluklar aracılığıyla sosyal ağ özelliklerine de sahiptir.



Pinterest – çoğunlukla moda, sanat, yemek, tasarım vb. alanlarla ilişkili ilham verici görsellerin ve fikirlerin paylaşılması için kullanılan bir ağıdır.



Reddit – insanların farklı konular hakkında tartıştığı, sorular sorduğu ve görüşlerini paylaştığı forum tabanlı bir platformdur.

Sosyal ağların önemli bir yönü kullanıcıların gizliliğidir. Durum paylaşımı yaparken (örneğin yakınlarınızın fotoğraflarını paylaşıırken), bu içeriklerin geniş bir insan kitlesi tarafından görülebileceği göz önünde bulundurulmalıdır. Bilgilerinizin herkes tarafından görülmesini istemiyorsanız, bunları yalnızca arkadaşlarınızın görebileceği şekilde ayarlamanız daha uygundur. Sosyal ağ kullanıcılarının sayısı, bu platformların kullanımının son derece kolay olduğunu göstermektedir.

İnternet, öğrenme, araştırma, sosyalleşme ve eğlence için harika bir ortamdır. Ancak gerçek dünyada olduğu gibi internette de riskler bulunmaktadır. Bu nedenle kendinizi ve başkalarını korumak için güvenli iletişim kurallarına uymak önemlidir. Bu kurallar, kişisel bilgileri hiçbir zaman tanımadığınız kişilerle paylaşmamayı, şüpheli bağlantılara tıklamaktan kaçınmayı ve nazik, sorumlu bir şekilde davranmayı içerir.

Çevrim içi bir hesap oluştururken (örneğin e-posta, okul portalı veya bir uygulama için) dikkatli olmalı ve güvenliğinize önem vermelisiniz. Kayıt sırasında birçok web sitesi kullanıcı adı ve parola ister. Bu aşamada en önemli konu güvenli bir parola seçmektir. Parola yeterince uzun olmalı (en az 12 karakter) ve küçük-büyük harfler, sayılar ve özel karakterlerden (örneğin @, #, \$, %) oluşan bir kombinasyon içermelidir. Doğum tarihi veya evcil hayvanın adı gibi kolayca öğrenilebilecek bilgilerin kullanılması önerilmez. Ayrıca parolanın başka kişilerle paylaşılmaması önemlidir. Parolanın düzenli olarak değiştirilmesi ve farklı hesaplar için farklı parolalar kullanılması önerilir. Bu şekilde verilerin kötüye kullanılma riski azaltılır.

Güvenli iletişim, internetin sorumlu kullanımının ve güvenli bir çevrim içi ortamda öğrenmenin temelidir.





SONUÇ

İnternet hizmetleri, modern dijital toplumun temelini oluşturur ve bilgiye, iletişime ve çeşitli hizmetlere hızlı ve kolay erişim sağlar. Bu hizmetler, çalışma, öğrenme, eğlence ve sosyal etkileşim biçimlerini geliştirerek günlük yaşamımızı dönüştürmektedir. Teknolojilerin sürekli gelişmesiyle birlikte internet hizmetleri, modern yaşamda giderek daha fazla bütünleşmiş ve vazgeçilmez hâle gelmekte, küresel bağlantı ve geleceği şekillendiren yenilikler sağlamaktadır.

ÖRNEK 1



Okul Amaçlı E-Posta Kullanımı

Bir öğrenci, öğretmeninden ev ödevini e-posta yoluyla alır. Mesajı açar, ekli belgeyi indirir, ödevi çözer ve yeni bir ek dosya ile birlikte geri gönderir. Bu süreç, fiziksel bir buluşmaya gerek kalmadan hızlı ve verimli bir şekilde gerçekleşir.

ÖRNEK 2



E-Ticaret Yoluyla Çevrim İçi Alışveriş

Bir kullanıcı internet üzerinden kulaklık satın almak istemektedir. Mağazanın web sitesini ziyaret eder, ürünü seçer, teslimat bilgilerini girer ve ödemeyi internet bankacılığı aracılığıyla gerçekleştirir. Kısa süre sonra sipariş onayını ve kargonun takip numarasını alır.

ÖRNEK 3



Bulut Hizmetinin Kullanımı

Bir öğretmen, tüm ders materyallerini Google Drive üzerinde saklamaktadır. Öğrencilerle ortak bir klasör oluşturur. Öğrenciler bu klasördeki belgeleri görüntüleyebilir, yorum yapabilir ve projeleri herhangi bir cihazdan gerçek zamanlı olarak düzenleyebilirler.



BİLGİ KONTROL SORULARI

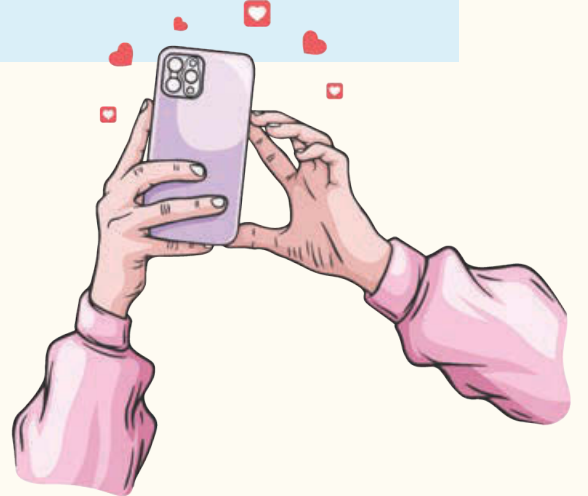


1. İnternet hizmetleri nedir ve temel işlevleri nelerdir?
2. Günlük yaşamında en sık kullandığın internet hizmetleri hangileridir?
3. İnternet hizmetleri insanlar arasındaki iletişimi nasıl etkiler?
4. İnternet hizmetleri öğrenmeyi ve eğitimi hangi şekillerde kolaylaştırır?
5. İnternet hizmetlerini kullanmanın en büyük avantajları nelerdir?
6. İnternet hizmetleri hangi güvenlik risklerini taşıyabilir?
7. Teknoloji, internet hizmetlerinin gelişimini nasıl etkilemektedir?
8. Günümüz toplumunda internet hizmetlerine erişimin önemli olmasının nedeni nedir?
9. Hangi internet hizmetlerine örnek verebilirsin?
10. İnternet hizmetleri küresel bağlantıya nasıl katkı sağlamaktadır?



DAHA FAZLA BİLGİ EDİNMEK İSTEYENLER İÇİN

YouTube ve Netflix gibi akış (streaming) sistemlerinin nasıl çalıştığını araştır. Verilerin depolanmasından video içeriklerinin cihazına aktarılmasına kadar olan süreci incele. Ayrıca, Moodle, Google Classroom veya Microsoft Teams gibi internet hizmetlerinin eğitimde nasıl kullanıldığını ve uzaktan eğitim açısından sağladıkları avantajları değerlendir.



İnternet sürekli gelişmektedir ve her aşaması, web teknolojilerinin yeni bir neslini ve kullanım biçimini temsil etmektedir. İnternetin evrimi ve yeniliklerinden söz edilirken en yaygın olarak Web 1.0, Web 2.0, Web 3.0, Web 4.0 ve Web 5.0 kavramları kullanılmaktadır. Bu terimler, internetin geçirdiği ve geçirmeye devam ettiği değişim sürecini tanımlar. Bu aşamalar kesin sınırlarla birbirinden ayrılmamış olup, birbirini takip ederek gelişmektedir.

Web 1.0, webin ilk neslidir ve 1990'lı yıllarda ile 2000'li yılların başında hâkim olmuştur. Bu dönemde web sayfaları statiktir, yani içerik sabittir ve kullanıcılar tarafından değiştirilemez. Bu aşamada kullanıcılarla çok az ya da neredeyse hiç etkileşim yoktur ve kullanıcılar pasif gözlemcilerdir. Web sayfası tek yönlüdür, içerik oluşturunca kullanıcıya doğru bilgi akışı vardır. Bu döneme örnek olarak, yorum yapma veya düzenleme imkânı sunmayan metin tabanlı Britannica.com ansiklopedisi ile yalnızca bağlantılar ve bilgilerden oluşan, etkileşim özelliği bulunmayan ilk Yahoo! sürümleri verilebilir.

Web 2.0 ile birlikte internet, kullanıcıdan web ortamına ve web ortamından kullanıcıya doğru çift yönlü iletişime geçmiştir. Bu dönemde internet dinamik, etkileşimli ve iş birliğine açık hâle gelmiştir. Kullanıcılar artık yalnızca okuyucu değil, aynı zamanda içerik üreticisidir. Bu döneme örnek olarak, sosyal ağlar (herkesin paylaşım yapabildiği, yorum yazabildiği, içerik paylaşabildiği ve iletişim kurabildiği platformlar), Wikipedia (herkes tarafından düzenlenebilen ansiklopedi), Blogspot ve WordPress (insanların görüşlerini, hikâyelerini ve önerilerini paylaştıkları blog platformları) verilebilir. Bu dönem büyük ilerlemeler sağlamış, ancak veri kötüye kullanımı ve yanlış bilgilendirme gibi yeni sorunları da beraberinde getirmiştir.

Web 3.0, iletişim kurma, bilgi paylaşma ve dijital hizmetleri kullanma biçiminde gerçekleşen bir sonraki büyük değişimi temsil etmektedir. Bu yeni dönem, önceki nesillerden farklı olarak merkeziyetsiz yapı, daha yüksek düzeyde gizlilik ve kullanıcı özerkliği sunmaktadır. Veriler, merkezi sunucular yerine blok zinciri (blockchain) teknolojileri aracılığıyla dağıtılmış şekilde saklanmaktadır. Basit bir ifadeyle bu sistem, herkesin verilerin bir kopyasına sahip olduğu ve hiç kimsenin diğerlerinin farkına varmadan verileri değiştiremediği bir yapıdır. Bu, verilerin tek bir merkezi sunucuda (örneğin bir banka veya şirket sunucusunda) tutulması yerine, dünyanın farklı yerlerindeki birçok bilgisayara dağıtılması anlamına gelir.

Bu dönem aynı zamanda yapay zekâyı da kapsamaktadır ve webi daha akıllı hâle getirmeyi amaçlamaktadır. Bu kapsamda Siri, Alexa ve Google Assistant gibi sesli asistanlar yer almaktadır. Bu sistemler, verileri ve işlevleri büyük şirketler (Apple, Amazon ve Google) tarafından yönetildiği için merkezi yapıda olsalar da, kullanıcı isteklerini daha iyi anlayabilmek amacıyla yapay zekâ ve makine öğrenmesini kullanmaktadır. Web 3.0'ın geleceği, büyük şirketlere bağlı olmayan, daha demokratik yapay zekâ asistanlarının gelişmesine olanak sağlayabilir. Bu döneme örnek olarak, kullanıcıları takip etmeyen ve onları ödüllendiren Brave Browser gösterilebilir. Web 3.0, yalnızca verilerin değil, anlam taşıyan verilerin bulunduğu anlamsal web (Semantic Web) olarak tanımlanmaktadır.

Web 4.0, internetin yalnızca akıllı değil, aynı zamanda öngörücü ve otonom olduğu bir sonraki evrim aşamasını temsil eder. Bu nesil, Nesnelerin İnterneti (IoT – Internet of Things) üzerine kuruludur. Bu yapıda cihazlar birbirleriyle iletişim kurar ve çoğu zaman insan müdahalesi olmadan kendi başlarına karar verir. Web 4.0, “düşünen” interneti (verileri analiz eden ve daha kullanıcı sormadan en uygun bilgileri öneren), doğal iletişimi (sesli asistanlar ve sanal yardımcılarla gerçek bir insanla konuşur gibi iletişim kurmayı) ve birbirleriyle iletişim kuran cihazları (akıllı buzdolapları, otomobiller, saatler vb.) kapsar. Bu cihazlar günlük yaşamı kolaylaştırmak amacıyla birlikte çalışır. Örneğin, akıllı saatin dışarı çıkmadan önce otomobile motorunu ısıtması gerektiğini bildirebilir. Bu döneme örnek olarak, aydınlatmayı, sıcaklığı ve güvenliği otomatik olarak kontrol eden akıllı ev sistemleri ile yapay zekâ yardımıyla öğrenebilen ve kendi kendine sürüş yapabilen Tesla otomobilleri gösterilebilir.

İnternetin önceki aşamalarında kullanıcılar ağırlıklı olarak bilgi aramış (Web 1.0), iletişim kurmuş ve içerik üretmiş (Web 2.0), daha akıllı sistemlere ve merkeziyetsiz platformlara sahip olmuşlardır (Web 3.0 ve Web 4.0). Ancak Web 5.0, bunun da ötesine geçerek insan duygularını anlayan ve dijital kimliğimiz üzerinde tam kontrol sağlayan bir yapı sunmaktadır. Henüz gelişiminin erken aşamalarında olmasına rağmen Web 5.0, yalnızca “düşünen” değil, aynı zamanda “hisseden” bir internet vizyonunu temsil etmektedir. Bu aşamada web yalnızca kullanışlı değil, aynı zamanda empati kurabilen bir yapıya sahip olacaktır. Bu döneme örnek olarak, kullanıcının üzgün olduğunu algılayan ve ona öneriler, yardım hatta moralini yükseltecek uygun müzikler sunan sanal bir asistan verilebilir.



SONUÇ

İnternet hizmetleri, modern dijital toplumun temelini oluşturur ve bilgiye, iletişime ve çeşitli hizmetlere hızlı ve kolay erişim sağlar. Bu hizmetler, çalışma, öğrenme, eğlence ve sosyal etkileşim biçimlerini geliştirerek günlük yaşamımızı dönüştürmektedir. Teknolojilerin sürekli gelişmesiyle birlikte internet hizmetleri, modern yaşamda giderek daha fazla bütünleşmiş ve vazgeçilmez hâle gelmekte, küresel bağlantı ve geleceği şekillendiren yenilikler sağlamaktadır.

ÖRNEK 1



Web 1.0 Dönemine Ait Statik Web Sayfası

1999 yılına ait bir okul web sayfası, yalnızca ders programı bilgilerini göstermekte olup etkileşim veya yorum yapma olanağı sunmamaktadır. İçerik statiktir ve nadiren güncellenmektedir.

ÖRNEK 2



Web 2.0 Döneminde Sosyal Ağ

Öğrenciler, fotoğraf paylaşmak ve iletişim kurmak için Instagram kullanmaktadır. İçerik oluşturmakta, yorum yapmakta, diğer kullanıcıları takip etmekte ve topluluktan geri bildirim almaktadırlar.

ÖRNEK 3



Web 4.0 Döneminde Sesli Asistan

Bir öğrenci, alarm kurmak, bir tanım aramak veya bir toplantı planlamak için Google Assistant kullanır. Asistan kullanıcının sesini anlar ve kullanıcının alışkanlıklarına dayanarak kişiselleştirilmiş bir yanıt verir.

BİLGİ KONTROLÜ SORULARI VE ETKİNLİKLER



1. Web teknolojilerinin gelişim aşamalarının her birinin temel özellikleri nelerdir?
2. Web teknolojilerinin gelişim aşamalarının her biri için örnekler veriniz!
3. "Kullanıcı tarafından oluşturulan içerik" nedir ve hangi web neslinde ortaya çıkmıştır?
4. "Anlamsal web" ne anlama gelir ve hangi nesil ile ilişkilendirilir?
5. Günlük yaşamda Web 4.0'ın bir parçası olan bir cihaz örneği verebilir misiniz?
6. Web 2.0, Web 3.0 ve sonraki nesiller arasında kesin bir sınır var mıdır? Neden?
7. Kullanıcının rolü Web 1.0'dan Web 5.0'a kadar nasıl değişmektedir?
8. Sizce web nesillerinden hangisi yaşam biçimini en fazla değiştirmiştir? Nedenini açıklayınız!



DAHA FAZLA BİLGİ EDİNMEK İSTEYENLER İÇİN

Web 3.0'da blok zinciri (blockchain) teknolojisinin şeffaflık ve güvenliği sağlamak için nasıl kullanıldığını araştır. Ayrıca Web 4.0 ve Web 5.0'da yapay zekâ kullanımının etik yönlerini, dijital iyi oluş, teknolojilere erişim ve ruh sağlığının korunması gibi konuları incele.



İnternet, devlet kurumları, üniversiteler, haber kuruluşları, bloglar, forum tartışmaları ve sosyal ağlar gibi farklı kaynaklardan gelen çok büyük bir bilgi kaynağıdır. Bu kaynakların bazıları bilimsel ve doğrulanmışken, bazıları resmi değildir ve doğru olmayabilir. Peki hangi kaynakların güvenilir ve inandırıcı olduğunu, yani okuduğun bilgilerin doğru ve güvenli olup olmadığını nasıl anlayabilirsin?

Bir web sitesinin güvenilir olup olmadığını değerlendirmek için kendine AŞAĞIDAKİ soruları sor:

- Bilgiyi kim yayınlamış? Bilgi, okul, üniversite, devlet kurumu veya tanınmış bir medya kuruluşu gibi bilinen ve güvenilir bir kaynaktan mı geliyor?
- Bir yazar belirtilmiş mi? Yazarın adı ve yaptığı iş (gazeteci, bilim insanı veya uzman olup olmadığı) belirtilmiş mi?
- Metin ne zaman yayımlanmış? Yakın zamanda mı yazılmış, yoksa çok eski mi? Bilgiler hâlâ geçerli ve doğru mu?
- Metindeki bilgiler hangi kaynaklardan alınmış? Yazılanları doğrulayan kaynaklardan veya diğer web sayfalarına bağlantılardan söz edilmiş mi?
- Metin nasıl yazılmış? Metin taraflı bir anlatıma mı sahip, güçlü duygular veya önyargılar kullanarak seni bir şeye inandırmaya mı çalışıyor, yoksa nesnel ve gerçeklere dayalı mı?
- Bir web sitesinin güvenli ve güvenilir olduğunu belirledikten sonraki adım, okuduğun bilginin kendisinin doğru olup olmadığını kontrol etmektir. Bunu yapmak için:
- Bilgiyi diğer kaynaklarla karşılaştırmalısın – başka web sitelerinin de aynı bilgiyi doğrulayıp doğrulamadığını kontrol etmelisin.
- Gerçekler ve kanıtlar aramalısın – bilgiyi destekleyen açıklamalar, sayılar veya bilimsel veriler bulunup bulunmadığına bakmalısın.
- Dolandırıcılıklara karşı çok dikkatli olmalısın – sahte haberler, yanlış bilgiler ve yalnızca tıklamanı sağlamak amacıyla hazırlanmış başlıklara (sözde “clickbait”) karşı dikkatli davranmalısın.

Başarılı bir internet araması için, aramanın teknik yönlerinin yanı sıra kullanıcılar arasında medya okuryazarlığının teşvik edilmesi çok önemlidir – yani gerçekleri görüşlerden, doğru bilgileri yanlış bilgilerden ve nesnel raporları taraflı yorumlardan ayırt etme becerisi. Kaynakların eleştirel değerlendirilmesi yalnızca teknik parametrelerin değerlendirilmesini değil, aynı zamanda bağlamsal analizi de gerektirir: Metni kim yayımlıyor, hangi amaçla yayımlıyor ve aynı bilginin alternatif versiyonları var mı? Bilgi manipülasyonları, dezenformasyonlar ve “sahte haberler” çoğunlukla sosyal ağlar ve yanlış web siteleri aracılığıyla yayılmaktadır. Bu nedenle doğrulanmış medyanın kullanılması, Snopes, FactCheck.org gibi platformlar veya yerel doğrulama kaynakları aracılığıyla gerçeklerin kontrol edilmesi ve belirli bir olayı analiz ederken birden fazla kaynağa başvurulması önerilir. Öğrenciler, buldukları ilk bilgiyi kabul etmemek, bunun yerine bilgiyi diğer kaynaklarla karşılaştırmak ve mantıksal bir tutarlılık olup olmadığını, saygın yazarlar veya kurumlar tarafından desteklenip desteklenmediğini ve sunulan verilerin doğrulanabilir olup olmadığını değerlendirmek alışkanlığını geliştirmelidir.

Web araması dijital çağda temel bir beceridir, ancak kaynakların uygun şekilde değerlendirilmesi olmadan yalnızca arama yapmak yeterli değildir. İçeriği, yazarını, yapısını ve bağlamını analiz ederken eleştirel düşünmenin uygulanması gereklidir. Gelişmiş arama tekniklerinin uygulanması ve güvenilirliğin değerlendirilmesiyle, öğrenme, araştırma ve bilgi edinme amacıyla kullanılacak daha güvenilir bilgiler elde edilir. Yalnızca bu yaklaşımla, karar verme süreçlerini etkileyebilecek manipülasyonlardan, dezenformasyonlardan ve yanlış sonuçlardan kaçınılabilir.



SONUÇ

İnternet öğrenme için güçlü bir araçtır, ancak doğru ve güvenilir bilgileri nasıl tanıyacağını kendin bilmelisin. Kaynakları dikkatli bir şekilde değerlendirerek ve bilgileri karşılaştırarak, kendini yalanlardan ve dolandırıcılıklardan koruyabilir ve doğru sonuçlara ulaşabilirsin. Web'i yalnızca bir arama aracı olarak değil, eleştirel bir düşünür olarak kullan.



ÖRNEK 1

Akademik Arama

Bir öğrenci “İklim değişiklikleri” konusu hakkında bir sunum hazırlamaktadır. İlk sonuçları kullanmak yerine, tırnak işaretleri kullanarak (“Makedonya’da iklim değişiklikleri”) arama yapar ve resmî kaynaklara ulaşmak için site:.gov.mk ekler.

ÖRNEK 2



Güvenilirlik Kontrolü

Bir kullanıcı sosyal ağda sansasyonel görünen bir haberle karşılaşır. Kullanıcı, içeriği bağımsız medya kaynaklarında kontrol eder ve haberin yanlış olduğunu ortaya çıkarmak için bir doğrulama platformu kullanır.

ÖRNEK 3



Sonuçların Filtrelenmesi

Bir öğretmen dijital okuryazarlık ile ilgili PDF materyalleri aramaktadır. Doğru ve ilgili belgeleri elde etmek için Google'da şu operatörleri kullanır: filetype:pdf intitle: "dijital okuryazarlık" site:.edu.mk.

BİLGİ KONTROLÜ SORULARI VE ETKİNLİKLER



1. İnternetin bilgi kaynağı olarak kullanılması ne anlama gelir?
2. Güvenilir (sağlam) kaynak nedir? Örnek ver!
3. Bir web sitesinin güvenilir olup olmadığını nasıl anlayabilirsin?
4. Bilgiyi kimin yazdığını bilmek neden önemlidir?
5. Bilginin tarafsız olması ne anlama gelir? Taraflı bir metni nasıl fark edebilirsin?
6. "Clickbait" nedir ve neden dikkatli olmalısın?
7. Web bilgilerini kontrol ederken kendine sorman gereken en önemli üç soru nelerdir?
8. İki web sitesinde farklı bilgiler bulursan ne yapmalısın?



DAHA FAZLA BİLGİ EDİNMEK İSTEYENLER İÇİN

Arama motorlarının teknik açıdan nasıl çalıştığını araştır – web crawler'ın ne olduğunu, web sayfalarının nasıl indekslendiğini ve sıralama için hangi algoritmaların kullanıldığını incele. Ayrıca "bilgi balonu" (filter bubble) kavramını ve internette arama yaptığımızda elde ettiğimiz sonuçları nasıl etkilediğini değerlendir.



İnterneti her kullandığında – Google’da bir şey aradığında, Instagram’da bir fotoğraf paylaştığında, YouTube’da video izlediğinde veya yalnızca bir bağlantıya tıkladığında bile – arkada bir iz bırakırsın. Bu iz dijital ayak izi olarak adlandırılır. Dijital ayak izi senin dijital geçmişindir – ve senin hakkında birçok bilgi verebilir. Dijital ayak izi daha ayrıntılı olarak, bir kişinin internette bıraktığı tüm bilgilerin toplamı olarak açıklanabilir – bu bilgiler bilinçli olarak bırakılan bilgiler olabilir (sosyal ağ profilleri, paylaşımlar, fotoğraflar gibi) veya sistemin biz belirli bir şey girmeden otomatik olarak topladığı bilinçsiz bilgiler olabilir (konum verileri, aramalar, çerezler veya uygulama etkinlikleri gibi). Bu, dijital ortamda kendini yalnızca aktif olarak gösterebileceğin değil, aynı zamanda pasif olarak da arkada veriler bırakabileceğin anlamına gelir – örneğin internetteki alışkanlıklarını takip eden çerezler (cookies) gibi.

Dijital ayak izinin olumlu ve olumsuz yönleri vardır.

Dijital ayak izinin olumlu yönleri:

- Çevrim içi alışveriş ve bankacılık gibi hizmetlere erişime yardımcı olur.
- Senin hakkında olumlu bir imaj oluşturulmasına yardımcı olabilir – olumlu bir dijital iz, birçok üniversite, kuruluş ve şirket internette bir kişi hakkında bulunan bilgileri kontrol ettiği için burs, staj veya iş elde etmene yardımcı olabilir.
- İnternet ilgi alanlarını hatırlar ve sana uygun içerikler sunar.
- Benzer ilgi alanlarına sahip kişilerle iletişim kurmayı ve ağ oluşturmayı sağlar, bu kişilerle iş birliği yapabilir ve onlardan öğrenebilirsin.
- Dijital ayak izinin olumsuz yönleri:
- Verilerin senin iznin olmadan kötüye kullanılabilir veya takip edilebilir.
- İnternette paylaştığın her şey sonsuza kadar erişilebilir durumda kalabilir. Fotoğraflar, yorumlar, videolar veya paylaşımlar bir zamanlar eğlenceli görünmüş olsa bile, daha sonra seni olumsuz bir şekilde gösterebilir.
- Düşünmeden yapılan paylaşımlar gelecekte senin hakkındaki algıyı etkileyebilir. Uygunsuz içerikler, hakaret içeren ifadeler veya başkalarına karşı saygısız davranışlar, itibarını olumsuz etkileyebilir.

İnternetin sorumlu bir şekilde kullanılması ve paylaşacağın şeyleri dikkatli seçmen, eğitim hayatın ve gelecekteki kariyerin için yararlı olacak olumlu ve güçlü bir dijital ayak izi oluşturmaya yardımcı olabilir. Bu nedenle:

- ✓ Bir şey paylaşmadan önce kendine şunu sor: "Ailem, öğretmenlerim veya gelecekteki işverenim bunu görse ister miydim?" "Bu bana gelecekte zarar verebilir mi?"
- ✓ Uygulamalarının ve profillerinin gizlilik ayarlarını kontrol et. Tüm dünyanın senin hakkında her şeyi bilmesine gerek yoktur.
- ✓ Kişisel bilgilerini (adres, telefon, şifreler, kart bilgileri gibi) – ne kendi bilgilerini ne de başkalarının bilgilerini – paylaşma.
- ✓ Kimi arkadaş olarak kabul ettiğine dikkat et. İnternette arkadaş canlısı görünen herkesin iyi niyetli olmadığı unutulmamalıdır.
- ✓ İyi bir dijital kimlik oluştur. İlgi alanlarını, projelerini ve fikirlerini gösteren içerikler paylaş.

Dijital ayak izinin bilinçli ve bilinçsiz şekilde oluşturulmasının yanı sıra, bu bilgilerin nasıl kullanıldığını anlamak da önemlidir. Şirketler verileri hedefli pazarlama için kullanır – bu, kullanıcıya ilgi alanlarına, önceki aramalarına veya konumuna göre reklamlar gösterilmesi anlamına gelir. Örneğin, bir kullanıcı ayakkabı araması yaparsa, ertesi gün birçok web sitesinde ayakkabı reklamları alabilir. Bu durum, etkinlikleri analiz eden ve ilgili içerikleri sunan algoritmaların sonucudur. Ancak kişiselleştirme yararlı olabilse de gözetim ve dijital özgürlük konusunda sorular ortaya çıkarır. Birçok durumda kullanıcılar nasıl, kim tarafından ve ne ölçüde takip edildikleri konusunda bilgilendirilmez. Genellikle bu süreçler nadiren okunan uzun gizlilik politikalarında belirtilir. Zamanla, kullanıcının kontrolü dışında üçüncü taraflara satılabilecek veya siyasi hedefleme, fikir manipülasyonu ya da ticari faaliyetler için kullanılacak karmaşık bir dijital profil oluşturulur. Bu riskler, eğitim, farkındalık ve takip engelleyiciler, VPN, şifrelenmiş iletişim ve uygulama izinlerinin sık sık kontrol edilmesi gibi gizlilik koruma araçlarının kullanılmasının gerekliliğini ortaya koymaktadır.

Kişisel gizliliğin bir diğer önemli unsuru, verilerin saklanma ve işlenme biçimidir. Platformlar, kişisel belgeler, fotoğraflar, yazışmalar, konumlar, kullanılan cihazlar ve davranış biçimleri dahil olmak üzere çok büyük miktarda bilgi toplar. Bu veriler, farklı ülkelerde bulunabilen sunucularda saklanır ve bu durumlarda yasal düzenlemeler farklı ve yetersiz olabilir. Kullanıcılar çoğu zaman hangi şirketlerin verilerine eriştiğini, hangi koşullar altında ve ne kadar süreyle kullandığını bilmez. Profiller silinse bile veriler yedek kopyalarda kalabilir veya başka kuruluşlara satılabilir. Bu konular Avrupa Birliği'ndeki Genel Veri Koruma Tüzüğü (GDPR) gibi yasalarla düzenlenmektedir, ancak etkili bir koruma sağlamak için küresel uyumlaştırmaya ihtiyaç vardır. Bireysel düzeyde kullanıcılar bilgi paylaşırken dikkatli olmalı, güçlü şifreler kullanmalı, tüm izinleri düşünmeden kabul etmemeli ve dijital alışkanlıklarını ve ayarlarını düzenli olarak gözden geçirmelidir.

Dijital ayak izi, internetteki her kullanıcının alışkanlıkları, ilgi alanları ve kimliği hakkında güçlü bir göstergedir. Her ne kadar bazen kişiselleştirme ve hizmetlerin iyileştirilmesi için yararlı olsa da, kötüye kullanıldığında ciddi riskler taşır. Kişisel gizliliğin korunmasından sorumluluk yalnızca kurumlara ait değildir, aynı zamanda çevrim içi ortamda bilinçli ve dikkatli davranması gereken kullanıcıların da sorumluluğundadır. Güvenlik araçlarının, eleştirel düşünmenin ve dijital okuryazarlığın kullanılmasıyla her birey kendi dijital ayak izini kontrol edebilir ve dijital dünyada gizliliğini koruyabilir.



SONUÇ

Dijital ayak izi güçlü bir araçtır, ancak risklerden kaçınmak için onu nasıl yöneteceğini bilmelisin. İnterneti doğru kullanarak verilerini koruyabilir ve olumlu bir çevrim içi itibar oluşturabilirsin.

ÖRNEK 1



Bilinçsiz Ayak İzi Oluşturma

Bir öğrenci birkaç uygulamaya aynı şifreyle giriş yapar ve konumuna ve kişilerine erişim izni verir. Veriler depolanır ve açık izni olmadan pazarlama ağlarına aktarılır.

ÖRNEK 2



Selfie ve Kişisel Verilerin Paylaşılması

Bir kullanıcı sosyal ağda konum bilgisi ve etiketlenmiş arkadaşlarıyla birlikte gerçek zamanlı olarak bir selfie fotoğrafı paylaşır. Bu tür bir paylaşım, analiz edilebilecek ve kötüye kullanılacak bir dijital iz bırakır.

ÖRNEK 3



VPN Kullanımı

Bir öğretmen, seyahat ederken kişisel verilerini herkese açık Wi-Fi ağlarından korumak için VPN kullanır. Bu sayede iletişim şifrelenir ve pasif dijital ayak izinin oluşturulması sınırlandırılır.

BİLGİ KONTROLÜ SORULARI VE ETKİNLİKLER



1. „Dijital ayak izi“ kavramı ne anlama gelir?
2. Hangi bilgi türleri senin dijital ayak izinin bir parçasıdır?
3. İnternette paylaştıklarına karşı dikkatli olman neden önemlidir?
4. Nasıl olumlu bir dijital ayak izi oluşturabilirsin?
5. İnterneti sorumlu kullanma ve olumlu bir dijital ayak izi oluşturma ile ilgili kendi kurallar listesini oluştur!
6. Silinen bir fotoğraf tamamen silinmiş olur mu? İnternette verilerin nerede saklandığını açıkla!



DAHA FAZLA BİLGİ EDİNMEK İSTEYENLER İÇİN

Algoritmaların internet üzerindeki davranışları nasıl takip ettiğini araştır – örneğin çerezler (cookies), izleme pikselleri (tracking pixels), parmak izi oluşturma (fingerprinting) ve veri madenciliği (data mining). Ayrıca şirketlerin mobil uygulamalar aracılığıyla nasıl veri topladığını ve “verilerin para birimi olarak kullanılması” kavramının ne anlama geldiğini öğren. Gizliliği koruma araçlarını, DuckDuckGo, Tor, Signal ve diğer güvenli iletişim platformlarını incele.



2.6

İnternetin Tehlikeleri

Günümüz dünyasında internet, günlük hayatımızın vazgeçilmez bir parçasıdır. İnternet, bilgilere hızlı erişim, sınırlar olmadan iletişim ve öğrenme ile eğlence için sonsuz olanaklar sağlar. İnternet sayesinde dünyanın her yerinden insanlarla bağlantı kurabilir, yeni konuları araştırabilir ve bilgini geliştirebilirsin. Tüm faydalarının yanında internet, kişisel yaşamın ve toplum üzerinde ciddi sonuçlara yol açabilecek birçok tehlikeyi de barındırır.

En yaygın tehlikelerden biri dijital şiddettir. Dijital şiddet, bir kişiyi küçük düşürme, incitme veya zarar verme amacıyla dijital teknolojilerin, yani cep telefonlarının ve internetin kullanılmasıdır. Dijital şiddet birçok şekilde ortaya çıkabilir. En yaygın dijital şiddet türlerinden bazıları şantaj, tehdit, taciz, internet üzerinden cinsel istismar, siber zorbalık, sahte profiller oluşturma ve kullanma ile başkalarına ait fotoğrafların kötüye kullanılmasıdır.

Bir diğer risk ise belirli insan gruplarına karşı ırk, din, cinsiyet veya diğer özellikleri nedeniyle ayrımcılığı ve hoşgörüsüzlüğü teşvik eden nefret içerikleridir. Bu tür içerikler insanlar arasındaki birlikte yaşamayı ve anlayışı olumsuz etkiler ve ciddi çatışmalara yol açabilir.

İnternette uyuşturucu kullanımı, alkol kötüye kullanımı, yeme bozuklukları (anoreksiya, bulimiya) veya hatta kendine zarar verme ve intihar gibi zararlı ve tehlikeli davranışları açıkça ya da gizlice teşvik eden içeriklerle karşılaşılabilir. Bu içerikler videolar, bloglar, sosyal ağlardaki gönderiler veya müzik sözleri şeklinde olabilir ve çoğu zaman “acıyla başa çıkma yolu” ya da “normal bir şey” olarak sunulur. Aktardıkları mesajlar çekici, isyankâr veya hatta kişisel kimliğin bir ifadesi gibi görünebilir, ancak gerçekte ruhsal ve fiziksel sağlık açısından ciddi riskler taşır. Bu içerikler gerçeklik hakkında çarpık bir algı oluşturabilir, kendine zarar verici davranışları teşvik edebilir veya gençlerde değer ve özsaygı duygusunu azaltabilir.

Dezenformasyonlar ve sahte haberler de çok tehlikelidir, çünkü bunlar insanları şaşırtmak veya onlara zarar vermek amacıyla kasıtlı olarak yayılır. Dezenformasyonu, insanları kandırmak, kamuoyunu yönlendirmek veya birine zarar vermek amacıyla yayılan kasıtlı olarak yanlış veya hatalı bilgi olarak tanımlayabiliriz.



Resim 4 Veri hırsızlığı



Bunun aksine, sahte haber, uydurulmuş veya yanlış şekilde sunulmuş, gerçekmiş gibi görünen haberdır. Zarar verme amacıyla veya zarar verme amacı olmadan hazırlanmış olabilir. Bu içerikler çoğu zaman gerçek haberler gibi görünür, dikkat çekici bir başlığa, fotoğraflara ve ikna edici görünen biçimlendirilmiş metinlere sahiptir. Ancak doğrulanmadıkları takdirde ciddi sonuçlara yol açabilirler. Örneğin, hastalıklar veya ilaçlarla ilgili yanlış bilgiler, özellikle insanların sağlık çalışanlarının önerilerini takip etmeyi bırakması durumunda sağlık güvenliğini tehlikeye atabilir. Doğal afetler veya salgın hastalıklar gibi kriz dönemlerinde dezenformasyonlar panik, güvensizlik ve karmaşa oluşturabilir.

Son olarak, bilgisayarına zarar verebilecek veya kişisel verilerini çalabilecek zararlı bilgisayar programlarını, yani virüsleri de unutmamak gerekir. Bunlar genellikle şüpheli web sitelerinde, e-postalardaki bağlantılarda veya doğrulanmamış uygulamalarda gizlenir. Bazıları verilerini kilitleyebilir, bazıları ise internetteki hareketlerini gizlice takip ederek şifreler veya banka bilgileri gibi hassas bilgileri toplayabilir.

İnternet ve dijital cihazlara bağımlılık, özellikle genç kullanıcılar arasında giderek daha sık görülen bir diğer tehlikedir. Ekranlara, oyunlara, sosyal ağlara veya video içerik platformlarına uzun süre maruz kalmak, uyku bozukluklarına, dikkat eksikliğine, sosyal izolasyona ve fiziksel aktivitenin azalmasına yol açabilir. Oyun bağımlılığı, Dünya Sağlık Örgütü tarafından ruhsal bir bozukluk olarak kabul edilen özel bir kategoridir. Kişinin sağlığını, okul sorumluluklarını veya sosyal ilişkilerini olumsuz etkilese bile güçlü bir oyun oynama isteği hissetmesiyle karakterizedir. Sosyal ağların aşırı kullanımı da benzer etkilere sahip olabilir, başkalarıyla kendini karşılaştırmaya, kaygıya veya kendinden memnuniyetsizliğe yol açabilir. Dijital dengeyi teşvik etmek, cihazsız zamanlar oluşturmak ve ekran dışındaki etkinliklere yönelmek, bağımlılığı önlemenin en önemli stratejilerinden bazılarıdır.

Güvenliğini, ruh sağlığını, gizliliğini veya davranışlarını olumsuz etkileyebilecek “riskli programlar” ve “riskli içerikler” şunları içerir:

- Virüs veya casus yazılım içerebilecek doğrulanmamış indirilebilir uygulamalar,
- Kabul edilemez şiddet sahneleri veya nefret söylemi içeren oyunlar,
- İnanç, ırk, cinsiyet, cinsel yönelim vb. nedenlerle insanlara hakaret eden videolar, metinler veya yorumlar,
- Uyuşturucu kullanımını moda gibi veya kendine zarar vermeyi duygularla başa çıkma yöntemi gibi gösteren platformlar,
- Ücretsiz olmayan belirli programların “kilidini açan” ücretsiz “hacker” araçları,
- Eğlenceli görünebilen ancak tehlikeli ve yıkıcı olan çevrimiçi meydan okumalar ve diğerleri.

Bu nedenle, interneti kullanırken şunları düşün:

- Senden kişisel verilerini paylaşman isteniyor mu?
- İçerik seni tehlikeli veya yasa dışı bir şey yapmaya yönlendiriyor mu?
- Ruh halini veya kendine ve başkalarına bakış açını olumsuz etkiliyor mu?
- Kaynak şüpheli veya anonim mi?
- İnternette olumsuz bir deneyim yaşarsan nasıl davranırsın?

İnterneti her zaman içeriklere dikkat ederek, eleştirel düşünerek ve kendine ve başkalarına karşı sorumluluk duygusuyla kullan.



SONUÇ

İnternet öğrenmek, sosyalleşmek ve eğlenmek için harika bir yerdir – ancak aynı zamanda ciddi risklerin gizlendiği bir alan da olabilir. Bu nedenle dikkatli ve sorumlu bir kullanıcı olman önemlidir.

ÖRNEK 1



E-posta Üzerinden Kimlik Avı (Phishing) Mesajı

Bir öğrenci, ödül kazandığını belirten ve onay için bir bağlantıya tıklaması gerektiğini söyleyen bir e-posta alır. Bağlantıya tıkladıktan sonra kişisel bilgilerini girer ve bu bilgiler daha sonra kötüye kullanılır.

ÖRNEK 2



Oyun Bağımlılığı

Bir lise öğrencisi günde 6 saatini çevrim içi oyun oynayarak geçirir, bu nedenle okul sorumluluklarını ihmal eder, geç uyur ve arkadaşlarıyla sosyalleşmekten kaçınır.

ÖRNEK 3



Sahte Haber Yayılması

Bir kullanıcı, daha sonra tamamen yanlış bir bilgi olduğu kanıtlanan sansasyonel bir “keşif” yazısını paylaşır. Bu durum, bazı arkadaşları arasında paniğe neden olur.

BİLGİ KONTROLÜ SORULARI VE ETKİNLİKLER



1. Kötü amaçlı programlara ve internet içeriklerine üç örnek veriniz!
2. Dezenformasyon insanlar ve toplum üzerinde nasıl etkili olabilir?
3. Dezenformasyon ile sahte haber arasındaki farkı açıklayınız!
4. İnternette nefret söylemi içeren şiddet içerikleri ergenleri nasıl etkileyebilir?
5. Uyuşturucu kullanımı veya kendine zarar verme ile ilgili tanıtım içeriklerini fark etmek neden önemlidir?
6. Seni rahatsız eden veya endişelendiren bir içerikle karşılaşırsan ne yapabilirsin?
7. İnterneti kullanırken eleştirel düşünmek neden önemlidir?



DAHA FAZLA BİLGİ EDİNMEK İSTEYENLER İÇİN

Araştırınız: “dark patterns” (karanlık tasarımlar) nedir? Bunlar, web sitelerinin kullanıcıları normalde gönüllü olarak yapmayacakları bir işlemi yapmaya yönlendirmek için kullandıkları tekniklerdir (örneğin: tüm çerezleri kabul ettirmek, ürün satın almaya yönlendirmek, bültene kaydolmaya zorlamak). Ayrıca dezenformasyonların toplumda protestolar, panik veya finansal dolandırıcılıklar gibi gerçek sonuçlara yol açtığı durumları analiz ediniz.



Resim 5 Bir web sitesinin kullanım koşulları

İnternet güvenliği, modern teknolojileri anlama, kişisel verileri koruma ve internette akıllı ve doğru davranış sergileme gibi birçok unsuru kapsar. Çevrim içi ortamda güvenli olmak, interneti senin için yararlı olacak şekilde kullanmak ve aynı zamanda çeşitli risk ve tehditlere karşı korunmak anlamına gelir.

En önemli konu kişisel verilerin korunmasıdır. İnternette bilgi paylaşımı bilinçli bir şekilde yapılmalı ve paylaşılan verilerin – adresler, fotoğraflar, alışkanlıklar ve görüşler gibi – bağlamından kopararak kullanılabileceği veya kötüye kullanılabileceği anlaşılmalıdır. Her kullanıcının gizlilik hakkı vardır, ancak aynı zamanda bu hakkı başkaları için de koruma sorumluluğu vardır. Diğer kişilere ait kişisel içeriklerin onların izni olmadan kaydedilmesi, paylaşılması veya kullanılması etik değildir. Aynı zamanda, internetteki sahte reklamlardan şüpheli mesajlara kadar dolandırıcı ve manipülatif içeriklerin fark edilmesi ve bu tür içeriklerin paylaşılmaması veya desteklenmemesi önemlidir. Bu şekilde daha güvenli ve güvenilir bir dijital ortamın oluşturulmasına katkıda bulunabilirsin.

AŞAĞIDAKİ öneriler, çevrim içi kimliğini korumana ve risklerden kaçınmana yardımcı olacaktır:

- ✓ Güçlü ve benzersiz parolalar kullan – Her hesabın için farklı ve karmaşık parolalara sahip olmak önemlidir. En az 12 karakter kullan, büyük ve küçük harfleri, sayıları ve özel karakterleri birleştir, kolay tahmin edilebilen kelimelerden veya isim ve doğum tarihi gibi kişisel bilgilerden kaçın.
- ✓ Kimlik avı (phishing) dolandırıcılıklarına ve zararlı bağlantılara dikkat et – Tanımadığın kaynaklardan gelen e-posta veya mesajlardaki şüpheli bağlantılara asla tıklama. Kimlik avı, sahte web siteleri veya mesajlar aracılığıyla kişisel verilerini ele geçirme girişimidir, bu nedenle kaynağın güvenilirliğini her zaman kontrol et.
- ✓ İki faktörlü kimlik doğrulamayı (2FA) etkinleştir – Bu, parolanın yanı sıra ek bir kod (örneğin cep telefonundan alınan bir kod) isteyen ek bir güvenlik katmanıdır. 2FA sayesinde bir kişi parolanı öğrense bile hesabına kolayca erişemez.
- ✓ Cihazlarını koru – Yazılımları ve işletim sistemini düzenli olarak güncelle, çünkü bu güncellemeler çoğunlukla güvenlik açıklarını gideren düzeltmeler içerir. Olası tehditleri tespit edecek ve engelleyecek antivirüs programları kullan.
- ✓ Web sitesinin "https://" ile başlayıp başlamadığını kontrol et, bu, güvenli bir bağlantıyı gösterir. Bankacılık işlemleri veya internet üzerinden alışveriş gibi hassas işlemler için halka açık Wi-Fi ağlarını kullanmaktan kaçın.
- ✓ Uygulamaları, filmleri ve müzikleri güvenilir ve güvenli web sitelerinden ve yaşına uygun kaynaklardan indir.
- ✓ Kullandığın cihazlarda ve platformlarda uygun gizlilik ayarlarını yap.
- ✓ Çoğu platform ve web sitesinde uygunsuz içerikleri bildirme, işaretleme veya kötü niyetli kullanıcıların kötüye kullanımını bildirme seçenekleri bulunur.

- ✓ Bilgi paylaşım seçeneklerini, tanımadığın kişilerin erişimini sınırlandıracak şekilde ayarla. Hangi verileri ve kimlerle paylaştığının farkında ol.
- ✓ Siber zorbalık veya çevrim içi bir saldırgan gibi hoş olmayan bir durumla karşılaşırsan, güvendiğin bir kişiyle veya sorunun çözülmesine yardımcı olacak ebeveynlerinle konuş.

Bilgisayarı virüsler, kötü amaçlı yazılımlar ve bilgisayar korsanı saldırıları gibi tehlikelerden korumana yardımcı olacak çeşitli programlar bulunmaktadır. AŞAĞIDAKİ örnekler farklı işlevler için kullanılır:

- **Antivirüs programları:** Kaspersky, Avast, Bitdefender – virüsleri tespit eder ve kaldırır.
- **Güvenlik duvarı (firewall),** internet trafiğini kontrol eder ve yetkisiz erişimleri engeller.
- **Kötü amaçlı yazılımlara karşı programlar,** Malwarebytes – casus yazılımlara ve tehlikeli uygulamalara karşı koruma sağlar.
- **İçerik filtreleri** – ebeveyn denetimleri ve istenmeyen web sitelerini engelleyen filtrelerdir.
- **İşletim sistemi güncelleme programları** – düzenli güncellemeler ile sistemin güvenliğini artırır.

Bu programlar bilgisayarın güvenli ve kararlı çalışmasına yardımcı olur.

Bilgisayarda güvenli çalışma, fiziksel ve zihinsel sağlığını olumsuz etkilememesi için doğru şekilde kullanılmalıdır. Aşağıda sağlıklı yaşamı destekleyen bazı etkinlikler verilmiştir.

1. **Ekran karşısında çalışmaya ara vererek gözlerini dinlendir,** her 30–40 dakikada bir mola ver. Yürü, esneme hareketleri yap veya gözlerini dinlendir.
2. **Sağlıklı alışkanlıklar oluştur,** sosyal medyada geçirilen zamanı sınırla. Ayrıca uyumadan önce cihazları kullanma (en az bir saat).
3. **Ebeveynler/öğretmenlerle konuş.** Rahatsız edici bir durum veya çevrim içi tacizle karşılaşırsan bir yetişkinle konuş.
4. **Dijital nezaket göster,** çevrim içi iletişimde nazik ol, uygunsuz veya tehlikeli içerikleri bildir.
5. **Ruh sağlığına dikkat et.** Kendini internetteki insanlarla karşılaştırma – birçok içerik gerçekçi değildir. İnterneti yalnızca eğlence için değil, öğrenme, hobiler ve gelişim için kullan.



NOT:

Kuzey Makedonya Cumhuriyeti'nde:

- Kişisel Verilerin Korunması Kanunu'na uygun olarak, her kullanıcının kişisel verilerinin korunması hakkı vardır. Verilerin paylaşılması veya işlenmesi sırasında gizliliğe saygı gösterilmeli, kişisel veriler yalnızca kişinin rızasıyla veya yasal yetkilendirmelere uygun olarak kullanılmalıdır. Ceza Kanunu, siber zorbalık, virüslerin ve kötü amaçlı yazılımların yayılması gibi siber suçlar için yaptırımlar öngörmektedir.

Ceza Kanunu, siber zorbalık, virüslerin ve kötü amaçlı yazılımların yayılması gibi siber suçlar için yaptırımlar öngörmektedir.

Kuzey Makedonya Cumhuriyeti'nde Elektronik Haberleşme Kanunu, ülkede internetin güvenli kullanımını sağlama konusunda önemli bir rol oynamaktadır. İnternetin güvenli, korumalı ve sorumlu bir şekilde kullanılmasını destekleyen bir çerçeve oluşturur.



NOT:

Dünya genelinde Güvenli İnternet Günü, çocuklar ve gençler için dijital teknolojilerin güvenli ve olumlu kullanımını teşvik etmek amacıyla her yıl şubat ayında kutlanmaktadır.



SONUÇ

İnterneti akıllı ve doğru bir şekilde kullanarak, onun tüm avantajlarından yararlanabilir ve aynı zamanda olası risklerden kendini koruyabilirsin. Kullanıcı iyi bilgilendirilmiş, dikkatli ve sorumlu olmalıdır, böylece internet herkes için güvenli bir ortam olarak kalabilir.

ÖRNEK 1



Kişisel Gizliliğe Saygı Gösterme

Öğrenci bir okul etkinliğinde grup fotoğrafı çeker, ancak fotoğrafı sosyal medyada paylaşmadan önce tüm katılımcılardan izin ister ve erişim kısıtlamaları belirler.

ÖRNEK 2



Hakaret İçerikli İçeriklerin Bildirilmesi

Kullanıcı, herkese açık bir grupta ayrımcı bir yorum fark eder ve aynı saldırganlıkla karşılık vermek yerine bunu platforma bildirmeye karar verir.

ÖRNEK 3



Eğitsel Bir Tartışmaya Katılma

Öğrenci çevrim içi bir okul tartışmasına katılır ve karşıt bir görüş ifade eder, ancak bunu nazik ve gerekçeli bir şekilde yaparak olumlu bir diyalogu teşvik eder.

BİLGİ KONTROLÜ SORULARI VE ETKİNLİKLER



1. Bilgisayar kullanırken güvenliği sağlamak için kullanılan üç program veya araç hangileridir?
2. Yasal düzenlemelere göre interneti güvenli kullanma kurallarına uymanın neden önemli olduğunu açıklayınız!
3. Güvenlik kurallarından birini uygulayarak şüpheli bir mesaj veya bağlantı aldığınızda nasıl hareket edeceğinizi açıklayınız!
4. İki durumu analiz ediniz ve kullanıcının güvenliği açısından hangi durumda daha büyük risk olduğunu belirleyiniz:
5. a) Şifreyi bir arkadaşla paylaşmak veya
6. b) Bilinmeyen bir web sitesinden program indirmek. Nedenini açıklayınız!
7. Sosyal ağları günde beş saatten fazla kullanmanın genel iyi oluş hâlini etkileyip etkilemeyeceğini değerlendiriniz! Görüşünüzü gerekçelendiriniz!
8. Sınıfınızdaki öğrencilere önereceğiniz interneti güvenli kullanmaya yönelik beş kural oluşturunuz!

KONU 2: GENEL TEKRARLAMA

1. Bilgisayar ağı ile internet arasındaki fark nedir?
2. Üç internet hizmeti (servisi) belirtiniz!
3. "Dijital ayak izi" kavramı ne anlama gelir?
4. İnternetin bilgi edinmeyi ve bilgi paylaşımını nasıl sağladığını kendi kelimelerinizle açıklayınız!
5. Web'de geçerli ve doğrulanmış kaynakların kullanılması neden önemlidir?
6. Dijital ayak izinin bir olumlu ve bir olumsuz sonucunu açıklayınız!
7. Bilgi kaynağı olarak geçerli olduğunu düşündüğünüz bir web sitesi bulunuz ve neden güvenilir olduğunu gösteren iki neden belirtiniz!
8. Bir okul projesini paylaşmak için elektronik posta veya bulut hizmetini nasıl kullanacağınıza örnek veriniz.
9. İnternetteki belirli bir tehlikeden (örneğin: virüs, sahte profil, kimlik avı mesajı) nasıl korunacağınızı açıklayınız!
10. İki internet hizmeti türünü karşılaştırınız: örneğin elektronik posta ve sosyal ağlar! Benzer ve farklı yönleri nelerdir?
11. Bazı web sitelerinin neden güvenli olmadığını analiz ediniz (üç belirtiyi belirleyiniz)!
12. Dijital ayak izinin bir kişinin gelecekteki fırsatlarını nasıl etkileyebileceğini açıklayınız!
13. AŞAĞIDAKİ özelliklere sahip bir kaynağın geçerli olup olmadığını değerlendiriniz: çok sayıda reklam içeren, yazarı ve tarihi olmayan bir web sitesi! Değerlendirmenizi açıklayınız!
14. Bir arkadaşın fotoğrafını onun bilgisi olmadan sosyal ağlarda paylaşmak etik açıdan kabul edilebilir mi? Nedenini açıklayınız!
15. Bir sosyal ağda tanımadığınız bir kişiden gelen arkadaşlık isteğini kabul etmenin güvenli olup olmadığını değerlendiriniz!
16. İnterneti güvenli kullanma konusunda en az üç kural içeren kısa bir etkileşimli sunum veya infografik oluşturunuz.
 - Kendi yaşınızdaki öğrenciler için bir mini rehber hazırlayınız! Rehberde şu bilgiler bulunmalıdır: internet tanımı, iki güvenlik tavsiyesi ve iyi bir dijital ayak izi ile kötü bir dijital ayak izi örneği.
17. Geçerli web kaynaklarından bulduğunuz bilgileri kullanarak dijital bir içerik (poster, kısa metin, görsel veya mini video) oluşturunuz!

KONU 3: C++ PROGRAMLAMA

Modern toplumda, programlama konusunda temel bilgi sahibi olmanın okuma ve yazma kadar gerekli olduğu kabul edilmektedir. Programlama aracılığıyla, bir problemin çözümü için gerekli adımların bilgisayarın anlayabileceği bir dille nasıl ifade edileceği konusunda farklı bir düşünme biçimi geliştirilmeye başlanır. Ancak herhangi bir programlama dilinin sözdizimi öğrenilmeden önce, her görevin arkasındaki mantığın kavranması gerekmektedir. Bu nedenle, programlama öğrenimindeki ilk adım kod yazmak değil, programcılara özgü düşünme biçiminin – algoritmik düşünmenin – geliştirilmesi için alıştırmayı yapmaktır.

Yeni Kavramlar

- Bilişimsel kavram
- Programlama ortamı
- Derleme
- Hata ayıklama (Debugging)
- Sıralı yapı
- Başlatma (Initialization)
- İç içe koşul
- Karşılaştırma ifadesi
- Döngü sayacı

3.1

Mantıksal ve Algoritmik Ödevler

Bu konuyu tamamladığında...

- ✓ Günlük hayattaki basit mantıksal durumları tanıma,
- ✓ Sağduyu ve mantıksal düşünmeyi kullanarak problemleri çözme,
- ✓ Yönergeleri adım adım takip etme,
- ✓ Oyunlarda ve görevlerde örüntüleri ve kuralları fark etme,
- ✓ Doğru ve yanlış ifadeler arasında ayırım yapma.



TEKRARLAMA SORULARI:

1. Scratch nedir?
2. Yarışma türü görevler nedir?
3. Değişkenler nedir?



AŞAĞIDAKİ KONULARDA ŞUNLARI ÖĞRENECEKSİN:

- Mantıksal bir görevi matematiksel hesaplamadan ayırt etme,
- Algoritmik bir problemin çözüm yöntemini analiz etme,
- Mantıksal düşünme sürecini adım adım takip etme,
- Temel kavramları kullanma: koşul, komut, sonuç,
- Basit bir algoritmanın adımlarını belirleme,
- Algoritmik düşünmenin neden önemli olduğunu anlama.

Programlamanın Yeni Bir Düşünme Dili Olarak Kullanılması

Programlama, matematik, mantık ve problem çözmeyi bir araya getiren bir disiplindir. Bir problemin analiz edilmesiyle başlayan, adımlar aracılığıyla çözümün planlanmasıyla devam eden ve uygulamayla tamamlanan bir süreçtir. Bu ilk derste hiçbir kod satırı kullanılmayacaktır. İlgi çekici ve basit örnekler aracılığıyla mantıksal ve algoritmik düşünmenin temel özellikleri ele alınacaktır. Bu örnekler, bir problemin nasıl oluşturulduğunu, nasıl analiz edildiğini ve çözümüne nasıl yaklaşıldığını anlamaya yardımcı olacaktır.

Mantıksal görev, bir problemin karmaşık hesaplamalara ihtiyaç duyulmadan, adım adım ve mantıklı düşünülerek çözülebildiği her durumu ifade eder. İlk olarak amaç belirlenir: Ne elde edilmesi gerekiyor? Daha sonra verilen tüm bilgiler değerlendirilir. Bu bilgilerin arasındaki bağlantılar analiz edilir ve hangilerinin önemli olduğu belirlenir. Son olarak, olası çözümler planlanır.

ÖRNEK 1

Nesnenin Yeri

Bir masanın üzerinde üç kutu vardır: kırmızı, mavi ve sarı. Kutulardan yalnızca birinde bir nesne bulunmaktadır. Her kutunun üzerinde bir ifade yazılıdır:

- **Kırmızı:** "Nesne bu kutudadır."
- **Mavi:** "Nesne kırmızı kutuda değildir."
- **Sarı:** "Nesne bu kutuda değildir."

Yalnızca bir ifade doğrudur. Nesne hangi kutudadır?

Analiz bir varsayımla başlar. Eğer kırmızı kutunun üzerindeki ifade doğruysa, nesne oradadır, ancak sarı kutunun üzerindeki ifade (nesnenin sarı kutuda olmadığı) de doğru olur. Bu durumda iki doğru ifade ortaya çıkar ve bu izin verilen bir durum değildir. Eğer mavi kutunun üzerindeki ifade doğruysa, yani nesne kırmızı kutuda değilse, nesne sarı kutudadır, bu durumda sarı kutunun ifadesi de doğru olur. Yalnızca bir ifadenin doğru olduğu tek seçenek, nesnenin mavi kutuda olmasıdır. Bu, hesaplama yapmadan mantığın kullanıldığı bir örnektir.

Mantıksal görevlerin çözümü sırasında geliştirilen ikinci önemli beceri, örüntüleri (şablonları) tanıma ve koşulları ilişkilendirmedir. Bu, verilen belirli sayıdaki bilgiler arasındaki ilişkinin incelenmesi ve bu bilgilerden hangi sonuçların çıkarılabileceğinin belirlenmesi anlamına gelir.

ÖRNEK 2



Yol Üzerinde Hareket Etme

Bir robot, 5 satır ve 5 sütundan oluşan bir tablo üzerinde hareket etmektedir. Sol üst köşeden başlamaktadır. Her harekette yalnızca aşağıya veya sağa doğru ilerleyebilir. Sağ alt köşeye ulaşmak için kaç farklı yol vardır?

Bu, her hareketin belirli kurallarla sınırlandırıldığı klasik bir “yol sayma” türü görevidir. Henüz herhangi bir hesaplama yapılmasa da, tüm olası hareket kombinasyonları düşünülebilir: kaç kez sağa ve kaç kez aşağıya gidileceği belirlenebilir. Bu durum, çözümü yapılandırma ve olası seçeneklerin sayısını değerlendirme becerisini geliştirir – algoritmik düşünmenin temelini oluşturur.

Mantıksal görevlerin bir sonraki önemli yönü, koşulların ve sonuçların belirlenmesidir. Birçok görevde, uyulması gereken kurallar veya sınırlamalardan oluşan bir küme bulunur ve bu kurallara dayanarak neyin gerçekleşebileceği veya gerçekleşemeyeceği sonucuna varılır. Buna koşulların analizi denir.

ÖRNEK 3



Eleme Yoluyla Bilmece

Beş öğrenci sırada oturmaktadır. Bilinenler:

- Petar, Marina’nın yanında oturmamaktadır.
- Saşko, Ana’nın sağında oturmaktadır.
- Ana, Petar’ın yanında değildir? Kim nerede oturmaktadır?

Mantıksal eleme ve çıkarım yoluyla doğru sıralama bulunabilir. Bu tür görevler dikkatli okumayı ve mümkün olmayan seçeneklerin aşamalı olarak elenmesini gerektirir. Sayılar kullanılmaz, bunun yerine kişiler arasındaki ilişkiler ve sınırlamalar dikkate alınır. Bu, sistematik düşünme ve mantıksal kesinliğin geliştirilmesi için mükemmel bir çalışmadır.



DAHA FAZLA BİLGİ EDİNMEK İSTEYENLER İÇİN

Algoritmalar günlük hayatta sıklıkla kullanılır: tarifleri takip etmek, navigasyon, hatta sudoku çözmek. “Dabar” yarışması, öğrencilerin bunlara benzer mantık sorularını çözdüğü uluslararası bir etkinliktir. Eğer bu örnekleri beğendiysen, yarışmanın resmi sayfasındaki diğer soruları deneyebilir veya kendi mantık bulmacalarını oluşturup arkadaşlarınla çözebilirsin.



BİLGİ KONTROLÜ SORULARI VE ETKİNLİKLER

1. Algoritmik görevler bağlamında mantıksal düşünme ne anlama gelir?
2. Bir görevi çözerken adımlar neden önemlidir?
3. Bir ifadenin mantıksal olarak imkânsız olduğu ne zaman söylenir?
4. Sayı içermeyen görevler programlama öğrenirken neden yararlı olabilir?
5. AŞAĞIDAKİlerden ikisi doğruysa mantıksal olarak yanlış olan cümleyi bulunuz:
 - Mavi, Sarı'nın solunda yaşar.
 - Yeşil, Mavi'nin sağında yaşar.
 - Sarı, Mavi'nin yanında değildir.
6. Üç arkadaşın her birinin bir kitabı vardır: roman, ansiklopedi ve masal. Bilinenler: Jana ansiklopedi okumuyor, Boban ise masal okumuyor. Kim ne okuyor?
7. En fazla üç koşullu ve en az iki olası çözümü olan kendi mantıksal görevinizi oluşturunuz!!

Öğrendiklerini Hatırla!

- Mantıksal görevler, verilerin ve ifadelerin dikkatli bir şekilde analiz edilmesini gerektirir.
- Algoritmik düşünme, açıkça tanımlanmış amaçlar ve adımlarla başlar.
- Problem çözme becerisini geliştirmek için sayılara veya koda ihtiyaç yoktur.
- Her problem, olasılıkların belirlenmesi ve mümkün olmayan seçeneklerin elenmesiyle sistematik olarak çözülebilir.



Bu konuyu tamamladığında...

- ✓ Olayların sıralaması hakkında mantıksal düşünebilirsin,
- ✓ Günlük görevlerde sıralamanın ve kuralların önemini fark edebilirsin,
- ✓ Karar verme stratejilerini kullanabilirsin,
- ✓ Problemleri çözerken kendini organize edebilirsin,
- ✓ Bir şeyin ne zaman daha verimli yapılabileceğini fark edebilirsin..

**TEKRARLAMA SORULARI:**

1. Koşulların analizi neyi ifade eder?
- 2.Yönlerin sınırlandırıldığı (yalnızca sağa veya aşağıya hareket edilebilen) bir labirent yolu düşününüz. Hedefe ulaşmak için kaç adım gereklidir.

**AŞAĞIDAKİ KONULARDA ŞUNLARI ÖĞRENECEKSİN:**

- Veri yapılarını açıklayabilirsin,
- Görevleri optimize etme stratejilerini tanıyabilirsin,
- “Böl ve yönet” ve “açgözlü optimizasyon” gibi yaklaşımları ayırt edebilirsin,
- Kaynak dağılımının neden önemli olduğunu anlayabilirsin,
- Kavramları günlük problemlerde uygulayabilirsin.

Bilişim dünyasında kavramlar, karmaşık görevleri çözme becerisinin üzerine kurulduğu temel unsurlardır. Verilerin düzenlenme, yönetilme ve optimize edilme biçimleri hakkında temel bir anlayış olmadan, programlama için sağlam bir temel geliştirilemez. Bilgisayarın ve insanın karşılaştığı her problemin bir yapısı, sınırlamaları ve ulaşılması gereken bir amacı vardır. Bu nedenle, mantıklı kararlar vermeye ve kaynakları verimli kullanmaya yardımcı olan tekniklerin öğrenilmesi gerekir.

Bunun yanı sıra, bilişim kavramları çözümlerin yalnızca doğru değil, aynı zamanda en uygun çözümler olmasını sağlar. Verileri düzenleme yöntemleri, çözüm modellerini anlama ve belirli bir problem için doğru stratejiyi seçme, bu ders aracılığıyla geliştirilecek

becerilerin bir parçasıdır. Bunlar, en iyi yolun seçilmesi, görevlerin dağıtılması veya zamanın düzenlenmesi gerektiğinde gerçek hayatta da kullanılan becerilerdir.

Veri Yapıları

Veri yapıları, bilgilerin daha kolay kullanılabilmesi için düzenlenme ve saklanma biçimini ifade eder. Örneğin, sınıf listesindeki öğrenci listesi veya ders programı tablosu, verilerin saklanmasına hizmet eden gerçek yaşam örnekleridir. Bilişim alanında bu tür yapılar, kullanım amaçlarına ve uygulama alanlarına bağlı olarak kuyruklar, kümeler, listeler, ağaçlar veya graflar olabilir. Bu yapıların her birinin kendine özgü avantajları vardır ve belirli bir verimlilik veya bilgiye erişim yöntemi gerektiğinde kullanılır.

ÖRNEK 1

Bir kütüphanede bütün kitapların rastgele yerleştirildiğini düşününüz. Bu durumda aranan kitabı bulmak zor ve zaman alıcı olacaktır. Ancak kitaplar türe, yazara veya başlığa göre düzenlenmişse, aranan kitaba ulaşmak çok daha kolay olur. Bilgisayarlarda da durum aynıdır, veriler yapılar hâlinde düzenlendiğinde, daha hızlı aranabilir ve daha verimli kullanılabilir.

Kaynakların Dağıtılması

Kaynakların dağıtılması, sınırlı kaynakların (zaman, alan, enerji) görevler veya süreçler arasında nasıl paylaştırıldığını ifade eden bir kavramdır. Bilişimde bu, işlemci süresinin birden fazla program arasında nasıl paylaştırılacağı veya belleğin farklı görevler için nasıl kullanılacağı anlamına gelebilir. Kaynakların doğru şekilde dağıtılması, görevlerin çok daha hızlı ve verimli bir biçimde yürütülmesini sağlayabilir, buna karşılık, hatalı bir dağıtım gecikmelere ve kaynak israfına yol açar.

ÖRNEK 2

Bir hafta içinde beş ders için bir çalışma planı hazırlaman gerektiğini düşününüz. Eğer bir derse gereğinden fazla zaman ayrılırsa, diğer dersler yeterince çalışamaz. İyi bir plan, her dersin önemine veya zorluk derecesine göre zamanın dengeli bir şekilde dağıtılmasını gerektirir. Bu, bilgisayarların görevleri yönetirken kullandığı temel kavramlardan biridir.

Açgözlü optimizasyon

Açgözlü (greedy) yaklaşım, her adımda o an için en iyi görünen seçeneğin tercih edildiği, gelecekteki sonuçların ise dikkate alınmadığı bir stratejidir. Bu yaklaşım her zaman en iyi çözümü sağlamasa da, çoğu durumda çok daha az kaynak kullanarak kabul edilebilir bir sonuca ulaşılmasını sağlar. Özellikle, mükemmel bir çözümden ziyade hızlı bir sonuca ulaşmanın önemli olduğu durumlarda kullanılır.

ÖRNEK 3



100 dinarın olduğunu ve mümkün olduğunca fazla atıştırmalık satın almak istediğini düşününüz. Her adımda gördüğün en ucuz ürünü seçersen daha fazla ürün alabilirsin, ancak en kaliteli ürünleri seçmemiş olabilirsin. Bu, bazı algoritmaların hemen en iyi yerel çözümü seçmesine benzeyen açgözlü yaklaşıma bir örnektir.

Böl ve yönet

“Böl ve yönet” (divide and conquer), büyük bir problemin daha küçük ve çözülmesi daha kolay alt problemlere ayrıldığı bir stratejidir. Daha sonra bu alt problemler ayrı ayrı çözülür ve elde edilen sonuçlar birleştirilerek tek bir nihai çözüm oluşturulur. Bu yöntem, problemin karmaşıklığını azalttığı ve çözümün verimliliğini artırdığı için bilişimde yaygın olarak kullanılmaktadır. Problemi bir bütün olarak ele almak yerine, onu parçalara ayırarak çözmeyi esas alır.

ÖRNEK 4



Bir sunum hazırlanırken metnin tamamı tek seferde yazılmaz. Önce başlık hazırlanır, ardından içerik bölümler hâlinde oluşturulur, görseller eklenir ve son aşamada tüm bölümler bir araya getirilir. Aynı yaklaşım bilgisayar tabanlı çözümlerde de uygulanabilir.

BİLGİ KONTROLÜ SORULARI VE ETKİNLİKLER



1. Veri yapıları neyi ifade eder?
2. Kaynakların dağıtılması neden önemlidir?
3. Açgözlü yaklaşım her zaman en iyi çözümü sağlar mı?
4. Veri yapısına örnek olabilecek günlük yaşamdan bir örnek veriniz!
5. Zamanın planlı bir şekilde dağıtıldığı bir çalışma planı hazırlayınız!
6. Bir veri yapısından yararlanarak bir kütüphaneyi nasıl düzenleyeceğinizi açıklayınız!

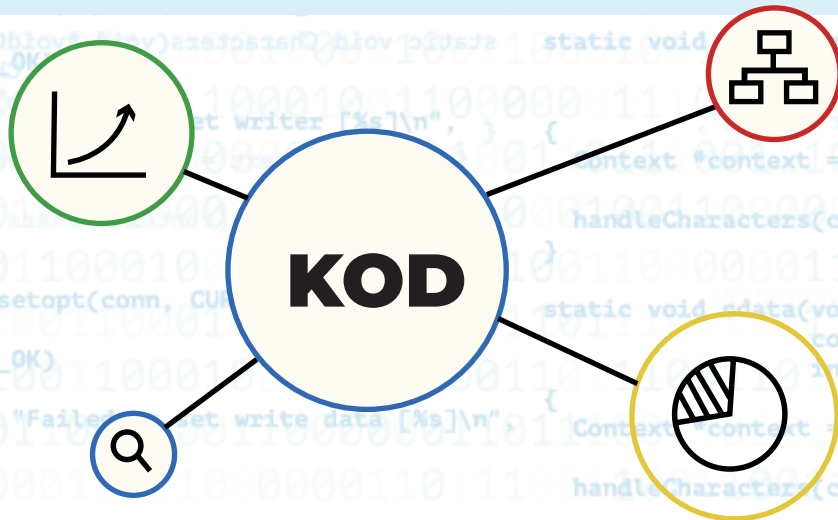
Öğrendiklerini Hatırla

- Veri yapıları, bilgilerin verimli bir şekilde yönetilmesinin temelini oluşturur.
- Kaynakların dağıtılması, zamanın ve alanın daha etkili kullanılmasını sağlar.
- Açgözlü yaklaşım, her adımda en iyi yerel çözümü seçer, ancak her zaman en iyi genel sonuca ulaştırmaz.
- Böl ve yönet yaklaşımı, karmaşık görevleri daha küçük parçalara ayırarak daha verimli çözmeyi sağlar.
- Bilişim kavramları yalnızca programlamada değil, günlük yaşamda da uygulanabilir.



DAHA FAZLA BİLGİ EDİNMEK İSTEYENLER İÇİN

Programlamada kullanılan aynı kavramlar, optimizasyon, dağıtım ve modelleme, ekonomi, yönetim ve matematikte de incelenmektedir. Örneğin, ulaşımда güzergâhların optimize edilmesi veya çalışma vardiyalarının planlanması. Grafikler gibi veri yapıları, sosyal ağlarda insanlar arasındaki bağlantıları analiz etmek için kullanılmaktadır. Açgözlü algoritmalar, hızlı karar vermek amacıyla mobil uygulamalarda (örneğin, en kısa yolun seçilmesi) kullanılmaktadır. Bunların gerçek sistemlerde nasıl kullanıldığını araştırınız ve bunları nerelerde uygulayabileceğinizi düşününüz.



3.3

İkili Sayılar: Gösterim ve Kullanım

Bu konuyu tamamladığında...

- ✓ Sayıları ve sayıların farklı gösterim biçimlerini ayırt edebilirsin,
- ✓ Günlük yaşamda bir değer olarak ondalık sayıları kullanabilirsin,
- ✓ Sayılarla çalışırken doğruluğun önemini fark edebilirsin,
- ✓ Farklı ölçü birimlerini kullanabilir ve temel aritmetik işlemleri yapabilirsin,
- ✓ Çevremizdeki teknolojilerde sayıların nerede ve nasıl kullanıldığını açıklayabilirsin.



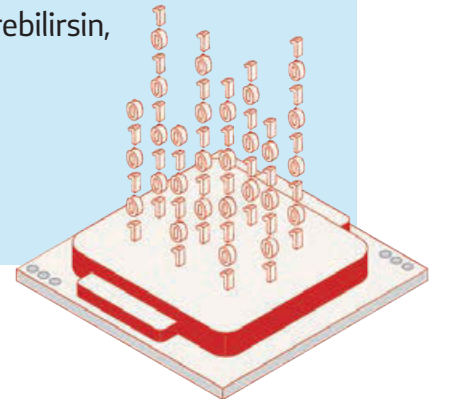
TEKRARLAMA SORULARI:

1. “Böl ve yönet” stratejisi ne anlama gelir?
2. Bir görevin parçalara ayrılmasının daha kolay çözüme ulaşmayı sağladığı bir senaryo yazınız!



AŞAĞIDAKİ KONULARDA ŞUNLARI ÖĞRENECEKSİN:

- İkili sistemin ne olduğunu ve nasıl çalıştığını açıklayabilirsin,
- Tam sayıları ikili gösterimle ifade edebilirsin,
- İkili ve ondalık sistem arasındaki farkı ayırt edebilirsin,
- İkili sayıları bilgisayarların temel çalışma prensibiyle ilişkilendirebilirsin,
- Bilgileri bitler ve baytlar olarak ifade edebilirsin,
- İkili sayıları pratik durumlarda uygulayabilirsin.



Dijital çağda, neredeyse her teknoloji sayıları kullanan süreçlerle desteklenmektedir. Sayılar yalnızca matematiğin bir parçası değil, aynı zamanda insan ile makine arasındaki iletişimin temelidir. Her bilgisayar, telefon, dijital saat veya mikroişlemci sayıları kullanarak iletişim kurar ve bilgileri işler. Ancak bu sayılar günlük hayatta kullanılan sayılarla aynı değildir. Bilgisayarlar, kesin, hızlı ve güvenilir işlem yapmayı sağlayan özel bir sistem kullanır: **ikili sistem**.

Ekranada görülen her şey – görüntüler, sesler, metinler – temel olarak bir dizi ikili değere dönüştürülür. Bu, bir bilgisayarın aldığı veya gönderdiği her bilginin sonunda yalnızca iki olasılıktan oluşan değerlere, yani 0 ve 1 sayılarına indirgenmesi anlamına gelir. Başka bir ifadeyle, bilgisayarlar ondalık sayılarla (0–9) değil, dijital mantığın temelini oluşturan ikili sayılarla “düşünür”.

Bilgisayar biliminde sayıların gösterimi, yalnızca iki rakamı kullanan **ikili sisteme** dayanır: 0 ve 1. Bu sistem tercih edilmiştir, çünkü elektronik cihazların iki durumu algılaması – elektrik akımının varlığı veya yokluğu, yani yüksek veya düşük gerilim değeri – çok daha basit ve verimlidir. Bu basitlik sayesinde, büyük miktarda bi.

ÖRNEK

5 sayısını ikili sistemde göstermek.

2'ye bölme yöntemi kullanılır:

$$5 / 2 = 2 \text{ kalan } 1$$

$$2 / 2 = 1 \text{ kalan } 0$$

$$1 / 2 = 0 \text{ kalan } 1$$

Kalanlar aşağıdan yukarıya doğru okunur: 101 Yani, 5 sayısının ikili sistemdeki gösterimi 101'dir.

Bilgilerin sıfırlar ve birlerle ifade edilmesine bitler denir. Bir bit, en küçük veri miktarıdır ve 0 veya 1 değerlerinden biri olabilir. Sekiz bit bir bayt oluşturur ve bu, verilerin depolanması ve aktarılmasında en yaygın kullanılan ölçü birimidir.

Fotoğraf makineleri gibi dijital cihazlarda, her pikseli belirli bir değerle kodlamak için ikili sistem kullanılır. Her piksel için ne kadar fazla bit kullanılırsa, renklerin ve ayrıntıların doğruluğu o kadar artar. Her cihazda, hatta en küçük bileşenlerde bile, ikili kod verilerin işlenmesi ve depolanmasında temel bir role sahiptir.

A harfi ikili kodda nasıl kodlanır?

ASCII standardında A harfinin sayısal değeri 65'tir. İkili sistemde gösterimi şöyledir:

$$65 / 2 = 32 \text{ kalan } 1$$

$$32 / 2 = 16 \text{ kalan } 0$$

$$16 / 2 = 8 \text{ kalan } 0$$

$$8 / 2 = 4 \text{ kalan } 0$$

$$4 / 2 = 2 \text{ kalan } 0$$

$$2 / 2 = 1 \text{ kalan } 0$$

$$1 / 2 = 0 \text{ kalan } 1$$

Aşağıdan yukarıya doğru okunur: 1000001

Bu, A harfinin bilgisayar belleğinde 1000001 olarak saklandığı anlamına geli.

BİLGİ KONTROLÜ SORULARI VE ETKİNLİKLER



1. İkili sayı nedir?
2. Bilgisayarlar neden ikili sistemi kullanır?
3. 9 sayısını ikili sisteme dönüştürünüz!
4. Bayt neyi ifade eder ve kaç bit içerir?
5. 10, 12 ve 15 sayılarını ikili sisteme dönüştürünüz!
6. Tam olarak 256 farklı değerın ifade edilebilmesi için kaç bit gerekir?

Öğrendiklerini Hatırla!

- İkili sistem yalnızca iki rakam kullanır: 0 ve 1.
- Bilgisayarlar, bileşenlerinin elektronik yapısı nedeniyle ikili sayılarla çalışır.
- Bilgisayardaki her sayı, harf veya veri ikili biçimde gösterilir.
- Bit ve bayt, verileri ölçmek için kullanılan temel birimlerdir.
- Ondalık sistemden ikili sisteme dönüşüm, 2'ye bölme yöntemiyle yapılır.



DAHA FAZLA BİLGİ EDİNMEK İSTEYENLER İÇİN

- İkili aritmetiğin nasıl olduğunu ve ikili sayıların nasıl toplandığını inceleyiniz.
- Bilgisayarın görüntüleri ve sesleri temsil etmek için ikili sistemi nasıl kullandığını araştırınız!
- Ondalık sayıların (floating point) gösteriminde kullanılan IEEE formatları hakkında okuyunuz!
- Negatif sayıların bilgisayarda nasıl gösterildiğini araştırınız (örneğin, "ikiye tümleyen" – two's complement yöntemiyle)



3.4

Kodlama ve Şifreleme Temelleri

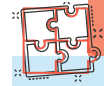
Bu konuyu tamamladığında...

- ✓ İkili sayıları tanıyabilir ve dijital gösterimlerle ilişkilendirebilirsin,
- ✓ Mantıksal ve algoritmik adımları ayırt edebilirsin,
- ✓ Bilginin ne olduğunu ve sembollerle nasıl ifade edildiğini anlayabilirsin.



TEKRARLAMA SORULARI:

1. 6 bit ile kaç farklı değer ifade edilebilir?
2. 100 sayısının ikili biçimde nasıl gösterileceğini yazınız!



AŞAĞIDAKİ KONULARDA ŞUNLARI ÖĞRENECEKSİN:

- Kodlamanın ne olduğunu ve nerelerde kullanıldığını açıklayabilirsin,
- Kodlama ile şifreleme arasındaki farkı ayırt edebilirsin,
- Günlük hayatta kullanılan kodlama örneklerini tanıyabilirsin,
- Şifrelerin neden kullanıldığını ve bilgilerin korunmasına nasıl yardımcı olduğunu anlayabilirsin,
- Açık ve gizli mesajlar arasındaki farkı ayırt edebilirsin,
- Verilerin farklı şekillerde nasıl ifade edilebileceği üzerine düşünebilirsin.

Dijital iletişim dünyasında, bilgilerin güvenilir, doğru ve anlaşılır bir şekilde ifade edilmesi, aktarılması ve saklanması için sürekli bir ihtiyaç vardır. Bu nedenle, insan uygarlığının ilk dönemlerinden itibaren mesajları bir biçimden başka bir biçime dönüştürmek için farklı yöntemler geliştirilmiştir. Bu süreç kodlama olarak adlandırılır. Kodlama sayesinde, yazılı bir dil kullanılmadan bile anlamların aktarılması mümkün olmuştur, özel bir alan olarak şifreleme ise bu mesajların gizli kalmasını sağlamıştır. Programlamayı ve dijital veri işlemeyi anlamak için öncelikle bu temel kavramların öğrenilmesi gerekir, çünkü bunlar modern dünyada bilgi güvenliği ve bilgi alışverişinin temelini oluşturur.

Kodlama ve şifreleme arasındaki fark, günlük konuşmalarda kullanıldığında çoğu zaman karıştırılır, ancak bilişim alanında bu kavramların çok kesin anlamları vardır. Kodlama, bilginin daha kolay okunması, işlenmesi veya saklanması amacıyla bir biçimden başka bir biçime dönüştürülmesini ifade eder. Örneğin, bilgisayarda harfleri ve sembolleri göstermek için ikili kodun kullanılması bir kodlama biçimidir. Diğer taraftan şifreleme, mesajın gerçek içeriğini gizleyen ve yalnızca yetkili kişilerin şifresini çözerek okuyabilmesini sağlayan bir yöntemdir. Bu kavramların anlaşılması, veri güvenliği, dijital sertifikalar ve bilgisayar ağları hakkında gelecekte öğrenilecek bilgiler için gerekli bir temel oluşturur.

Kodlama

İnsanlık tarihinde kodlama, özellikle bilgilerin yalnızca belirli alıcılar tarafından anlaşılması gerektiği durumlarda, insanlar arasında mesajların iletilmesini sağlayan bir yöntem olarak kullanılmıştır. Bu tür bir kodlama biçiminin en bilinen tarihsel örneklerinden biri, telgraf iletişimde kullanılan Mors alfabesidir. Bu sistemde, alfabedeki her harf kısa ve uzun sinyallerin (nokta ve çizgilerin) birleşimiyle gösterilmiştir. Günümüzde basit bir yöntem gibi görünse de Mors kodu, basit elektrik darbeleri aracılığıyla iletişimi mümkün kıldığı için devrim niteliğinde bir kodlama yöntemi olarak kabul edilmiştir. Bilişim açısından bu kodlama yöntemi, sembol (veya sinyal) kombinasyonlarının anlamları doğru bir şekilde ifade etmek için kullanıldığı dijital bilgi gösteriminin öncüsü olarak kabul edilir.

ÖRNEK 1



Mors Kodu

Mors kodunda "A" harfi, bir kısa noktanın ardından bir uzun çizgi geldiğini ifade eden ".-" şeklinde gösterilir. Aynı şekilde "B" harfi "-..." olarak gösterilir. İngiliz alfabesindeki tüm harflerin, sayıların ve bazı özel işaretlerin bu alfabede kendine ait bir kodu vardır. Mors alfabesinin kullanımı, alıcılar tarafından anlaşılan elektrik sinyallerinin veya ışık darbelerinin gönderilmesine dayanıyordu. Bu sayede, sesli iletişimin mümkün olmadığı zor koşullarda bile bilgilerin aktarılması sağlanıyordu.

Bilişim sistemlerinde kodlama yalnızca verileri gizlemek amacıyla değil, aynı zamanda makinelerin işleyebileceği bir biçimde temsil etmek için de kullanılır. Bilgisayar dünyasındaki en temel ve en önemli kodlama türü ikili kodlamadır. Tüm dijital dünya, bilgilerin yalnızca iki durumla – 0 ve 1 – ifade edilmesine dayanır. Bu ikili değerler, sayılar, metinler, görüntüler, sesler ve videolar gibi her tür verinin gösteriminde kullanılır. Örneğin, metindeki her karakter ASCII (American Standard Code for Information Interchange) gibi standart bir kod aracılığıyla ifade edilir, burada her harfin, sayının veya özel işaretin kendine ait benzersiz bir ikili değeri vardır.

ÖRNEK 2



ASCII Kodlama

ASCII standardına göre "A" harfinin değeri 65'tir. İkili gösterimde bu değer 01000001 şeklindedir. Benzer şekilde, "a" harfinin değeri 97'dir ve ikili biçimde 01100001 olarak gösterilir. Bu tür kodlar sayesinde metinler, bilgisayarın kolayca saklayabileceği ve işleyebileceği ikili değer dizilerine dönüştürülebilir. Bu durum, kodlamanın her zaman gizli bir işlev taşımadığını, aynı zamanda insan ile makine arasındaki iletişim için temel bir unsur olduğunu göstermektedir.

Şifreleme

Şifreleme, mesajın anlamını gizleme süreci olarak, güvenli iletişimin önemli bir parçası olmuştur. İlk ve en bilinen örneklerden biri, adını Julius Caesar'dan alan ve onun tarafından askeri mesajları şifrelemek için kullanılan Sezar şifresidir. Bu şifreleme yönteminin temel ilkesi, mesajdaki harflerin alfabede belirli bir sayı kadar kaydırılmasıdır. Böylece, orijinal mesajdaki her harf, alfabede birkaç sıra ileride bulunan başka bir harfle değiştirilir. Günümüzde bu yöntem zayıf ve kolayca çözülebilir olarak kabul edilse de, aynı anahtarın hem şifreleme hem de şifre çözme işlemlerinde kullanıldığı simetrik şifreleme kavramına iyi bir giriş örneği oluşturmaktadır.

ÖRNEK 3



Sezar Şifresi

Sezar şifresinde 3 konumluk bir kaydırma kullanılırsa, "A" harfi "D", "B" harfi "E" olur ve bu şekilde devam eder. Örneğin, "INFO" kelimesi "LSKR" şeklinde dönüştürülür. Mesajın şifresini çözmek için alıcının, üç konum geriye doğru bir kaydırma kullanıldığını bilmesi gerekir. Bu basit yöntem, bilginin nasıl gizlenebileceğini, ancak anahtarın aktarımının ne kadar güvenli olması gerektiğini de gösterir, eğer bir kişi anahtarı ele geçirirse mesajı kolayca çözebilir.

Modern dijital toplum, iletişimlerin, işlemlerin ve veri depolamanın güvenliğini sağlamak için gelişmiş şifreleme yöntemlerine dayanmaktadır. Geçmişteki Sezar şifresi gibi basit tekniklerin aksine, günümüzdeki yöntemler karmaşık matematiksel algoritmalar ve kriptografik protokoller kullanır. En yaygın kullanılan iki model simetrik ve asimetrik şifrelemedir. Simetrik şifrelemede, verilerin şifrelenmesi ve şifre çözülmesi için aynı anahtar kullanılır, bu nedenle iletişimde bulunan her iki tarafın da aynı anahtara sahip olması ve bu anahtarı gizli tutması gerekir. Diğer taraftan, RSA algoritması gibi asimetrik şifrelemede iki anahtar kullanılır: açık anahtar ve gizli anahtar. Bu yöntem, önceden gizli anahtar paylaşımına gerek olmadan güvenli iletişim ve kimlik doğrulama olanağı sağlar.

RSA ile Asimetrik Şifrelem

Bir kişi internet üzerinden başka bir kişiye mesaj göndermek istediğinde, mesajı alıcının açık anahtarıyla şifreleyebilir. Mesaj ulaştığında, alıcı mesajın şifresini kendi özel (gizli) anahtarıyla çözer. Bu şekilde, bir kişi mesajı ele geçirse bile özel anahtar olmadan mesajı okuyamaz. RSA ve benzeri algoritmalar, günümüzde HTTPS, çevrim içi bankacılık ve dijital imza yoluyla gerçekleştirilen güvenli iletişimlerin temelini oluşturur.

Kriptografi, bilgileri uygun bir anahtar olmadan anlaşılmayacak bir biçime dönüştürerek koruma bilimidir. Antik çağlardan beri kullanılmıştır, ancak dijital çağda internetin, bankacılık sistemlerinin, sağlık hizmetlerinin ve devlet kurumlarının günlük işleyişinin vazgeçilmez bir parçası hâline gelmiştir. Sıradan kodlamadan farklı olarak kriptografinin amacı yalnızca verileri dönüştürmek değil, aynı zamanda onları yetkisiz erişimlere karşı korumaktır. Kriptografik sistemler matematiksel ilkelere ve algoritmalara dayanır ve kimlik doğrulama, gizlilik, veri bütünlüğü ve değişmezlik özelliklerini sağlar. Bir kişi bir web sitesine giriş yaptığında, çevrim içi işlem gerçekleştirdiğinde veya elektronik posta gönderdiğinde, bilginin güvenliğini sağlayan kriptografidir.

Modern kriptografi, okunabilir verilerin (diğer adıyla “açık metin”) kriptografik algoritmalar ve anahtarlar kullanılarak şifrelenmiş verilere (“şifreli metin”) dönüştürülmesi ilkesine dayanır. Bu işlem sonucunda veri, doğru şifre çözme anahtarına sahip olmayan herkes için anlamsız hâle gelir. Bu süreç, hem şifreleme hem de şifre çözme işlemlerinde aynı anahtarın kullanıldığı simetrik veya iki farklı anahtarın – açık ve özel anahtarın – kullanıldığı asimetrik olabilir. Simetrik sistemler daha hızlıdır, ancak anahtarların güvenli bir şekilde paylaşılmasını gerektirir, asimetrik sistemler ise daha fazla esneklik sunar ve anahtarın güvenli şekilde aktarılmasının mümkün olmadığı durumlarda kullanılır. Kriptografinin uygulamaları elektronik posta, bankacılık işlemleri, dijital imzalar, sertifikalar ve bilgisayar veya mobil cihazlarda depolanan verilerin şifrelenmesinde görülmektedir. Bu teknolojiler olmadan dijital dünyaya duyulan güven neredeyse imkânsız olurdu.

BİLGİ KONTROLÜ SORULARI VE ETKİNLİKLER

1. "Kodlama" kavramı neyi ifade eder ve şifrelemeden farkı nedir?
2. Simetrik ve asimetrik şifreleme arasındaki farkı açıklayınız!
3. Sezar şifresinin temel unsurları nelerdir ve nasıl çalışır?
4. RSA algoritmasında açık ve özel anahtarın rolü nedir?
5. AŞAĞIDAKİ mesajı Mors koduna dönüştürünüz: "HELLO"!
6. "DATA" kelimesini anahtar 3 kullanarak Sezar şifresi ile şifreleyiniz!
7. Bir elektronik posta gönderirken RSA algoritmasının nasıl kullanılacağını açıklayınız!
8. Simetrik şifreleme için gizli bir anahtarınız olduğunu düşününüz! Bu anahtarı başka bir kişinin ele geçirmesini engelleyecek şekilde bir çalışma arkadaşınızla nasıl paylaşacağınızı açıklayınız!

Öğrendiklerini Hatırla!

- Kodlama, bilginin daha kolay işlenmesi, saklanması veya aktarılması amacıyla bir biçimden başka bir biçime dönüştürülmesi sürecidir.
- Şifreleme, verilerin korunması için kullanılır, bilgi, anahtar olmadan okunamayacağı şekilde şifrelenir.
- Kodlamaya örnek olarak, harflerin nokta ve çizgilerle gösterildiği Mors kodu verilebilir.
- Sezar şifresi, alfabedeki harflerin kaydırılmasına dayanan en eski şifreleme biçimlerinden biridir.
- RSA, güvenli iletişim için açık ve özel anahtar kullanan asimetrik bir kriptografik algoritmadır.
- Simetrik şifrelemede, şifreleme ve şifre çözme işlemleri için aynı anahtar kullanılır.



DAHA FAZLA BİLGİ EDİNMEK İSTEYENLER İÇİN

- Modern kriptografide, RSA'nın yanı sıra AES (Advanced Encryption Standard) ve ECC (Elliptic Curve Cryptography) gibi diğer algoritmalar da kullanılmaktadır.
- Dijital sertifikalar ve SSL protokolü, internet iletişiminde güvenliğini sağlamak için şifreleme kullanır.
- Hash fonksiyonları (örneğin SHA-256), verilerin bütünlüğünü kontrol etmek için kullanılır.
- Kuantum kriptografisi, kırılmaz şifreler oluşturmak için kuantum olaylarını kullanan gelişmiş bir alandır.
- Şifreleme ve kodlama ilkeleri, siber güvenliğin temelini oluşturur ve bu olmadan modern dijital iletişim kötüye kullanıma karşı kolay bir hedef olur.

Bu konuyu tamamladığında...

- ✓ Bilgisayarın ne olduğunu ve temel bileşenlerinin neler olduğunu açıklayabilirsin,
- ✓ Donanım ve yazılımı ayırt edebilirsin,
- ✓ Günlük görevler için dijital cihazları kullanabilirsin.

**TEKRARLAMA SORULARI:**

1. Mors kodu neden bir kodlama biçimi olarak kabul edilir, ancak şifreleme olarak kabul edilmez?
2. Simetrik bir anahtar kullanarak bir mesajın nasıl şifreleneceğini ve şifresinin nasıl çözüleceğini açıklayan bir yönerge yaz!

**AŞAĞIDAKİ KONULARDA ŞUNLARI ÖĞRENECEKSİN:**

- Programlamanın ne olduğunu açıklayabilirsin,
- Belirli bir problemi çözme sürecini tanımlayabilir ve açıklayabilirsin.

Günlük hayatta bilgisayarlar, hesaplama, metin işleme, çizim, iletişim ve daha birçok görev için kullanılır. Ancak bilgisayar kendi başına bu görevleri nasıl gerçekleştireceğini bilemez. Yapması gereken her adım için açık ve kesin talimatlar alması gerekir. Programlamanın temelinde tam olarak bu vardır. Programlama, bilgisayar tarafından anlaşılabilen özel bir dilde – programlama dilinde – yazılmış bir dizi talimat oluşturma sürecidir. Bu talimatlar, bilgisayara tam olarak ne yapacağını, hangi sırayla ve hangi koşullar altında gerçekleştireceğini belirten program adı verilen yapılarda bir araya getirilir. Bu dersin amacı, öğrencilerin programlamanın temel özünü, kullanım alanlarını ve günümüz dijital dünyasında bu becerinin neden önemli olduğunu öğrenmelerini sağlamaktır.

Programlama, insanların – yani programcıların – bilgisayar için talimatlar oluşturduğu bir etkinliktir. Bu talimatlara veya komutlara kod adı verilir ve bunlar bir programlama diliyle yazılır. Programlama dili, insanın makineyle iletişim kurmasını sağlayan bir araçtır. Her dilin, doğal dillere benzer şekilde kendi kuralları, dil bilgisi ve yapısı vardır. Örneğin, “Işığы yak” cümlesi insan için açık bir anlam taşıırken, turnOn(light) komutu da anlaşılır bir

programlama dilinde yazıldığında bilgisayar için anlam ifade edebilir. Amaç, bilgisayarın ne yapmasını istediğimizi adım adım açıklamaktır.

ÖRNEK 1

Bir robota ev ödevini yapması için talimat vermen gerektiğini düşün. Eğer bulaşıkların yıkanmasını istiyorsan, talimatlar şöyle olabilir: "lavaboya git", "suyu aç", "deterjan koy", "bulaşığı yıka", "durula", "kurutmaya bırak". Benzer şekilde, programcı da bilgisayara açık adımlar verir – her adım doğru ve belirsizlik içermeyecek şekilde olmalıdır.

Bir program bir kez yazıldığında, kod doğru olduğu sürece bilgisayar aynı görevi her zaman aynı şekilde ve hatasız olarak gerçekleştirebilir. Bu durum, programı güçlü bir otomasyon aracı hâline getirir. Zaman alacak ve insan hatalarına açık olacak işlemler, makine tarafından hızlı ve doğru şekilde gerçekleştirilen görevlere dönüşür. Programlar yalnızca büyük ve karmaşık projelerde kullanılmaz, günlük hayattaki birçok alanda da bulunur: hesap makinesi, cep telefonu, ATM, buzdolabı veya akıllı saat – bunların hepsi insan tarafından yazılmış bir programa göre çalışır.

ÖRNEK 2

Bir hesap makinesinde " $5 + 3 =$ " yazıldığında, arka planda sayıları okuyan, "+" işlemini tanıyan, toplamı hesaplayan ve sonucu gösteren bir program çalışır. Tüm bu adımlar daha önceden program kodunda tanımlanmıştır.

Programcının yalnızca kod yazmadığını anlamak önemlidir, programcı aynı zamanda mantıklı düşünür, görevleri daha küçük bölümlere ayırır, tüm olası durumların dikkate alınıp alınmadığını kontrol eder ve kodun doğru çalıştığından emin olmak için test eder. Bir program beklenen şekilde çalışmadığında, "hata ayıklama" (debugging) adı verilen işlem yapılır, yani hatalar bulunur ve düzeltilir. Programlama, mantıksal düşünme, yaratıcılık ve disiplinin birleşimidir. Büyük matematik bilgisine sahip olmak zorunlu değildir, önemli olan problem çözme becerisidir.

ÖRNEK 3

Seyahat için gerekli süreyi hesaplayan bir uygulama yapmak için mesafe ve hız bilgileri girilir, program ise zaman = mesafe / hız işlemini yaparak süreyi hesaplar. Ancak programcı, bir kişinin hız = 0 girmesi durumunda ne olacağını önceden düşünmelidir. Bu durum hataya yol açar, bu nedenle iyi bir programcı ek bir kontrol ekler: "Eğer hız = 0 ise, şu mesajı göster: Geçerli bir değer giriniz."

```
function filterUsers(list, options = {}) {  
  function return list.filter(user => {  
    const byName = options.name ? user.name.toLowerCase().includes(options.name.toLowerCase()) : true;  
    const byEmail = options.email ? user.email.toLowerCase().includes(options.email.toLowerCase()) : true;  
    const byMinAge = options.minAge !== "number" ? user.age >= options.minAge : true;  
    const byMaxAge = options.maxAge !== "number" ? user.age <= options.maxAge : true;  
    return byName && byEmail && byMinAge && byMaxAge;  
  });  
}
```



SONUÇ

Programlama yalnızca kod yazmak değildir, bilgisayarın gerçekleştirmesi gereken adımları doğru ve mantıklı bir şekilde ifade etme sanatıdır. Programlama sayesinde bilgisayarlar, günlük hayattaki süreçleri otomatikleştirmek, hızlandırmak ve kolaylaştırmak için kullanılır. Programlama öğrenmek, problem çözme becerisi kazanmak, yeni şeyler oluşturmak ve her gün kullandığımız dijital araçların nasıl çalıştığını anlamak anlamına gelir.

TEKRARLAMA SORULARI



1. Programlama dili nedir?
2. Donanım ile program arasındaki fark nedir?
3. Programlama her zaman matematik gerektirir mi?
4. Bir programcının temel faaliyetleri nelerdir?
5. Hata ayıklama (debugging) nedir ve neden önemlidir?

Öğrendiklerini Hatırla!

- Programlama, bilgisayar için talimatlar yazma sürecidir.
- Talimatlar programlama dilleriyle yazılır.
- Programlar görevleri otomatikleştirmek için kullanılır.
- Programcı mantıklı düşünür ve programı test eder.
- Hatalar hata ayıklama (debugging) yoluyla giderilir.



DAHA FAZLA BİLGİ EDİNMEK İSTEYENLER İÇİN

- Günümüzde en popüler programlama dillerinin hangileri olduğunu araştır!
- Her adımın açık ve doğru olması gereken bir mantık bulmacasını çözmeyi dene!
- Scratch veya Code.org gibi ücretsiz çevrim içi programlama öğrenme platformlarından birini ziyaret et!



3.6

Programlama Dili ve Çevirici (Derleyici/Yorumlayıcı)

Bu konuyu tamamladığında...

- ✓ Donanım ve yazılımı ayırt edebilirsin,
- ✓ Programlamanın ne olduğunu açıklayabilirsin,
- ✓ Bir mantıksal işlemi talimatlar dizisi (sıralı komutlar) aracılığıyla açıklayabilirsin.



TEKRARLAMA SORULARI:

1. Kendi sözlerinle programlamanın ne olduğunu açıkla!
2. Hayattan, bir program ile otomatikleştirilebilecek bir görevi açıkla!
3. Program kullanan birkaç cihaz belirt!
4. Programda bir hata oluştuğunda programcı ne yapar?
5. Her talimatın açık ve doğru olması neden önemlidir?



AŞAĞIDAKİ KONULARDA ŞUNLARI ÖĞRENECEKSİN:

- Çeviricinin (translator) ne olduğunu açıklayabilirsin,
- Derleyicinin (compiler) ne olduğunu açıklayabilirsin,
- Çevirici ve derleyicinin farklı çalışma yöntemleri arasındaki farkı açıklayabilirsin.

İnsanlar bilgisayarlarla iletişim kurmak istediklerinde, makinenin anlayabileceği özel diller kullanılır. Programlama dilleri olarak adlandırılan bu diller, bilgisayarın çalıştırabileceği talimatların yazılmasını sağlayan, biçimsel ve mantıksal olarak düzenlenmiş bir ifade biçimidir. İnsanların günlük iletişimde kullandığı doğal dillerden farklı olarak, programlama dilleri kesin, yapılandırılmış ve belirsizlik içermeyen özelliklere sahip olmalıdır. Böyle bir dil olmadan programlama mümkün olmazdı, çünkü bilgisayar insan konuşmasını değil, yalnızca makine kodunu – yani ikili komutlardan oluşan dizileri – anlar.

Programlama dili, belirli bir görevi çözmek için gerçekleştirilmesi gereken mantığın ve adımların ifade edilmesini sağlar. Her dilin, kodun yazılma kurallarını belirleyen kendi sözdizimi (syntax) vardır. C++, Python, Java, JavaScript ve diğerleri gibi birçok programlama dili bulunmaktadır ve her birinin kendine özgü yazım tarzı ve kullanım alanları vardır. Bir dilin seçimi, çözülen göreve bağlıdır, bazı diller mobil uygulamalar için, bazıları bilimsel hesaplamalar için, bazıları ise web geliştirme için daha uygundur. Görünüşleri farklı olsa da hepsinin ortak amacı aynıdır: insan fikirlerini bilgisayarın çalıştırabileceği bir biçime dönüştürmek.

ÖRNEK 1



C++ dilinde metni görüntülemek için `cout << "Merhaba"`, komutu yazılır. Python'da ise aynı amaç `print("Zdravo")` şeklinde ifade edilir. Her iki örnek de aynı sonucu – metnin görüntülenmesini – sağlar, ancak her programlama dilinin sözdizimine uygun olarak farklı şekilde yazılır.

Programlama dili bilgisayar tarafından doğrudan anlaşılmaz. Yazılan kodun önce makine diline "çevrilmesi" gerekir. Bu görevi, iki tür olabilen çeviriciler yerine getirir: derleyiciler (compiler) ve yorumlayıcılar (interpreter). Derleyici, tüm kaynak kodu alır, analiz eder ve bağımsız olarak çalıştırılabilen bir programa dönüştürür. Yorumlayıcı ise kodu satır satır okur ve ayrı bir dosya oluşturmadan doğrudan çalıştırır.

ÖRNEK 2



C++ dilinde bir program yazıldığında, derleyici önce tüm kodu bir .exe çalıştırılabilir dosyaya dönüştürür. Daha sonra bu dosya bilgisayarda çalıştırılabilir. Python'da ise yorumlayıcı, yazılan kodu doğrudan okur ve hemen çalıştırır. Aradaki fark, bir gösterinin filme kaydedilmesi (derlenmiş) veya sahnede izleyicilerin karşısında canlı oynanması (yorumlanmış) arasındaki farka benzer.

Çevirici yalnızca kodu makine diline dönüştürmez, aynı zamanda sözdiziminin doğru olup olmadığını da kontrol eder. Kodda hatalar varsa – eksik noktalı virgül, parantez veya uygun olmayan bir değişken adı gibi – çevirici bunu bildirir ve işlemi devam ettirmez. Bu kontrol son derece önemlidir, çünkü yanlış veya güvenli olmayan talimatların çalıştırılmasını önler. Böylece programlama dili ve çevirici birlikte güvenli ve verimli bir yazılım geliştirme süreci oluşturur.

ÖRNEK 3



Bir programcı C++ dilinde `cout << "Merhaba"` komutunu noktalı virgül olmadan yazarsa, derleyici şu hatayı verir: "expected ';' before end of statement". Böylece programcı, program çalıştırılmadan önce hatanın nerede olduğunu hemen öğrenir ve gerekli düzeltmeyi yapabilir.

DAHA FAZLA BİLGİ EDİNMEK İSTEYENLER İÇİN

- C++ için g++ derleyicisinin veya Python yorumlayıcısının nasıl çalıştığını araştır!
- Kısa bir Python kodu yazmayı dene ve sözdizimi hatası yaptığında ne olduğunu gözlemler!
- <https://www.learnpython.org/> sayfasını ziyaret et ve basit örnekleri dene!



SONUÇ

Programlama dilleri ve çeviriciler, programlama sürecindeki temel araçlardır. Bunlar olmadan bilgisayarlar için komutlar yazmak, çevirmek ve çalıştırmak mümkün olmazdı. İnsan ile makine arasında doğru, güvenilir ve verimli bir iletişim kurulmasını sağlarlar. Nasıl çalıştıklarını anlamak, yazılım ve dijital çözümlerin geliştirilmesiyle ilgili gelecekte edinilecek tüm bilgilerin temelini oluşturur.

BİLGİ KONTROLÜ SORULARI VE ETKİNLİKLER



1. Derleyici ve yorumlayıcının çalışma yöntemlerini karşılaştırın!
2. Programlama dilinde sözdizimi (syntax) neden önemlidir?
3. Bir programlama dili tüm görevler için kullanılabilir mi?
4. Programın çalıştırılma sürecinde çeviricinin rolü nedir?
5. Kodda bir hata varsa ne olur?

Öğrendiklerini Hatırla!

- Programlama dilleri, bilgisayara verilecek talimatları yazmak için kullanılır.
- Derleyici, programın tamamını makine koduna dönüştürür.
- Yorumlayıcı, programı satır satır çalıştırır.
- Sözdizimi (syntax), uyulması gereken kurallar bütünüdür.
- Çevirici, kodda hata varsa uyarı verir.



3.7

Programlama Dilleri Arasındaki Farklar: Scratch, C++, Java, Python, PHP, Lisp

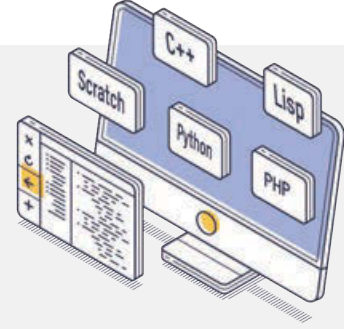
Bu konuyu tamamladığında...

- ✓ Programlama dilinin ne olduğunu açıklayabilirsin,
- ✓ Derleyici (compiler) ve yorumlayıcı (interpreter) arasındaki farkı ayırt edebilirsin,
- ✓ Sözdizimi (syntax) hatalarını tanıyabilirsin.



TEKRARLAMA SORULARI:

1. Programlama dili nedir?
2. Hangi tür çeviriciler vardır?
3. Derleyicinin (compiler) çalışma prensibi nedir?
4. Yorumlayıcının (interpreter) çalışma prensibi nedir?
5. Sözdizimi (syntax) hatası nedir?



AŞAĞIDAKİ KONULARDA ŞUNLARI ÖĞRENECEKSİN:

- Farklı programlama dilleri arasındaki farkın ne olduğunu açıklayabilirsin,
- Farklı programlama dillerinin çalışma biçimlerini ayırt edebilirsin.

Programlama dünyasında onlarca, hatta yüzlerce programlama dili bulunmaktadır ve her birinin kendine özgü avantajları, kuralları, kullanım alanları ve hedef kitlesi vardır. Her dil belirli bir amaçla oluşturulmuştur: bazıları eğitim için, bazıları sistem yazılımları geliştirmek için, bazıları web uygulamaları için, bazıları ise yapay zekâ alanı için daha uygundur. Her ne kadar hepsi bilgisayarın çalıştıracağı talimatları yazmak için kullanılsa da bu talimatların yazılma, okunma ve çalıştırılma biçimleri büyük ölçüde farklılık gösterebilir. Bu nedenle öğrencinin birkaç farklı programlama diliyle tanışması, programlamanın olanaklarını daha geniş bir şekilde anlaması açısından önemlidir.

Yeni başlayanların genellikle ilk tanıştığı dil Scratch'tir. Bilindiği gibi Scratch, MIT tarafından geliştirilen görsel bir programlama dilidir. Kod yazmak yerine, yapboz parçaları gibi birbirine bağlanan grafik bloklar kullanılır. Bu yaklaşım sözdizimi hatalarını ortadan kaldırır ve öğrencilerin programın mantığına, işlem sırasına ve koşullarına odaklanmasını sağlar. Scratch, oyunlar, animasyonlar ve basit simülasyonlar oluşturmak için kullanılır ve "gerçek" kodun korkutucu görünümü olmadan etkileşimli bir öğrenme yöntemi sunar.

ÖRNEK 1



Scratch'te, eğer karakterin 10 adım ileri hareket etmesini istiyorsanız, basitçe "10 adım ileri" bloğu seçilir ve diğer bloklarla birleştirilir. Metin tabanlı kod yazmaya veya noktalı virgül kullanımıyla ilgilenmeye gerek yoktur.

Bir sonraki önemli dil C++'tır. C++, günümüzde hâlâ kullanılan en eski ve en güçlü programlama dillerinden biridir. Bu dil derlenen (compiled) bir dildir, yani tüm program önce makine koduna çevrilir ve ancak bundan sonra çalıştırılır. C++, hızı, bellek üzerindeki kontrolü ve düşük seviyeli işlemleri gerçekleştirme yeteneğiyle bilinir. İşletim sistemleri, gömülü sistemler ve hatta video oyunları geliştirmede kullanılır. Ancak karmaşık sözdizimi nedeniyle tamamen yeni başlayanlar için ilk dil olarak önerilmez, fakat ileri düzey öğrenciler için oldukça uygundur. C++, birçok sosyal ağın ve kripto para sistemlerinin temelinde kullanılan teknolojilerden biridir. Hızı ve kararlılığı, onu birçok farklı yazılım türü için ideal bir programlama dili hâline getiri.

ÖRNEK 2



C++'ta metin görüntülemek için `cout << "merhaba"`, kodu kullanılır, ancak bundan önce başlık dosyası olarak `#include <iostream>` eklenmeli ve giriş noktası olan `main()` fonksiyonu yazılmalıdır. Bu durum, Scratch'e göre daha yüksek bir başlangıç seviyesi gerektirir.

Java ve Python, geniş kullanım alanına sahip iki modern programlama dilidir. Java, nesne yönelimli ve oldukça yapılandırılmış bir dildir, mobil uygulamalarda (özellikle Android), masaüstü programlarında ve web uygulamalarında kullanılır. Python ise betik (script) dili olarak bilinir ve basit sözdizimi ile okunabilirliği sayesinde çok popülerdir. Makine öğrenmesi, veri işleme, bilimsel çalışmalar ve otomasyon alanlarında kullanılır. Birçok yeni başlayan kişi, kolaylığı nedeniyle programlamaya Python ile başlar.

ÖRNEK 3



Python'da metin görüntülemek için `print("Merhaba")` yazmak yeterliyken, Java'da aynı sonucu elde etmek için daha fazla kod satırı ve nesne kullanımı gerekir. Bu durum, Python'u programlamaya ilk başlayan kişiler için daha sezgisel hâle getirir.

PHP ve Lisp ise uzmanlaşmış programlama dilleridir. PHP, özellikle dinamik web sayfalarında sunucu taraflı web programlama için en çok kullanılan dillerden biridir. HTML kodunun içine eklenebilir ve veri tabanlarıyla etkileşim kurmayı ve formları işlemeyi sağlar. Lisp ise en eski programlama dillerinden biridir ve yapay zekâ ile fonksiyonel programlama alanlarındaki kullanımıyla bilinir. Başlangıç seviyesinde nadiren kullanılsa da Lisp, programlamaya farklı bir bakış açısı kazandırır, çünkü özyineleme (recursion) ve sembolik veri işleme yöntemlerini kullanır.

ÖRNEK 4



PHP’de metin görüntülemek istiyorsanız, `<?php echo "Merhaba!"; ?>` şeklinde yazılır, bu, programlama dilinin HTML’e nasıl yerleştirildiğini gösterir. Lisp’te (+ 1 2) ifadesi toplama anlamına gelir – operatör başta yer alır, bu da çoğu programlama dilinin tersidir.



SONUÇ

Programlama dilleri arasındaki farkları anlamak, öğrencilerin hangi dille devam edecekleri konusunda bilinçli bir seçim yapmalarını sağlar. Bu dillerin her birinin kendine özgü avantajları, dezavantajları ve kullanım alanları vardır. Görsel sezgi için Scratch’ten, sistem kontrolü için C++’a, uygulamalar için Java ve Python’dan, web ve yapay zekâ için PHP ve Lisp’e kadar seçenekler geniş ve keşfedilmeye değerdir.

BİLGİ KONTROLÜ SORULARI VE ETKİNLİKLER



1. Scratch ve Python’u kod yazma yöntemleri açısından karşılaştırınız!
2. PHP neden sıklıkla web geliştirmede kullanılır?
3. Java ve C++ programlama dillerinin benzer özellikleri nelerdir?
4. Yapay zekâ uygulamaları için hangi programlama dili daha uygundur?
5. Derlenen (compiled) ve yorumlanan (interpreted) bir dil arasındaki farkı bir örnekle açıklayınız!

Öğrendiklerini Hatırla!

- Scratch, görsel bir programlama dilidir ve başlangıç seviyesindeki kullanıcılar için uygundur.
- C++, derlenen ve ileri düzey görevler için güçlü bir programlama dilidir.
- Python, basit bir sözdizimine ve geniş kullanım alanına sahiptir.
- Java, nesne yönelimli ve kararlı bir programlama dilidir.
- PHP, web geliştirme için, Lisp ise yapay zekâ alanı için kullanılır.



DAHA FAZLA BİLGİ EDİNMEK İSTEYENLER İÇİN

- Scratch’i yükleyin ve bir animasyon yapmayı deneyin!
- Python’da kısabir program yazın ve aynı görevi Java’da nasıl yapıldığını karşılaştırın!
- PHP’nin ziyaret ettiğiniz web sitelerinde nasıl kullanıldığını araştırın!
- Lisp’i çevrim içi bir derleyicide deneyin ve sözdizimini analiz edin!



3.8

Entegre Programlama Ortamı

Bu konuyu tamamladığında...

- ✓ IDE'nin ne olduğunu açıklayabilirsin,
- ✓ Derleme (compile) ile yorumlama (interpret) arasındaki farkı ayırt edebilirsin,
- ✓ Sözdizimi (syntax) hatasını tanıyabilirsin.



TEKRARLAMA SORULARI:

1. Görsel programlama dili nedir?
2. C++ en çok hangi amaçlarla kullanılır?
3. Başlangıç seviyesindeki kullanıcılar için en uygun dil hangisidir?
4. Python'un avantajları nelerdir?



AŞAĞIDAKİ KONULARDA ŞUNLARI ÖĞRENECEKSİN:

- IDE'nin ne olduğunu açıklayabilirsin,
- Farklı geliştirme ortamlarını tanıyabilirsin.

Modern programlamada kod yazma, test etme ve çalıştırma süreci boş bir konsolda veya komut satırında manuel olarak gerçekleştirilmez. Bunun yerine, tüm geliştirme sürecini kolaylaştıran özel araçlar kullanılır. Bu araçlar entegre geliştirme ortamı (IDE) adıyla bilinir. Bir IDE, etkili programlama için gerekli tüm bileşenleri içeren kapsamlı bir yazılım paketidir: kod düzenleyici, derleyici, hata ayıklayıcı ve çalıştırma konsolu. Bu bütünleşik ortam, hem başlangıç seviyesindeki kullanıcılar hem de profesyonel programcılar tarafından kullanılır, çünkü yapılandırılmış bir çalışma yöntemi ve geliştirme süreci üzerinde görsel kontrol sağlar.

Entegre geliştirme ortamı, programcının "evi" olarak kabul edilir, programların yazıldığı, kontrol edildiği ve çalıştırıldığı yerdir. Basit bir metin düzenleyiciden farklı olarak

IDE, kodun otomatik tamamlanması, sözdizimi hatalarının vurgulanması, tür kontrolü ve derleme araçlarına hızlı erişim gibi özellikler sunar. Bu özellikler yalnızca geliştirme sürecini hızlandırmakla kalmaz, aynı zamanda verimliliği de artırır. IDE sayesinde öğrenci, kodun herhangi bir bölümünün doğru yazılıp yazılmadığını hemen görebilir ve programı tek bir düğmeye basarak test edebilir.

ÖRNEK 1



Visual Studio Code’da cout yazılırken IDE, ifadenin geri kalanını otomatik olarak önerir, << ekler ve endl seçeneğini sunar. Bu durum hata yapma riskini azaltır ve yazma hızını artırır.

Bir IDE içerisinde kodun doğrudan derlenmesi ve çalıştırılması da mümkündür. Öğrencinin komut satırını manuel olarak açmasına, dosyayı aramasına ve ayrı bir komut aracılığıyla çalıştırmasına gerek yoktur. Tek bir tıklamayla tüm program derlenir ve çalıştırılır. Ayrıca IDE, aynı anda birden fazla dosyanın düzenlenmesini, proje yapısının takip edilmesini ve kod bölümleri arasındaki bağımlılıkların yönetilmesini sağlar.

ÖRNEK 2



Code::Blocks’ta program yazıldıktan sonra “Build and Run” düğmesine basılır ve sonuç hemen alt pencerede görüntülenir. Hataları dışarıdan aramaya gerek kalmaz.

Entegre geliştirme ortamının bir diğer önemli bileşeni ise hata ayıklayıcı DIR(debugger). Bu araç, kodun çalıştırılmasını adım adım takip etmeyi sağlar. Bu özellik sayesinde öğrenci, “breakpoint” adı verilen bir durdurma noktası belirleyebilir ve programın o anda değişkenlerde hangi değerlere sahip olduğunu görebilir. Bu özellik, programın mantığını anlamak ve sözde mantıksal hataları çözmek açısından büyük önem taşır.

ÖRNEK 3



Eğer bir programda x değişkeninin değerinin 5 olması gerekirken IDE, bu değer 3 olduğunu gösteriyorsa, hata ayıklayıcı (debugger) aracılığıyla hatanın meydana geldiği satır bulunur ve düzeltme önerilir.

Modern IDE’ler ayrıca donanım cihazları için simülatörler, yerleşik terminaller veya Git gibi sürüm kontrol sistemleriyle entegrasyon gibi ek modüller de sunar. Eğitim ortamında bazı IDE’ler, başlangıç seviyesindeki kullanıcılar için daha basit bir görünüm ve sınırlı özelliklerle uyarlanmıştır. Örneğin Python için Thonny veya mikrodenetleyiciler için Arduino IDE kullanılabilir. Bu tür ortamlar, işlevsellik ve basitlik arasında başlangıç seviyesine uygun bir denge sağlayarak görsel araçlardan gerçek koda geçişi kolaylaştırır.



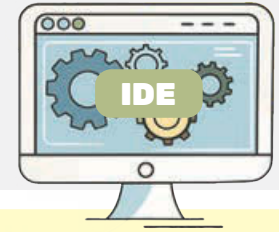
SONUÇ

Entegre geliştirme ortamı (IDE), programların yazılması ve test edilmesi için merkezi bir çalışma alanıdır. Başlangıç seviyesindeki bir proje veya profesyonel bir uygulama olması fark etmeksizin, IDE kullanımı çalışmayı büyük ölçüde kolaylaştırır, verimliliği artırır ve hata olasılığını azaltır. IDE'nin bileşenlerini ve işlevlerini bilmek, gerçek programlamaya yönelik atılan ilk adımdır.



TEKRARLAMA SORULARI:

1. IDE nedir?
2. Bir IDE'nin üç ana bileşeni nelerdir?
3. Hata ayıklayıcı (debugger) ne için kullanılır?
4. IDE olmadan bir program çalıştırmak mümkün müdür?
5. Python için hangi IDE'ler kullanılır?



BİLGİ KONTROLÜ SORULARI VE ETKİNLİKLER



1. IDE ile sıradan bir metin düzenleyici arasındaki farkı açıklayınız!
2. IDE'de otomatik tamamlamanın rolü nedir?
3. Hata ayıklayıcı, hataların bulunmasına nasıl yardımcı olur?
4. Breakpoint nedir ve ne zaman kullanılır?
5. Başlangıç seviyesindeki kullanıcılar için en uygun IDE hangisidir ve neden?

Öğrendiklerini Hatırla!

- IDE, kod yazma, kontrol etme ve çalıştırma işlemleri için entegre bir ortamdır.
- Kod düzenleyici, derleyici ve hata ayıklayıcı içerir.
- Hataların bulunmasına ve giderilmesine yardımcı olur.
- Sonuçların hızlı ve görsel olarak incelenmesini sağlar.
- Başlangıç seviyesindeki kullanıcılar ve profesyoneller için uygundur.



DAHA FAZLA BİLGİ EDİNMEK İSTEYENLER İÇİN

- Visual Studio Code'u yükleyin ve Python'da bir program yazın!
- Hata ayıklama özelliğini deneyin ve breakpoint kullanın!
- Git entegrasyonunun IDE içinde nasıl çalıştığını araştırın!
- İki IDE'yi karşılaştırın ve farklılıklarını yazın!



Bu konuyu tamamladığında...

- ✓ Entegre geliştirme ortamının (IDE) ne olduğunu açıklayabilirsin,
- ✓ Temel programlama dillerini ayırt edebilirsin,
- ✓ IDE örnekleri verebilirsin

**TEKRARLAMA SORULARI:**

1. IDE ne işe yarar?
2. C++ için bildiğin geliştirme ortamları nelerdir?

**AŞAĞIDAKİ KONULARDA ŞUNLARI ÖĞRENECEKSİN:**

- Hata ayıklamanın (debugging) ne olduğunu açıklayabilirsin,
- Derleme (compile) işlemini gerçekleştirebilirsin.

Bir program oluşturma süreci yazılan kodla sona ermez. Aksine, kod yazma işlemi yazılım geliştirme sürecinin yalnızca ilk aşamasıdır. Programın derlenmesi, çalıştırılması ve beklendiği gibi çalışıp çalışmadığının kontrol edilmesi gerekir. Genellikle kodun ilk sürümünde sözdizimi hataları, mantıksal hatalar veya çalışma zamanı hataları bulunur. Bu nedenle programlamada önemli bir aşama olan hata ayıklama (debugging), koddaki hataların bulunması ve giderilmesi sürecidir. Bu sürecin bütününe anlamak, programlama öğrenen herkes için önemlidir, çünkü teorik bilgileri pratik uygulamaya dönüştürmeyi sağlar.

Program yazma işlemi, IDE'nin bir parçası olan kod düzenleyicinin kullanılmasıyla başlar. Seçilen programlama dilinde komutlar yazılır, sözdizimine, elemanların düzenine ve doğru mantığa dikkat edilir. Kodun okunabilir olması, boş satırlarla düzenlenmesi ve mantıksal bölümleri açıklayan yorum satırlarının eklenmesi gerekir. Doğru yazılmış bir kod, hata olasılığını azaltır ve daha sonraki analiz işlemlerini kolaylaştırır.

ÖRNEK 1



C++ programı yazarken bir satırın sonunda , işaretini eklemeyi unutursak, derleme sırasında bir sözdizimi hatası alırız. Code::Blocks gibi IDE'ler hatalı satırı otomatik olarak vurgular ve şu tür bir mesaj görüntüler: "expected ',' before 'return'".

Yazma işleminden sonra kodun derlenmesi gerekir. Bu, kaynak kodun makinenin anlayabileceği koda dönüştürülmesi anlamına gelir. Bu süreç, yanlış yazılmış anahtar kelimeler, unutulmuş parantezler veya hatalı ifadeler gibi sözdizimi hatalarını kontrol eder. Hatalar varsa derleyici, hatanın bulunduğu satırı ve hata türünü belirten mesajlar gösterir. Öğrenciler bu mesajları okumayı ve yorumlamayı öğrenir, bu, her programcının günlük çalışmalarının bir parçasıdır.

ÖRNEK 2



Eğer bir komutta tam sayı (int) beklenen bir yere metin yerleştirilirse, derleyici hata verir, çünkü bir metin, tam sayı türündeki bir değişkene atanamaz. IDE şu hata mesajını gösterir: `"invalid conversion from 'const char' to 'int'***"`.

Derleme işlemi başarıyla tamamlandıktan sonra programın çalıştırılması aşamasına geçilir. Bu aşamada kod yürütülür ve sonuç ekranda gösterilir. Program beklenildiği gibi davranmıyorsa, mantıksal bir hata bulunabilir. Bu tür hatalarda program sorunsuz bir şekilde çalışır, ancak doğru sonucu üretmez. Mantıksal hatalar, derleyici tarafından tespit edilmediği için bulunması en zor hata türleridir.

ÖRNEK 3



Eğer programın bir çarpma işlemi yapması gerekirken kodda * yerine + kullanılmışsa, sonuç yanlış olacaktır ve herhangi bir uyarı verilmeyecektir. Program çalışacaktır, ancak istenen işlemi yerine getirmeyecektir.

Hata ayıklama (debugging), bu tür mantıksal veya gizli hataların bulunması ve analiz edilmesi sürecidir. Bu süreçte breakpoint (kesme noktası) gibi araçlar kullanılır. Breakpoint, programın çalışmasının belirli bir satırda duraklatılmasını sağlar. Ayrıca adım adım yürütme (step-by-step) özelliği kullanılarak değişkenlerin durumu takip edilir. IDE'ler, değişkenlerin değerlerini görsel olarak göstermeye ve kod satırları arasında kolayca ilerleyerek denetim yapmaya olanak tanır. Bu teknik, öğrencilerin kendi kodlarında tam olarak neler gerçekleştiğini anlamalarına yardımcı olur.

```
function filterUsers(list, options = {}) {  
  function return list.filter(user => {  
    return (options.name ? user.name.toLowerCase().includes(options.name.toLowerCase()) : true;  
    options.email ? user.email.toLowerCase().includes(options.email.toLowerCase()) : tr  
    of options.minAge !== "number" ? user.age >= options.minAge : true; })  
  }  
}
```



SONUÇ

Program yazma süreci kodun yazılmasıyla sona ermez. Her program dikkatli bir şekilde derlenmeli, test edilmeli ve gerektiğinde hata ayıklama işleminden geçirilmelidir. Hatalar bir başarısızlık olarak değil, sürecin doğal bir parçası olarak görülmelidir. Düzeltile her hata, yeni bir öğrenme fırsatı sunar. Hata ayıklama becerisini kazanmak, programlamadaki en önemli becerilerden birini kazanmak anlamına gelir.

BİLGİ KONTROLÜ SORULARI VE ETKİNLİKLER



1. Bir sözdizimi (syntax) hatasına örnek veriniz!
2. Derleyicinin verdiği hata mesajı neyi ifade eder?
3. Hata ayıklama işleminin adım adım nasıl yapıldığını açıklayınız!
4. Hata ayıklama için hangi IDE araçları kullanılır?
5. Bazı hatalar neden derleme sırasında tespit edilemez?

Öğrendiklerini Hatırla!

- Kod yazarken sözdizimine ve mantığa dikkat edilmelidir.
- Hatalar, sürecin doğal bir parçasıdır.
- Derleyici, sözdizimini kontrol eder.
- Programın çalıştırılması, sonucun doğru olup olmadığını gösterir.
- Hata ayıklama (debugging), hataların bulunması ve giderilmesi sürecidir.



DAHA FAZLA BİLGİ EDİNMEK İSTEYENLER İÇİN

- Kasıtlı olarak mantıksal bir hata içeren kısa bir program yazın ve hatayı bulmaya çalışın!
- Visual Studio Code'da hata ayıklamanın (debugging) nasıl çalıştığını araştırın!
- Breakpoint'in nasıl ayarlanacağını öğrenin!
- Hata mesajlarını İngilizceden Makedoncaya çevirin ve ne anlama geldiklerini açıklayın!



Bu konuyu tamamladığında...

- ✓ Entegre geliştirme ortamının (IDE) temel bileşenlerini tanıyabiliyorsun,
- ✓ Kod düzenleyiciyi, derleyiciyi ve çalıştırma penceresini birbirinden ayırt edebiliyorsun,
- ✓ Ekranda bir sonucun görüntülenmesinin ne anlama geldiğini biliyorsun.

**TEKRARLAMA SORULARI:**

1. Bir programı derlemek ne anlama gelir?
2. Programlamadaki başlıca hata türleri nelerdir?
3. Hata ayıklama neden önemlidir?
4. Breakpoint nasıl kullanılır?
5. Mantıksal hata ne anlama gelir?

**AŞAĞIDAKİ KONULARDA ŞUNLARI ÖĞRENECEKSİN:**

- Programlamada ifadenin (statement) ne olduğunu açıklayabilirsin, ekran görüntüleme ifadelerini diğer komutlardan ayırt edebilirsin,
- Programların kullanıcı ile nasıl iletişim kurduğunu anlayabilirsin.

Programlama dünyasında, bilgisayara verdiğimiz her komut ifade (statement) olarak adlandırılır. İfadeler, her programın temel yapı taşlarıdır, bilgisayara tam olarak ne yapması gerektiğini bildirir. Bu ifadeler aracılığıyla hesaplama yapma, değer saklama veya – bu ders için en önemli konu olan – ekranda mesaj ve sonuç görüntüleme gibi çeşitli işlemler gerçekleştirilir. Bilginin ekranda gösterilmesi, program ile kullanıcı arasındaki ilk etkileşim biçimidir. Bir program yazarken, yalnızca çalışması değil, aynı zamanda ne yaptığını

gösterebilmesi de önemlidir. Bu nedenle, ekrana çıktı verme ifadelerinin öğrenilmesi, öğrencilerin mantıksal ve iletişimsel düşünme becerilerini geliştirmelerinde önemli bir adımdır.

İnsanla iletişim kuran her programda bilgi gösterme ihtiyacı vardır. Bu bilgi, bir hesaplama sonucu, bir yönerge, bir uyarı veya başarı mesajı olabilir. Bunun için ekrana mesaj göndermeyi sağlayan özel ifade türleri kullanılır. Bu ifadeler sayesinde program, kullanıcı için anlaşılır ve faydalı hâle gelir. Böyle bir ifade çalıştırıldığında ekranda belirtilen şey tam olarak görünür, metin, sayı veya bunların birleşimi. Bu görüntüleme, programda bir işlemin gerçekleştiğini gösteren görsel bir kanıttır. Örneğin, bir matematik işleminin sonucunu gösterebilir veya basit bir karşılama mesajı gönderebiliriz.

Sadece bağımsız mesajlar değil, daha önce hesaplanmış veya girilmiş değerler de sıklıkla görüntülenir. Bu durumda program yalnızca istenen hesaplama adımını gerçekleştirmekle kalmaz, aynı zamanda sonucu kullanıcıya bildirir. Bu, kontrol açısından son derece önemlidir, çünkü kullanıcı görüntülenen sonucu inceleyerek programın doğru çalışıp çalışmadığını değerlendirebilir. Böylece bir etkileşim oluşur, kullanıcı bir veri girer, program bunu işler ve ardından sonucu gösterir. Bu son aşama olmadan program “sessiz” görünür ve içinde neler olduğunu gösterme imkânı olmaz.

Programdaki ifadelerin sıralamasının, mesajların ekranda gösterilme sırasını belirlediğini anlamak çok önemlidir. Programlar, aksi belirtilmediği sürece yukarıdan aşağıya ve sırayla çalıştırılır. Bu nedenle önce bir şeyin, ardından başka bir şeyin gösterilmesini istiyorsak, ifadelerin doğru sırada yerleştirilmesi gerekir. Sıralama değiştirilirse ekrandaki sonuç da farklı olur. Bu şekilde öğrenciler programın nasıl “düşündüğünü” öğrenir, programın mantığı kesin ve düzenli bir şekilde takip edilir ve her değişiklik görünen çıktıda sonuç oluşturur.

Bazen mesajların ayrı satırlarda, alt alta gösterilmesi istenir. Bunun için yeni satırı belirleyen mekanizmalar kullanılır. Bu mekanizmalar sayesinde ekrandaki metnin görünümü kontrol edilebilir: tek satırda mı yoksa birden fazla satıra yayılmış şekilde mi gösterileceği, boşluk bırakılıp bırakılmayacağı veya tüm bilgilerin birlikte görüntülenip görüntülenmeyeceği belirlenebilir. Bu, bilgilerin görsel olarak düzenlenmesi için faydalıdır, böylece çıktı düzenli, okunabilir ve kullanıcı için daha anlaşılır olur. Bu tür ayrıntılar sayesinde programlamanın yalnızca mantık değil, aynı zamanda sonucu gören insanla iletişim kurma süreci olduğu öğrenilir.

Ekran görüntüleme ifadeleri, hemen görünür sonuç verdikleri için öğrenilen ilk konular arasındadır. Bir mesaj ekranda görüldüğünde öğrenci programın çalıştığını anlar ve başarı hissi kazanır. Bu ifadeler sayesinde program akışının dinamik yapısı anlaşılmaya başlanır, çünkü gösterilen mesajlar ve sonuçlar, “arka planda” gerçekleşen işlemlerin doğrudan yansımasıdır. Ayrıca geri bildirim sağlarlar, eğer bir şey doğru görüntülenmiyorsa bu durum mantıksal bir hataya işaret edebilir. Bu nedenle bu ifadeler yalnızca görüntüleme aracı değil, aynı zamanda programı kontrol etme ve geliştirme aracıdır.

BİLGİ KONTROLÜ SORULARI VE ETKİNLİKLER



1. Programlamada bir ifade (statement) nedir?
2. Ekran görüntüleme ifadesi neden önemlidir?
3. Kullanıcı programın ne yaptığını nasıl öğrenir?
4. İfadelerin sırası değiştirilirse ne olur?
5. Doğru sırayla gösterilmesi gereken iki mesajdan oluşan bir senaryo oluşturunuz!
6. Tüm mesajlar aynı satırda gösterilse çıktı nasıl görünür? Açıklayınız!
7. Ekran görüntülemenin neden her algoritmanın önemli bir parçası olduğunu açıklayınız!

Öğrendiklerini Hatırla!

- İfadeler, programın sırayla gerçekleştirdiği komutlardır.
- Görüntüleme ifadesi, kullanıcı ile iletişim kurulmasını sağlar.
- Program, sonuçları veya mesajları ekran aracılığıyla gösterir.
- İfadelerin sıralanması önemlidir, çıktı bunun sonucuna bağlıdır.
- Çıktının görsel olarak düzenlenmesi, programda neler olduğunu anlamaya yardımcı olur.

DAHA FAZLA BİLGİ EDİNMEK İSTEYENLER İÇİN

- Mobil uygulamalarda mesajların nasıl görüntülendiğini araştırın!
- Bir menü gösteren programı hayal edin ve görüntüleme sırasının nasıl olacağını açıklayın!
- Gösterilen mesajın önceki bir değere bağlı olduğu bir senaryo yazın!
- Birden fazla gösterilen soru ve sonuç içeren bir testin nasıl görüneceğini düşünün!

3.11

Sözde Dil (Psödo Kod)

Bu konuyu tamamladığında...

- ✓ Sayısal ve metinsel değerleri ayırt edebilirsin,
- ✓ Basit matematiksel işlemleri gerçekleştirirsin, veri girişinin ve çıkışının ne anlama geldiğini tanırırsın, bir görevde basit mantıksal adımlarla çalışırsın, değişkenlerin verileri saklamak için kullanıldığını bilirsin.



TEKRARLAMA SORULARI:

1. Görüntülenen bilginin düzenli olması neden önemlidir?
2. Bir karşılama mesajı gösteren bir program hayal et! Her adımın ne yaptığını açıkla!
3. Bir işlemin sonucunun gösterilmesi gereken bir durum oluştur! Mantığını açıkla!



AŞAĞIDAKİ KONULARDA ŞUNLARI ÖĞRENECEKSİN:

- Geçerli bir sözde kod (psödo kod) komutunu tanıyabilir ve yazabilirsin, değişkenleri tanımlayabilir ve değer atayabilirsin, "gir" ve "göster" komutlarıyla veri girişi ve çıkışı gerçekleştirebilirsin, sözde koddaki aritmetik işlemleri uygulayabilirsin, birkaç adımdan oluşan tam bir algoritma yazabilirsin.

Psödo dil, gerçek kodlamadan önce kullanılan, programlamanın basitleştirilmiş ve açıklayıcı bir biçimidir. Bunun yardımıyla, sözdizimi ayrıntılarına takılmadan programlama mantığı öğrenilir. Doğrudan bilgisayarda çalıştırılmaz, bunun yerine bir programın izleyeceği adımları planlamak ve açıklamak için kullanılır. Özellikle programlamaya ilk kez başlayan öğrenciler için oldukça yararlıdır. Psödo dil sayesinde öğrenciler, ekranın arkasındaki kodun nasıl çalıştığı hakkında fikir edinirler: verilerin nasıl alındığı, nasıl işlendiği ve sonucun nasıl elde edildiği anlaşılır.

Değer Atama:

Her programda kullanılan ilk ve en temel işlem, değer atamadır. Psödo dilde bu işlem, ← oku ile yapılır. Bu işaret, sağ taraftaki değerın sol taraftaki değişkene yerleştirileceğini gösterir. Örneğin:

tam sayı değişken $a \leftarrow 10$

Bu, a adında bir değişken oluşturulduğu, bu değişkenin bir tam sayı sakladığı ve başlangıç değerinin 10 olduğu anlamına gelir. Bu komut hem başlangıç değeri vermek hem de mevcut değeri güncellemek için kullanılabilir. Böylece, daha sonraki tüm hesaplamaların temeli olan veri saklama kavramı öğrenilir.

Veri Girişi:

Programın kullanıcıdan bilgi alabilmesi gerekir. Bunun için klavyeden değer girmeyi sağlayan "gir" komutu kullanılır. Böylece program etkileşimli hâle gelir. Örneğin:

sayıyı girin

Bu, kullanıcının bir değer gireceği ve bu değer daha sonra sayı değişkenine kaydedileceği anlamına gelir. Giriş verileri olmadan program yalnızca önceden belirlenmiş değerlerle çalışır ve kullanım alanı sınırlanır. "Gir" komutu, algoritmaların dinamik davranış göstermesine yönelik bir adımdır.

Veri Çıkışı:

Bir diğer önemli unsur, verilerin gösterilmesidir. "Göster" komutu ile program kullanıcıya bir mesaj veya sonuç gönderebilir. Örneğin:

sayı "değerini" göster

Bu komut, "Değer: " metni ile birlikte sayı değişkeninin o andaki değerini ekrana yazdırır. Çıktı, hesaplama sonuçlarının görülmesi veya programın doğru çalıştığının kontrol edilmesi için gereklidir. Çıktı olmadan kullanıcı programın yaptığı işlemler hakkında bilgi sahibi olamaz.

Aritmetik İşlemler:

Değerler atandıktan ve veriler girildikten sonra hesaplamalar yapılabilir. Psödo dilde $+$, $-$, $*$, $/$ gibi temel aritmetik işlemler desteklenir. Toplama işlemine örnek:

tam sayı değişken toplam $\leftarrow a + b$

Bu, a ve b değerlerinin toplandığı ve sonucun toplam değişkeninde saklandığı anlamına gelir. Aritmetik işlemler, matematik problemleri, not hesaplamaları, sayma işlemleri, istatistik ve daha birçok alanda kullanılan temel işlemlerdir. Bu işlemler sayesinde programlar bilgileri işleyebilir.

Giriş, çıkış, değer atama ve aritmetik işlemlerin birleştirilmesiyle işlevsel bir algoritma elde edilir. Örneğin:

göster "İki sayı giriniz:"

gir x, y

tam sayı değişken çarpım $\leftarrow x * y$

göster "Çarpım: ", çarpım

Bu algoritma, kullanıcının iki sayı girmesini sağlar, ardından program bu sayıları çarpar ve sonucu ekranda gösterir. Bu tür örneklerle, algoritmik düşünme yoluyla problem çözmenin tam yapısı öğrenilir. Bu bilgiler, daha sonra öğrenilecek koşullar, döngüler ve fonksiyonlar gibi daha karmaşık konuların temelini oluşturur.

BİLGİ KONTROLÜ SORULARI VE ETKİNLİKLER



1. Psödo dilde $\leftarrow$ sembolü ne anlama gelir?
2. Klavyeden değer girmek için hangi kelime kullanılır?
3. "Göster" komutunun ne yaptığını açıklayınız!
4. Girilen bir sayının karesini gösterecek bir algoritma yazınız!
5. Kullanıcıdan üç sayı girmesini isteyin ve toplamalarını gösterin!
6. $(x + y) / 2$ değerini hesaplayacak bir algoritma oluşturunuz!
7. Kullanıcı tarafından girilen isim ile "Hoş geldin, [isim]" mesajını gösterecek bir algoritma yazınız.

Öğrendiklerini Hatırla!

- Psödo dilde basit ve anlaşılır komutlar kullanılır.
- Değişkenler "değişken" komutu ile oluşturulur, değer ise $\leftarrow$ sembolü ile atanır.
- "Gir" ile kullanıcıdan veri alınır, "göster" ile sonuçlar görüntülenir.
- *Temel matematiksel işlemler yapılabilir: +, -, , /.
- Giriş, hesaplama ve çıkışın birleşimi işlevsel bir algoritma oluşturur.



DAHA FAZLA BİLGİ EDİNMEK İSTEYENLER İÇİN

Mantığı daha fazla biçimlendirmek istersek, eğer ... ise, tekrar et gibi yapılar ve $x > y$ türündeki koşullar ile psödo dil kullanabiliriz. Bu yapılar, karar vermeyi ve işlemlerin tekrarlanmasını sağlar. Ayrıca gerçek programlama dillerine (Python veya C++ gibi) benzeyen, ancak sözdizimi kurallarına bağlı olmayan psödo kod biçimleri de kullanılabilir. Bu durum, psödo dili mantıksal düşünme ile uygulamalı programlama arasında bir köprü hâline getirir.

Bu konuyu tamamladığında...

- ✓ Ekrana çıktı veren basit bir komut yazabilmek.
- ✓ Sözde kod (pseudocode) ve komut (instruction)
- ✓ kavramlarını anlayabilmek.

**TEKRARLAMA SORULARI:**

1. İki değişkenin çarpımı nasıl yazılır?
2. Girdi, aritmetik işlem ve çıktının birlikte kullanılması neden önemlidir?
3. İki sayının farkını hesaplayan bir sözde kod (pseudocode) yazınız!

**AŞAĞIDAKİ KONULARDA ŞUNLARI ÖĞRENECEKSİN:**

- Farklı veri türlerinde değişkenler oluşturabilmek.

Programlamada ifade dizisi (komut dizisi), belirlenmiş bir sıraya göre — birbiri ardına, herhangi bir koşul veya atlama olmadan — çalıştırılan bir grup talimattır. Bu, program oluşturmanın temel mekanizmalarından biridir, çünkü açık ve mantıksal adımların tanımlanmasını sağlar. Bir program başlangıçtan sona doğru okunduğunda, her ifade belirli bir işlevi yerine getirir ve kodda görüldüğü sıraya göre çalıştırılır. Bu ardışıklığa sıralı çalıştırma (sekansiyel yürütme) denir ve komutların kendisi sıralı bloğu oluşturur. Sekans, daha sonra koşullar ve döngüler aracılığıyla program mantığının geliştirildiği başlangıç noktasıdır, bu nedenle doğru anlaşılması büyük önem taşır.

Bir programdaki her ifade, bilgisayarın gerçekleştirdiği bir komutu temsil eder. Bu ifadeler herhangi bir ek kontrol yapısı olmadan, örneğin koşullar veya tekrarlar kullanılmadan alt alta sıralandığında bir sekans oluştururlar. Böyle bir blokta sıra son derece önemlidir, bilgisayar komutları yukarıdan aşağıya doğru doğrusal biçimde çalıştırır. Bu, bir komutun sonucunun sonraki komutu etkileyebileceği anlamına gelir. Eğer bir komut, gerekli veriler

hazırlanmadan önce çalıştırılacak şekilde yanlış yere yerleştirilirse hata veya istenmeyen sonuç ortaya çıkabilir. Bu nedenle programlamaya küçük ve mantıksal olarak birbirine bağlı komut dizileri yazarak başlamak önerilir.

ÖRNEK 1



Değişken tamsayı $a \leftarrow 10$

Değişken tamsayı $b \leftarrow 5$

Değişken tamsayı toplam $\leftarrow a + b$

Yazdır «Toplam: », toplam

Bu örnekte her satır bir önceki satıra bağlıdır. Sıralama değiştirilirse, örneğin Yazdır komutu toplam $\leftarrow a + b$ ifadesinden önce yer alırsa, hata oluşur. Bu nedenle komutların sırası dikkatlice planlanmalıdır.

İyi düzenlenmiş bir işlem sırası (sekans), kodun kolay okunmasını ve öngörülebilir şekilde çalışmasını sağlar. Eğitim bağlamında bu, öğrencilerin adım adım ve mantıklı düşünmeyi, ayrıca bir işlemten diğerine nasıl geçildiğini anlamalarını sağlar. Böylece yalnızca programlama becerileri değil, aynı zamanda analitik düşünme becerileri de gelişir. Bu yaklaşım yalnızca kod yazarken değil, günlük yaşamda karşılaşılan problemlerin çözümünde de yararlıdır, çünkü bu durumlarda da mantıksal bir işlem sırası izlemek önemlidir.

ÖRNEK 2



Ekrana yaz «İki tam sayı giriniz: «

Oku x, y

Tam sayı değişkeni çarpım $\leftarrow x * y$

Ekrana yaz «Çarpım: », çarpım

Bu işlem sıralamasında (sekans), kullanıcı önce değerleri girer, ardından hesaplama yapılır ve son olarak sonuç ekranda gösterilir. Geçerli bir sonuç elde edebilmek için her satırın doğru sırayla yazılması gerekir.

Bunun yanı sıra, sekans program mantığının modüler bir şekilde düzenlenmesini, yani aynı amaca hizmet eden işlemlerin bir araya getirilmesini sağlar. Böylece kolay anlaşılabilen ve test edilebilen mantıksal bloklar oluşturulabilir. Bu nedenle programcılar, programlarını genellikle işlevsel bölümlere ayırır ve her bölüm kendi işlem sırasından (sekansından) oluşur. Daha sonra koşullar ve döngüler gibi daha karmaşık yapılara geçildiğinde bile, bu yapıların içindeki her bölüm yine bir komut dizisinden (sekansından) oluşacaktır.

ÖRNEK 3

Tam sayı değişkeni $m \leftarrow 8$

Tam sayı değişkeni $n \leftarrow 3$

Tam sayı değişkeni $\text{fark} \leftarrow m - n$

Ekrana yaz «Fark: », farkzlika

Yine, komutlar sırayla çalıştırılmalıdır. Eğer fark değişkenini hesaplanmadan önce kullanmaya çalışırsak, hata oluşur.



SONUÇ

Komut dizisi (sekans), her programın temelini oluşturur. Komutların sırasıyla nasıl çalıştırıldığını anlamak, programlama öğrenmenin en önemli adımlarından biridir. Bu kavram anlaşılmadan daha karmaşık programlama yapılarının öğrenilmesi mümkün değildir. Bu nedenle öğrenciler yalnızca komut yazmayı değil, aynı zamanda bu komutları mantıklı ve anlamlı bir sırayla düzenlemeyi de öğrenmelidir. Bu sayede sağlam, işlevsel ve okunabilir programlar geliştirebilirler.

BİLGİ KONTROLÜ SORULARI VE ETKİNLİKLER



1. İki tam sayı okuyan ve bunların toplamını ve çarpımını ekrana yazdıran bir program yazınız!
2. Önce değerlerin girildiği, ardından bir hesaplama yapılan ve son olarak sonucun gösterildiği bir sekans oluşturunuz!
3. Komutların sırasını değiştirin ve çıktıyı karşılaştırın!
4. Bir değişkeni tanımlanmadan önce kullanmamak neden önemlidir?

Öğrendiklerini Hatırla!

- Sekans, mantıksal olarak birbirine bağlı komutlardan oluşur.
- Her komut atlama yapılmadan sırayla çalıştırılır.
- Hatalar çoğunlukla komut sırasının bozulması nedeniyle ortaya çıkar.
- Kodun mantıksal bloklar halinde düzenlenmesi, okunmasını kolaylaştırır.
- İyi yazılmış bir sekans, kararlı bir programın temelidir.



DAHA FAZLA BİLGİ EDİNMEK İSTEYENLER İÇİN

- Robotikte sıralı çalıştırmanın nasıl uygulandığını araştırın!
- Günlük görevlerdeki sekansları tanıyın: yemek pişirme, okula yürüme gibi!
- Üç farklı sekans kullanan bir program yazmayı deneyin!
- Sekansın ortasında kesilirse ne olduğunu deneyerek gözlemleyin!



Bu konuyu tamamladığında...

- ✓ Sıralı çalıştırmayı tanır,
- ✓ Komut dizisi (sekans) yazabilir.

**BİLGİ KONTROLÜ SORULARI VE ETKİNLİKLER:**

1. İfade dizisinin (komut sekansının) ne olduğunu açıklayın!
2. Komutların sırasının doğru olması neden önemlidir?
3. Mantıksal olarak birbirine bağlı üç komut yazın!
4. Komutların yerleri değiştirilirse ne olur?
5. Sıralı çalıştırmanın avantajları nelerdir?

**AŞAĞIDAKİ KONULARDA ŞUNLARI ÖĞRENECEKSİN:**

- Değişkenler ve sabitler oluşturur.
- Değişkenlere başlangıç değeri atar.
- Değişkenlerin ve sabitlerin çalışma biçimleri arasındaki farkı ayırt eder.

Her programda değişen veya aynı kalan bilgiler kullanılır. Bilgisayarın bu bilgilerle çalışabilmesi için veriler **değişkenler** veya **sabitler** adı verilen yapılarda saklanır. Değişkenler, programın çalışması sırasında değişebilen verileri saklamak için kullanılırken, sabitler değeri değiştirilemeyen veriler için kullanılır. Bu iki yapı, programlamadaki her türlü mantığın temelini oluşturur ve bilgilerin esnek bir şekilde yönetilmesini sağlar.

Değişken, bilgisayarın belleğinde bulunan ve değişebilen bir değeri saklamak için kullanılan isimlendirilmiş bir alandır. Her değişkenin bir adı (**tanımlayıcı**), veri tipi ve değeri vardır. İsim, değişkeni tanımaya yarar, veri tipi, hangi tür değerleri içerebileceğini belirtir, değer ise saklanan gerçek sayı veya metindir. Bir değişken kullanılmadan önce tanımlanmalıdır, yani programa hangi tür veriyi saklayacağı bildirilmelidir. Ayrıca değişkene hemen başlangıç değeri verilebilir, buna **başlatma (initialization)** denir.

ÖRNEK 1



```
int yaş = 16
```

Bu örnekte, «yaş» adlı bir değişken tanımlanmıştır ve bu değişken bir tam sayı değeri içerir. `int` veri türü, bu değişkende saklanabilecek değerin tam sayı olduğunu, yani ondalık kısmı olmayan bir sayı olduğunu belirtir. Değişkene atanan başlangıç değeri 16'dır. Bu değer, başlangıç değeri (ilk değer) olarak adlandırılır ve programın çalışması sırasında, görevin ihtiyaçlarına veya kullanıcının etkileşimine bağlı olarak değiştirilebilir. «yaş» değişkeni bir kez tanımlandığında, kullanılabilir hale gelir ve programın farklı bölümlerinde, ihtiyaç duyulan her yerde kullanılabilir. Değişken tanımlandıktan sonra, farklı işlemler ve hesaplamalar için serbestçe kullanılabilir.

Örneğin, kullanıcıya yeni bir yaş değeri girme imkânı verilirse, değişkenin değeri girilen yeni değerle kolayca güncellenebilir. Ayrıca değişken, toplama, çıkarma veya diğer değişkenlerle karşılaştırma gibi matematiksel işlemlerde kullanılabilir. Değişkenin içerdiği değer herhangi bir anda ekranda gösterilebilir. Bu özellik, programları daha etkileşimli ve dinamik hale getirir. Bunun gibi değişkenlerin kullanılmasıyla programlar daha esnek olur ve duruma veya kullanıcı girişine göre uyarlanabilir. Bu nedenle değişkenler, programlamanın temel unsurlarından biridir, çünkü programların çalışması sırasında bilgilerin saklanmasına ve değiştirilmesine olanak sağlarlar.

ÖRNEK 2



```
int const sayı = 8,
```

Bu örnekte, «sayı» adlı sabit bir değişken tanımlanmıştır ve bu değişken bir tam sayı içerir. Bu değişkenin türü `int`'tir, bu, saklanan değerin tam sayı olması gerektiği anlamına gelir. `const` belirleyicisi ise, bu değişkene bir kez değer atandıktan sonra programın çalışması boyunca bu değerin değiştirilemeyeceğini belirtir. Bu durumda başlangıç değeri 8'dir ve programın tüm çalışması boyunca sabit ve değişmeden kalır. Bu tür sabitlerin kullanılması, özellikle programda kullanılacak değerlerin önceden bilindiği ve hiçbir zaman değiştirilmemesi gerektiği durumlarda oldukça yararlıdır.

Sabit değişkenler çok önemlidir çünkü programı daha okunabilir, anlaşılır ve güvenli hale getirir. Sabitin değeri tanımlandıktan sonra değiştirilemediği için programcı, değerinin yanlışlıkla veya istenmeden değiştirilmesinden kaynaklanan hataların oluşmayacağından emin olur. Bu da kod içinde oluşabilecek hataları veya yanlış anlamaları önlemeye yardımcı olur. Sabitler, matematiksel sabitler (örneğin π veya yerçekimi sabiti) gibi değişmeyen değerleri tanımlamak için kullanılabileceği





gibi, program boyunca aynı kalan parametreleri tanımlamak için de kullanılır. Örneğin haftanın gün sayısı veya yılın ay sayısı gibi değerler sabit olarak tanımlanabilir. Bu tür sabitlerin kullanılması, gelecekte değerlerin değiştirilmesini de kolaylaştırır, çünkü değer yalnızca kodun tek bir yerinde tanımlanır. Bu özellik, özellikle çok satırlı ve karmaşık programlarda programların daha kolay bakımını ve yönetilmesini sağlar.



SONUÇ

Değişkenler ve sabitler, her programın temel yapı taşlarıdır. Bunlar, değiştirilebilir veya sabit olan verilerin saklanması ve yönetilmesini sağlar. Bu yapılar sayesinde birçok mantıksal işlem, hesaplama ve kullanıcıyla etkileşim gerçekleştirilir. Değişkenlerin ve sabitlerin nasıl tanımlandığını, kullanıldığını ve korunduğunu anlamak, programlama öğrenme sürecinde temel bir adımdır. Bunlar, programın bilgileri «hatırlamasını» ve işlemlerini sağlayan araçlardır.

BİLGİ KONTROLÜ SORULARI VE ETKİNLİKLER



1. Bir yükseklik değişkeni tanımlayın ve ona bir değer atayın!
2. MAX_OCENKA adlı bir sabit oluşturun ve değerini 5 olarak ayarlayın!
3. Sabitin değerini değiştirmeye çalışırsanız ne olur?
4. Değişkenlerin açıklayıcı isimlere sahip olması neden önemlidir?

Öğrendiklerini Hatırla!

- Değişkenler, değişebilen değerleri saklamak için kullanılır.
- Sabitler, değişmeyen değerleri saklar.
- Her değişkenin bir türü, adı ve değeri vardır.
- Başlatma (initialization), başlangıç değerinin atanmasıdır.
- Sabitler const ile tanımlanır ve değiştirilemez.



DAHA FAZLA BİLGİ EDİNMEK İSTEYENLER İÇİN

- Oyunlarda değişkenlerin nasıl kullanıldığını araştırın!
- Hem değişkenleri hem de sabitleri kullanan bir program yapmayı deneyin!
- Başlatma (initialization) yapılmadan bir değişken kullanırsanız ne olduğunu inceleyin!
- Farklı programlama dillerinin (C++, Python, Java) sabitlerle nasıl çalıştığını karşılaştırın!
 - Değişkenin değerini birden fazla kez değiştiren bir kod yazın!



3.14

Değişken Türleri

Bu konuyu tamamladığında...

- ✓ Değişkenleri sabitlerden ayırt eder
- ✓ Programda metin ve sayıları birbirinden ayırt eder.



TEKRARLAMA SORULARI:

1. Değişken nedir?
2. Bir değişkenin başlatılması (ilk değer verilmesi) nedir?
3. Bir değişken yerine sabit ne zaman kullanılır?
4. Bir sabitin değerini değiştirmek mümkün müdür?



AŞAĞIDAKİ KONULARDA ŞUNLARI ÖĞRENECEKSİN:

- Farklı türlerde değişkenler oluşturur.

Bilgisayar programlamasında kullanılan her veri, belirli bir veri tipine ait olmak zorundadır. Veri tipi, programa ne kadar bellek ayırması gerektiğini ve bu veriyi nasıl işleyeceğini bildirir. Örneğin, ekranda aynı görünen 5 sayısı ile «5» metni tamamen farklı verilerdir. Bu nedenle doğru ve işlevsel bir program yazarken uygun değişken tipinin seçilmesi temel bir adımdır. En sık karşılaşılan temel veri tipleri şunlardır: Tam sayılar (int), ondalıklı sayılar (double veya float), karakterler (char), mantıksal değerler (bool).

ÖRNEK 1



```
double ortalama = 4.75,
```

Burada ortalama değişkeni, bir öğrencinin not ortalamasını saklar ve bu değer ondalıklı bir sayıdır. Eğer int kullanılsaydı, 4,75 değeri 4 olarak alınır. Burada matematikteki gibi yuvarlama söz konusu değildir, ondalık kısmın kesilmesi söz konusudur. Aslında bu durumda yalnızca tam sayı kısmı alınır.

Bir diğer ilginç veri türü ise char türüdür. Bu tür, harfler veya özel semboller gibi tek tek karakterleri saklamak için kullanılır. Bu tür, baş harfler, işaretler veya semboller işlendiğinde idealdir. Karakterin etrafına tek tırnak işaretleri yazılarak tanımlanır ('A', '%', '#'). Bu tür ile matematiksel işlemler yapılmaz, sembolik gösterim için kullanılır.

ÖRNEK 2



`char harf = 'A',`

Char veri türü, harfler, rakamlar veya semboller gibi tek tek karakterleri saklamak için kullanılır. Örneğin, `char harf = 'A',` ifadesi, harf adlı bir değişken tanımlar ve bu değişkende A karakterini saklar. C++ dilinde karakterler her zaman tek tırnak işaretleri içinde yazılır. Bu veri türü, metinsel verilerle tek tek karakter düzeyinde çalışırken veya program içinde belirli karakterleri kontrol etmek ve karşılaştırmak için kullanılır.

ÖRNEK 3



`string isim = "Ana"`

Bu örnekte `string` (karakter dizisi) türünde "isim" adlı bir değişken tanımlanmıştır. Bu değişken türü, metinsel verileri, yani kelimeleri, ifadeleri veya cümleleri saklamak için kullanılır. Bu özel durumda "isim" değişkenine "Ana" başlangıç değeri verilmiştir. Bu, programın çalışması sırasında bu değişkene başvurulduğunda tam olarak bu metinsel veriyi içereceği anlamına gelir. `String` türü, isimler, adresler, mesajlar ve bir veya daha fazla karakterden oluşan benzer bilgilerin saklanması ve işlenmesi gerektiğinde oldukça kullanışlıdır.

Bir kez tanımlandıktan sonra "isim" değişkeninin değeri ihtiyaca göre değiştirilebilir, yani başka bir metin eklenebilir veya mevcut metin üzerinde değişiklik yapılabilir. Örneğin, kullanıcı yeni bir isim girerse değişkenin değeri yeni bilgiyle güncellenir. Bunun yanında metinsel değişkenler, birden fazla kelimeyi birleştirme (birleştirme/konkatinasyon), metinleri karşılaştırma, dizi içindeki karakterleri arama ve birçok farklı işlem yapma imkânı sağlar. Bu özellikler, programların daha dinamik ve esnek olmasını sağlar, çünkü metinsel bilgiler kullanıcı ile bilgisayar arasındaki iletişimin temel unsurlarından biridir.

ÖRNEK 4



`bool aktif = true,`

Bu örnekte, `bool` türünde ve «aktif» adlı mantıksal bir değişken tanımlanmıştır. `bool` türündeki mantıksal değişkenler yalnızca iki farklı değere sahip olabilir: `true` (doğru) veya `false` (yanlış). Verilen durumda, değişkene başlangıç değeri olarak `true` atanmıştır. Bu, bu değişkenin temsil ettiği koşulun şu anda sağlandığı veya aktif olduğu anlamına gelir.

Mantıksal değişkenler, programın akışını kontrol etmede önemli bir role sahiptir, çünkü programdaki komutların sonraki çalıştırılma şekli bu değişkenlerin değerine bağlıdır. Eğer bu değişkenin değeri `true` ise program belirli işlemleri gerçekleştirmeye devam eder. Eğer değeri `false` ise program başka bir işlem gerçekleştirir veya belirtilen işlemi atlar. Bu şekilde mantıksal değişkenler, programcının programın mevcut





koşullara bağlı olarak karar vermesini sağlamasına olanak tanır. Bu da programları daha etkileşimli, dinamik ve farklı durumlara ve kullanıcı ihtiyaçlarına uyarlanabilir hale getirir.



SONUÇ

Değişken türleri, programlamada yapı ve mantık oluşturmayı sağlar. Doğru tür seçimi ile doğruluk, bellek kullanımında optimizasyon ve kodun anlaşılabilirliği elde edilir. Tam sayılar (int), ondalıklı sayılar (float ve double), karakterler (char) ve mantıksal değerler (bool) neredeyse her programın temelini oluşturur. Bunların kullanım amaçlarını ve aralarındaki farkları anlamak, programlamaya yeni başlayan herkes için temel bir öneme sahiptir.

BİLGİLERİ TEKRARLAMA SORULARI



1. Bir kitaptaki sayfa sayısını saklamak için hangi değişken türünü kullanırsın?
2. Int ve double hassasiyet açısından nasıl farklılaşır?
3. Double tam sayı ile tanımlanırsa ne olur?
4. Char ne için kullanılır? Örnek verin.
5. bool türü ne zaman kullanılır?

Öğrendiklerini Hatırla!

- Her verinin kendine ait bir türü vardır.
- Int tam sayılar için kullanılır.
- Double ondalıklı sayılar için kullanılır.
- Char tek bir karakter saklar.
- Bool mantıksal değerleri saklar: true veya false.
- Doğru tür seçimi daha iyi bir program anlamına gelir.



DAHA FAZLA BİLGİ EDİNMEK İSTEYENLER İÇİN

- Float, long, short veri türlerini araştırın!
- Türler arasında değer aktarımı sırasında ne olduğunu inceleyin!
- Bool türünün programın akışını yönettiği bir örnek yazın!
- Char türü bir sayı olabilir mi? Araştırın!
- Double türü int türüne nasıl dönüştürülür?



3.15

Bir Değişkene Değer Atama

Bu konuyu tamamladığında...

- ✓ Değişkenleri türlerine göre ayırt eder,
- ✓ Tam sayıyı, ondalıklı sayıyı, karakteri ve mantıksal değeri tanır,
- ✓ Uygun ad ve tür ile yeni bir değişken oluşturur.



TEKRARLAMA SORULARI:

1. int türünde bir yaş değişkeni tanımlayın!
2. Ağırlık için bir double değişkeni yazın!
3. Baş harf için bir char değişkeni oluşturun!
4. Sayının pozitif olup olmadığını kontrol edecek bir bool tanımlayın!



AŞAĞIDAKİ KONULARDA ŞUNLARI ÖĞRENECEKSİN:

- Değişkenlere değer atarsın.
- Değer çakışmasını (üst üste gelmesini) fark edersin.

Programlama dünyasında her değişken yalnızca bir ada ve tipe değil, aynı zamanda bir değere de sahiptir. Bu değer, değişken oluşturulurken belirlenebilir veya programın çalışması sırasında daha sonra atanabilir. Bir değişkenin değerinin belirlenmesi veya değiştirilmesi işlemine değer atama adı verilir. Bu işlem için = operatörü kullanılır. Matematiksel eşitlik işaretine benzese de programlamada = işareti "ata" anlamına gelir. Yani ifadenin sağ tarafı hesaplanır ve sonuç soldaki değişkende saklanır.

Bir değişkene değer atama işlemi birkaç farklı şekilde gerçekleştirilebilir. Bunlardan ilki başlatma (initialization) işlemidir, yani değişken oluşturulurken ona başlangıç değeri verilmesidir. Bu yöntem basittir ve kodun okunabilirliğini artırır. Örneğin, başlangıç sıcaklığını hemen belirlemek istersek şu şekilde yazabiliriz:

ÖRNEK 1



```
int sıcaklık = 22
```

Bu durumda sıcaklık adlı int türündeki değişkene, oluşturulurken 22 değeri verilir. Bu işleme başlatma (initialization) denir ve değer önceden bilindiğinde en yaygın kullanılan yöntemdir.

Değişkenin bir hesaplama sonucunda değer alması gerektiği durumlarda, önce değer verilmeden tanımlanabilir ve daha sonra bir ifade aracılığıyla değer atanabilir. Örneğin, iki dersin not ortalamasını hesaplamak istediğimizde bu yöntem kullanılabilir.

ÖRNEK 2



Burada ortalama başlangıçta yalnızca tanımlanmıştır, ancak daha sonra kendisine bir ifadeden elde edilen değer atanır. Bu durum, değişkenin dinamik olarak değer alabileceğini gösterir ve etkileşimli programlar için temel bir özelliktir.

Buna ek olarak, bir değişkenin değeri programın çalışması sırasında birden fazla kez değiştirilebilir. Bu, değişkenlerin esnek olduğu ve içeriğinin güncellenebildiği anlamına gelir. Buna değer üzerine yazma (değer değiştirme) denir ve özellikle sayma, biriktirme veya koşullarla çalışırken yaygın olarak kullanılan bir durumdur.

ÖRNEK 3



```
int sonuç = 0
```

```
sonuç= sonuç + 5 sonuç = sonuç* 2
```

Öncelikle sonuç değişkenine 0 değeri verilir. Daha sonra değeri 5 artırılır ve ardından 2 ile çarpılır. Son değer 10 olur. Bu, kod içinde daha karmaşık mantıkların oluşturulmasını sağlayan açık bir değer değiştirme örneğidir.

Önemli olarak belirtilmelidir ki = operatörünün sol tarafında her zaman bir değişken bulunmalıdır, yani değerini değiştirebileceğimiz bir ifade olmalıdır. Sağ taraf ise herhangi bir ifade olabilir: sayılar, diğer değişkenler, matematiksel işlemler ve hatta fonksiyon çağrıları. Bu durum, = operatörünü her programlama dilinin en önemli bölümlerinden biri yapar.

```
function filterUsers(list, options = {}) {  
  function filter(user => {  
    return options.name ? user.name.toLowerCase().includes(options.name.toLowerCase()) : true;  
    return options.email ? user.email.toLowerCase().includes(options.email.toLowerCase()) : true;  
  })  
  return list.filter(filter);  
}
```



SONUÇ

Değişkene değer atama, programlamada temel ve vazgeçilmez bir araçtır. Bu işlem sayesinde esneklik, hesaplama ve veri saklama sağlanır. Bu işlem olmadan değişkenler boş ve anlamsız kalırdı. Başlatma (ilk değer verme), hesaplama ve değer değiştirme arasındaki farkı anlamak, programların doğru ve kararlı bir şekilde çalışması için büyük önem taşır.

BİLGİ KONTROLÜ SORULARI VE ETKİNLİKLER



1. Bir değişkenin başlatılması (ilk değer verilmesi) ne anlama gelir?
2. = operatörü ne zaman kullanılır?
3. Bir sabite değer atanmaya çalışılırsa ne olur?
4. $x = x + 1$ ile $x = 1$ arasındaki farkı açıklayın!
5. İfadenin sağ tarafı birden fazla değişken içerebilir mi?

Öğrendiklerini Hatırla!

- = eşitlik anlamına gelmez, değer atama anlamına gelir.
- Değer sağdan sola doğru atanır.
- Bir değişkenin değeri birden fazla kez değiştirilebilir.
- Değer atama, her hesaplama algoritmasının temelidir.
- Başlatma, temiz ve düzenli kod oluşturmayı sağlar.



DAHA FAZLA BİLGİ EDİNMEK İSTEYENLER İÇİN

- «Chain assignment» (zincirleme atama) nedir araştırın: $a = b = c = 5$!
- Farklı türlerdeki ifadeler atanırsa ne olduğunu inceleyin!
- Bir const değişkeni başlatmayı deneyin – ne olur?
- $x = x + 1$ matematiksel olarak geçerli midir? Programlama açısından geçerli midir?
- Bir değişken program boyunca kaç değere sahip olabilir?



3.16

Değişken Değerinin Gösterimi

Bu konuyu tamamladığında...

- ✓ Değişkenler oluşturur ve değişkenlere değerler atar, temel
- ✓ Temel veri türlerini: tamsayı, ondalıklı ve karakter türlerini kullanır,
- ✓ Metinsel ve sayısal verileri birbirinden ayırt eder.



TEKRARLAMA SORULARI:

1. Yaş değişkenini bildir ve 18 değeri ile başlat!
2. Fiyat değişkenini oluştur, ona 120 değeri ver, sonra onu 30 artır!
3. Alan = uzunluk * genişlik formülünü ifade et!
4. Toplam değişkeni, sayı 1 ve sayı 2'nin toplamı olarak elde edilsin!
5. Klavyeden veri okunduktan sonra bir değişkene değer ver!



AŞAĞIDAKİ KONULARDA ŞUNLARI ÖĞRENECEKSİN:

- Değişkenlerin değerlerini ekranda gösterirsin,
- Temel yazdırma komutlarını kullanırsın,
- Çıkışı özel karakterlerle biçimlendirirsin,
- Görüntülemeye metinleri değişkenlerle birleştirirsin,
- Tek bir komut içinde birden fazla değişken gösterirsin,
- Yeni satırda veya sekme kullanarak yazdırmanın nasıl çalıştığını anlarsın.

Bir değişken değerinin gösterimi, program ile kullanıcı arasındaki iletişimin en önemli araçlarından birini oluşturur. Bunun yardımıyla, hesaplama sonuçları, verilerin durumu veya hata mesajları ekranda açık bir şekilde gösterilebilir. Bu işlem, yazdırma için özel komutların kullanılmasıyla gerçekleştirilir ve C++ programlama dilinde en sık kullanılan komut `cout`'tur. Basit bir işlem olmasına rağmen, değerlerin doğru şekilde gösterimi, programın okunabilirliği, doğruluğu ve profesyonelliği açısından büyük önem taşır.

C++ ile hazırlanan programlarda başlangıçta şu ifade yer alır:

```
#include <iostream>
```

```
using namespace std
```

`#include <iostream>` derleyiciye, C++ dilinde giriş-çıkış işlemleri için kullanılan standart kütüphaneyi, yani **iostream (input-output stream)** başlık dosyasında bulunan kütüphaneyi eklemesini söyler. Bu sayede klavyeden veri girişi için `cin`, ekranda gösterim için `cout` ve hata mesajlarını görüntülemek için `cerr` gibi temel nesnelerin kullanılması mümkün olur. Bu komut olmadan program, **cout** veya **cin** ifadelerinin ne anlama geldiğini bilemez ve derleme sırasında hatalar oluşur.

`using namespace std`, programa, `std` (standard) ad alanındaki isimlerin kullanılacağını bildirir, böylece her zaman önlerine **std::** yazmaya gerek kalmaz. Örneğin, `std::cout` yazmak yerine yalnızca `cout` kullanılabilir. Bu, başlangıç seviyesindeki kullanıcılar için söz dizimini basitleştirir ve kodun daha kolay okunmasını sağlar, ancak daha büyük projelerde diğer ad alanlarıyla oluşabilecek çakışmaları önlemek için `std::cout` kullanılması daha uygundur.

Bir değeri göstermenin en temel yolu `cout` komutu aracılığıyla gerçekleştirilir. `cout` adı, **"character output"** (karakter çıkışı) anlamına gelir ve `std` ad alanının bir parçasıdır. Bu nedenle `std::cout` şeklinde veya `using namespace std`, kullanılmışsa yalnızca `cout` şeklinde kullanılır. Programcının metinleri, sayıları veya değişkenlerin değerlerini doğrudan ekranda göstermesine olanak sağlar. `cout` komutu her zaman, veriyi ekrana yönlendiren `<<` operatörü ile birlikte kullanılır. Örneğin:

ÖRNEK 1

```
n.cpp x
1  #include <iostream>
2
3  using namespace std;
4
5  int main()
6  {
7      int sayi = 7;
8      cout << "Sayının değeri: " << sayi << endl;
```

Bu komut ekranda şu sonucu gösterecektir: Sayı değeri: 7. Metin ile değişkenin tek bir satırda birleştirilebildiği görülmektedir, bu durum dinamik ve anlaşılır bir gösterim sağlar.

Çoğu durumda birden fazla değerin gösterilmesi veya verilerin satır ya da tablo şeklinde düzenlenmesi gerekir. Bu nedenle yeni satır için `\n` veya sekme için `\t` gibi özel karakterler kullanılır. Bu sözde "escape" karakterleri doğrudan gösterilmez, bunun yerine çıkışın görünümünü etkiler. Örneğin:

ÖRNEK 2



```
1 #include <iostream>
2
3 using namespace std;
4
5 int main()
6 {
7     double pi = 3.14159;
8     cout << "Pi sayısının yaklaşık değeri: " << pi << endl;
9     return 0;
10 }
```

Çıktı şu şekilde olacaktır: Pi sayısının yaklaşık değeri: 3.14159. Daha karmaşık durumlarda, ondalık basamak sayısını kontrol etmek veya metnin hizalanmasını sağlamak için <iomanip> gibi ek kütüphaneler kullanılabilir.

ÖRNEK 3



```
1 #include <iostream>
2
3 using namespace std;
4
5 int main()
6 {
7     int a = 5, b = 10;
8     cout << "a = " << a << "\tb = " << b << endl;
9     return 0;
10 }
11
```

Çıktı tablo şeklinde düzenlenecektir: a = 5 b = 10. Bu, \t karakterinin bilgileri tablo biçiminde ayırdığını, endl komutunun ise \n ile aynı şekilde yeni bir satır başlattığını gösterir.

Yazdırma işlemi çoğunlukla değerlerin kontrol edilmesi, hata ayıklama veya kullanıcıya belirli olaylar hakkında bilgi verilmesi amacıyla kullanılır. Bu gibi durumlarda bilgilerin açık ve kısa bir şekilde gösterilmesi büyük önem taşır. Biçimlendirme, sabit genişlik ayarlama, ondalık sayıların yuvarlanması veya verilerin sütunlar hâlinde hizalanması ile geliştirilebilir.

Tüm bu teknikler, bilgilerin açık ve düzenli bir şekilde gösterilmesini sağlar. Bu durum yalnızca estetik açıdan değil, aynı zamanda program sonuçlarının doğruluğunun ve zamanında anlaşılmasının sağlanması açısından da büyük önem taşır. Bu özellikle büyük verilerin analizinde, tablolarla çalışmada veya mantıksal akışların izlenmesinde geçerlidir.

```
function filterUsers(list, options = {}) {  
function return list.filter(user => {  
return options.name ? user.name.toLowerCase().includes(options.name.toLowerCase()) : true;  
return options.email ? user.email.toLowerCase().includes(options.email.toLowerCase()) : tr  
true
```



SONUÇ

Değişken değerinin gösterilmesi, her programın gerekli bir parçasıdır. Bu işlem, kullanıcı ile makine arasındaki arayüzü oluşturur. Doğru kullanımı sayesinde program daha okunabilir, işlevsel ve profesyonel hale gelir. Değişkenlerin, metinlerin ve özel karakterlerin birleştirilmesi, esnek ve kullanıcıya uygun çıktıların oluşturulmasını sağlar. Bu konunun öğrenilmesi, gerçek uygulamalar geliştirmeye bir adım daha yaklaşmayı sağlar.

BİLGİLERİ TEKRARLAMA SORULARI



1. cout işlevi nedir ve ne için kullanılır?
2. Bir değeri görüntülemek için hangi operatör kullanılır?
3. \n ile endl arasındaki fark nedir?
4. Tek bir komutta birden fazla değişkenin değeri nasıl gösterilebilir?
5. Çıktıda \t kullanımının etkisi nedir?

Öğrendiklerini Hatırla!

- cout işlevi, değerleri görüntülemenin temel yoludur.
- << operatörü, ekrana yönlendirme yapmak için kullanılır.
- \n ve \t gibi kaçış karakterleri metnin biçimlendirilmesini sağlar.
- Metin ve değerler birleştirilebilir.
- Görüntüleme, kullanıcıyla doğru iletişim kurmak için önemlidir.



DAHA FAZLA BİLGİ EDİNMEK İSTEYENLER İÇİN

- <iomanip> kütüphanesini ve setprecision ile setw fonksiyonlarını araştırın!
- Sonuçları tablo biçiminde göstermeyi deneyin!
- Hata ayıklama (debug) amacıyla çıktı kullanın – her adımdaki değerleri gösterin!
- Çizgiler ve başlıklar içeren bir program arayüzü oluşturun!
- C++ dilinde Kiril karakterleriyle metin görüntüleyin!



Bu konuyu tamamladığında...

- ✓ Değişkenleri kullanır ve onlara değer atar,
- ✓ Sonucu ekranda gösterir,
- ✓ Tam sayılar ile ondalıklı sayıları ayırt eder.

**TEKRARLAMA SORULARI:**

1. İsim ve yaş değişkenlerini tanımlayın ve ekranda gösterin!
2. Değerleri yeni satırda göstermek için \n kullanın!
3. Boy ve kilo için değişkenler oluşturun ve bunları açıklamayla birlikte gösterin!
4. Üç sayıyı sıralı şekilde, sekme boşluğu kullanarak gösterin!
5. cout kullanarak $5 + 3 = 8$ ifadesini gösterin!

**AŞAĞIDAKİ KONULARDA ŞUNLARI ÖĞRENECEKSİN:**

- Toplama, çıkarma, çarpma ve bölme gibi aritmetik işlemleri kullanırsın.
- İşlem önceliği kurallarını uygularsın.
- İfadeleri kontrol etmek için parantez kullanırsın.
- Tam sayı bölmesi ile ondalıklı bölme arasındaki farkı anlarsın.
- Kalan (mod) operatörünü kullanırsın.
- Birden fazla işlemi birleştiren ifadeleri çözersin.

Aritmetik işlemler, programlamada yapılan neredeyse her hesaplamanın temelini oluşturur. Bu işlemler sayılar üzerinde işlem yapılmasını sağlar, toplama, çıkarma, çarpma veya bölme gibi işlemlerden bahsedilebilir ve en basit işlemlerden en karmaşık mantıksal çözümlere kadar hemen hemen tüm durumlarda kullanılır. Bu derste ifadelerin nasıl oluşturulduğu ve hesaplandığı, hangi operatörlerin kullanıldığı ve işlem sırasının sonuç üzerinde nasıl etkili olduğu açıklanacaktır.

Aritmetik işlemler standart matematik sembollerini içerir: + toplama, - çıkarma, * çarpma, / bölme ve % bölme işleminden kalan için kullanılır. Bu operatörler, ifadeler oluşturmak için değerler ve değişkenlerle birlikte kullanılır. Temel bir ifade şu şekilde olabilir: $a + b * c$, burada a, b ve c değişkenlerdir.

ÖRNEK 1

```
1  #include <iostream>
2
3  using namespace std;
4
5  int main()
6  {
7      string isim;
8      int a = 5, b = 3, c = 2;
9      int sonuc = a + b * c;
10     cout << "Sonuç: " << sonuc << endl;
11     return 0;
12 }
```

Bu durumda, çarpma işlemi toplama işleminden daha önceliklidir, bu nedenle önce $b * c = 6$ hesaplanır, ardından $a + 6 = 11$ işlemi gerçekleştirilir.

Bir ifadede birden fazla işlem bulunduğunda, işlem önceliği kurallarına dikkat edilmelidir. Kural şöyledir: Çarpma ve bölme işlemleri, toplama ve çıkarma işlemlerinden daha yüksek önceliğe sahiptir. Hesaplama sırasını değiştirmek istenirse parantezler kullanılır.



ÖRNEK 2



```
1  #include <iostream>
2
3  using namespace std;
4
5  int main()
6  {
7      int a = 5, b = 3, c = 2;
8      int sonuc = (a + b) * c;
9      cout << "Sonuç: " << sonuc << endl;
10     return 0;
11 }
```

Burada, önce $a + b = 8$ işlemi gerçekleştirilir, daha sonra $8 * c = 16$ hesaplanır, çünkü parantez kullanılmıştır.

Özellikle tam sayı bölmesi ile ondalıklı bölme arasındaki farkı anlamak çok önemlidir. Tam sayılar kullanıldığında, bölme işleminin sonucu her zaman ondalık kısım olmadan tam sayı olarak elde edilir. Doğru bir ondalıklı sonuç elde etmek için sayılardan en az birinin ondalıklı sayı (double veya float türünde) olması gerekir.

ÖRNEK 3



```
1  #include <iostream>
2
3  using namespace std;
4
5  int main()
6  {
7      int a = 5, b = 2;
8      double sonuc = (double)a / b;
9      cout << "Ondalıklı sonuç: " << sonuc << endl;
10     return 0;
11 }
```

Burada a, bölme işleminden önce ondalıklı sayıya dönüştürülmüştür, bu nedenle sonuç 2 değil, 2.5 olacaktır.

Ek olarak, % operatörü bölme işlemindeki kalanı elde etmek için kullanılır. Örneğin, $7 \% 3$ işlemi 1 sonucunu verir, çünkü 3, 7 sayısının içinde iki kez bulunur ve 1 kalan kalır. Bu operatör özellikle çiftlik (tek-çift olma), sayma ve bölünebilirlik kontrolü ile ilgili problemlerin çözümünde oldukça kullanışlıdır.

```
function filterUsers(list, options = {}) {  
  function return list.filter(user => {  
    return (options.name ? user.name.toLowerCase().includes(options.name.toLowerCase()) : true;  
    options.email ? user.email.toLowerCase().includes(options.email.toLowerCase()) : tr  
    of options.minAge !== "number" ? user.age >= options.minAge : true; }) : true  
  }  
}
```



SONUÇ

Aritmetik ifadeler her programın mantığının ayrılmaz bir parçasıdır. Operatörlerin, operatör önceliklerinin ve parantez kullanımının doğru anlaşılması, doğru ve öngörülebilir sonuçların elde edilmesini sağlar. İşlemleri birleştirebilme ve yürütme sırasını kontrol edebilme becerisi, görevlerin çözülmesi ve gerçek uygulamaların geliştirilmesi için büyük önem taşır.

BİLGİ KONTROLÜ SORULARI VE ETKİNLİKLER



1. Hangi aritmetik operatörleri biliyorsun?
2. Hangi operatör en yüksek önceliğe sahiptir?
3. İfadelerde parantezler ne zaman kullanılır?
4. a / b ile $(double)a / b$ arasındaki farkı açıkla!
5. $8 \% 3$ işleminin sonucu ne olacaktır?

Öğrendiklerini Hatırla!

- Temel operatörler şunlardır: $+$, $-$, $*$, $/$, $\%$.
- İşlem önceliği sonucu etkiler.
- İşlemlerin sırasını değiştirmek için parantezler kullanılır.
- Tam sayı bölmesi ondalık değer döndürmez.
- $\%$ operatörü kalanı döndürür.
- Doğruluk, değişkenlerin türüne bağlıdır.



DAHA FAZLA BİLGİ EDİNMEK İSTEYENLER İÇİN

- `<cmath>` kütüphanesindeki `pow()`, `sqrt()`, `fabs()` fonksiyonlarını araştır!
- Üç sayının ortalama değerini hesaplayan bir ifade yaz!
- Birden fazla parantez ve işlem içeren ifadeleri çöz!
- Beş işlemin tamamını kullanan mini hesap makinesi oluştur!
- Sıra hatalarının sonuçların doğruluğunu nasıl etkilediğini analiz et!



Bu konuyu tamamladığında...

- ✓ Değişkenler oluşturursun,
- ✓ Değişkenlere değer atarsın,
- ✓ Sonucu ekranda görüntülersin.

**TEKRARLAMA SORULARI:**

1. $4 + 2 * 3$ işleminin sonucunu hesapla ve işlem sırasını açıkla!
2. Önce x ile y 'yi toplayıp, ardından sonucu z 'ye bölen bir ifade yaz!
3. 25'in 6'ya bölünmesinden kalan değeri hesapla!
4. $a = 7$ değerini ondalıklı sayıya dönüştür ve 2'ye böl!
5. $a = (b + c) * d$ şeklinde bir ifade yaz!

**AŞAĞIDAKİ KONULARDA ŞUNLARI ÖĞRENECEKSİN:**

- Klavyeden veri girişi için kullanılan komutları kullanırsın,
- Farklı veri türlerini tanırsın,
- Temel girdi doğrulaması yaparsın,
- Metinsel ve sayısal veri girişi arasındaki farkı ayırt edersin,
- Aynı anda birden fazla veri girersin,
- Veri girişi sırasında oluşan hataları çözersin.

Bu örnekte program, kullanıcının bir tam sayı girmesine olanak sağlar ve ardından girilen değeri doğrulama amacıyla tekrar gösterir. Girilen veri türünün, verinin saklandığı değişkenin türüyle uyumlu olması gerektiğini anlamak önemlidir. Kullanıcı sayı beklenirken metin girerse hata oluşur. Bu durum, temel bir kontrolün yani girdi doğrulamasının gerekliliğini ortaya koyar. Uygun doğrulama mekanizmaları programlama diline ve uygulamanın karmaşıklığına bağlı olarak değişir, ancak başlangıç düzeyinde program, geçersiz bir değer girildiğinde kullanıcıdan yeniden giriş isteyecek şekilde tasarlanabilir.

Klavyeden veri girmenin temel yöntemi, değerleri okuyan ve bunları değişkenlere kaydeden özel fonksiyonların veya operatörlerin kullanılmasıyla gerçekleştirilir. C++ gibi çoğu programlama dilinde bu amaçla cin operatörü kullanılır. Bu operatörün kullanılmasıyla program klavyeden gelen verileri "dinleme" ve girilen bilgileri belirli bir değişkene yerleştirme imkânı sağlanır. Bu, girilen verilere tepki veren etkileşimli uygulamaların oluşturulması için büyük önem taşır.

ÖRNEK 1



```
1  #include <iostream>
2
3  using namespace std;
4
5  int main()
6  {
7      int sayi;
8      cout << "Bir sayı girin: ";
9      cin >> sayi;
10     cout << "Girdiğiniz sayı: " << sayi << endl;
11     return 0;
12 }
```

Bu örnekte program, kullanıcının bir tam sayı girmesine olanak tanır, girilen sayı daha sonra doğrulama amacıyla yeniden ekranda gösterilir.

Girilen veri türünün, verinin saklandığı değişkenin türüyle uyumlu olması gerektiğinin anlaşılması önemlidir. Kullanıcı, sayı beklenirken metin girerse hata oluşur. Bu durum, temel bir denetim olan girdi doğrulamasının (validasyonunun) gerekliliğini ortaya koyar. Uygun doğrulama mekanizmaları, kullanılan programlama diline ve uygulamanın karmaşıklığına bağlıdır. Ancak başlangıç düzeyinde program, geçersiz bir değer girildiğinde kullanıcıdan veriyi yeniden girmesini isteyecek şekilde tasarlanabilir.

ÖRNEK 2



```
1  #include <iostream>
2
3  using namespace std;
4
5  int main()
6  {
7      string isim;
8      cout << "Adınızı ve soyadınızı girin: ";
9      getline(cin >> ws, isim);
10     cout << "Adınız: " << isim << endl;
11     return 0;
12 }
```

getline() fonksiyonu, sözcükler arasındaki boşluklar da dâhil olmak üzere tam bir metin satırının girilmesini sağlar.



SONUÇ

Veri girişi, her etkileşimli programın temelini oluşturur. cin gibi operatörler aracılığıyla program kullanıcısıyla iletişim kurar ve girilen değerlere göre işlem yapar. Veri türlerinin, girdi doğrulamasının ve birden fazla veriyi okuma yöntemlerinin anlaşılması, güvenilir ve kullanışlı uygulamalar geliştirmek açısından büyük önem taşır. Giriş verilerinin doğru şekilde yönetilmesi, programın kararlı çalışmasını sağlar ve kullanıcı kaynaklı hatalara karşı daha dayanıklı olmasına katkıda bulunur.

BİLGİ KONTROL SORULARI



1. cin ile sayı yerine metin girilirse ne olur?
2. Birden fazla değer aynı anda nasıl girilir?
3. Girdi doğrulaması neden önemlidir?
4. Tam sayı girişi için hangi operatör kullanılır?
5. getline() fonksiyonunun görevi nedir?

Öğrendiklerini Hatırla!

- Veri girişi için cin operatörü kullanılır.
- Girilen değerler, değişkenin veri türüne uygun olmalıdır.
- Hataları önlemek için girdi doğrulaması yapılmalıdır.
- getline() fonksiyonu, metnin tamamını tek satır hâlinde okur.
- cin boşluk karakterlerini okuyamaz, bu nedenle cümleleri okumak için getline() kullanılır.



TEKRARLAMA SORULARI:

1. İki sayı gir ve bunların toplamını hesapla!
2. Kullanıcıdan 10 ile 100 arasında bir sayı girmesini iste. Geçerli bir sayı girilinceye kadar işlemi tekrarla!
3. Bir öğrencinin adını ve soyadını girerek ekranda göster!
4. Kullanıcıdan bir sayı girmesini isteyen ve bu sayının çift olup olmadığını kontrol eden bir program yaz!
5. Üç tam sayı gir ve bunların ortalamasını hesaplayıp ekranda göster!



DAHA FAZLA BİLGİ EDİNMEK İSTEYENLER İÇİN

- Geçersiz veri girişini denetlemek için cin.fail() fonksiyonunun nasıl kullanıldığını araştır!
- Parola girişi yapan ve parolayı doğrulayan bir program yaz!
- Ad, yaş ve adres bilgilerinin girildiği küçük bir veri girişi formu oluştur!
- Metinsel veri girişi için doğrulama uygula (örneğin, en az 3 karakter)!
- Ondalıklı sayıların (double) girilmesiyle ilgili denemeler yap!



Bu konuyu tamamladığında...

- ✓ Basit C++ programları yazarsın,
- ✓ Değişkenleri kullanırsın,
- ✓ Değerleri ekranda görüntülersin,
- ✓ Temel matematik işlemleri yazarsın.

**TEKRARLAMA SORULARI:**

1. Klavyeden kenar uzunluklarını girerek bir dikdörtgenin çevresini hesaplayan bir program yaz.

**AŞAĞIDAKİ KONULARDA ŞUNLARI ÖĞRENECEKSİN:**

- C++ dilinde karşılaştırma operatörlerini kullanırsın,
- Farklı karşılaştırma ilişkilerini ayırt edersin,
- Bir ifadenin doğru veya yanlış olup olmadığını değerlendirirsin,
- Karşılaştırma operatörlerini değişkenlerle birleştirirsin,
- Karşılaştırma sonucunu program içinde kullanırsın.

Günlük hayatta sık sık değerlerin karşılaştırılması yapılır: bir sayının diğerinden büyük olup olmadığı, iki değer eşit olup olmadığı veya bir koşulun yerine getirilip getirilmediği kontrol edilir. Programlamada da bu karşılaştırmalar, karar vermek ve programın akışını kontrol etmek amacıyla kullanılır. Bu derste karşılaştırma işlemlerinin ne olduğu, C++ programlama dilinde nasıl çalıştıkları ve gerçek programlama örneklerinde nasıl uygulandıkları açıklanacaktır.

Programlamada karşılaştırma işlemleri, karar verme mekanizmasının temelini oluşturur. Bu işlemler aracılığıyla iki sayının eşit olup olmadığı, bir sayının diğerinden büyük veya küçük olup olmadığı ve değerlerin farklı olup olmadığı belirlenebilir. C++ dilinde altı

temel karşılaştırma operatörü bulunur: == (eşit), != (eşit değil), < (küçük), > (büyük), <= (küçük veya eşit), >= (büyük veya eşit). Bu operatörler her zaman mantıksal bir değer döndürür, yani true (doğru) veya false (yanlış) sonucu verir ve bu sonuçlar daha sonra koşullu yapılarda kullanılabilir.

ÖRNEK 1

```
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      int a = 5, b = 10;
7      bool sonuc = a < b;
8      cout << "a < b işleminin sonucu: " << sonuc << endl;
9      return 0;
10 }
```

C++ dilinde değerler karşılaştırılırken, bu değerlerin veri türü önemli bir rol oynar. Eğer tam sayılar karşılaştırılıyorsa, karşılaştırma doğrudan gerçekleştirilir. Ancak ondalıklı sayılar veya karakterler karşılaştırılıyorsa, doğruluğa ve ASCII değerlerine dikkat edilmelidir. Metinsel verilerin (string) karşılaştırılmasında, karakterlerin sıralamasına dayanan özel bir mantık kullanılır. Bu şekilde isimleri, notları veya gerçek hayattaki diğer verileri karşılaştıran programlar yazılabilir.

ÖRNEK 2

```
1  #include <iostream>
2  #include <string>
3  using namespace std;
4
5  int main()
6  {
7      string isim1 = "Ana";
8      string isim2 = "Bojan";
9      cout << "isim1 < isim2 mi? " << (isim1 < isim2) << endl;
10     return 0;
11 }
```

Çoğu zaman bir karşılaştırma işleminin sonucu, daha sonra programın diğer bölümlerinde kullanılan bool türündeki bir değişkende saklanır. Örneğin, karşılaştırmının sonucuna bağlı olarak farklı bir mesaj ekrana yazdırılabilir, programın çalışma akışı değiştirilebilir veya belirli bir işlem gerçekleştirilebilir. Mantıksal değerler, daha karmaşık koşullar oluşturmak için mantıksal operatörlerle de birleştirilebilir. Bu konu, bir sonraki dersin temelini oluşturur.

ÖRNEK 3



```
1 #include <iostream>
2 using namespace std;
3
4 int main()
5 {
6     int yil = 2025;
7     bool artikYilMi = (yil % 4 == 0);
8     cout << "Yıl artık yıl mı? " << artikYilMi << endl;
9     return 0;
10 }
```

Farklı değerler ve operatörlerle alıştırmak, karşılaştırmaların arkasındaki mantığın anlaşılması açısından önemlidir. Koşulların karmaşıklığı artırılarak programlamadaki sonraki aşamalara, koşullu yapıların (if/else), döngülerin ve mantıksal operatörlerin kullanımına hazırlık yapılır. Başlangıç aşamasında, operatörlerin doğru yazımını ve beklenen sonuçları öğrenmek için sayılar ve stringler arasında basit karşılaştırmalar yapmak en uygun yöntemdir.



SONUÇ

Karşılaştırma operatörleri, karar vermesi gereken her programın vazgeçilmez bir parçasıdır. Bu operatörler, değerler arasındaki ilişkilerin kontrol edilmesini sağlar ve koşullu yapıların temelini oluşturur. Alıştırmalar ve pratik örnekler aracılığıyla, programlamada mantıksal kararların nasıl alındığı konusunda anlayış geliştirilir.



TEKRARLAMA SORULARI:

1. Tüm karşılaştırma operatörlerini gerçek bir örnek içeren cümlelere dönüştür!
2. <, >, ==, != operatörlerini kullanarak -100 ile 100 arasındaki sayılarla alıştırmak yap!
3. İlk harfleri aynı olan stringleri karşılaştırdığında ne olduğunu kontrol et!

BİLGİ KONTROLÜ SORULARI VE ETKİNLİKLER



1. C++ dilinde == operatörü ne döndürür?
2. İki değerin farklı olup olmadığını kontrol etmek için hangi operatör kullanılır?
3. bool değişkeni nedir?
4. Bir karşılaştırma işlemi hangi değeri döndürür?
5. 15 sayısının 10 sayısından büyük olup olmadığını kontrol eden bir program yaz!
6. İki ismin eşit olup olmadığını kontrol eden bir program yaz!
7. Sayı 100'den küçükse ekrana «EVET» yazdıran bir program yaz!
8. İki stringi karşılaştıran ve sözlük sırasına göre hangisinin «daha küçük» olduğunu ekrana yazdıran bir program yaz!

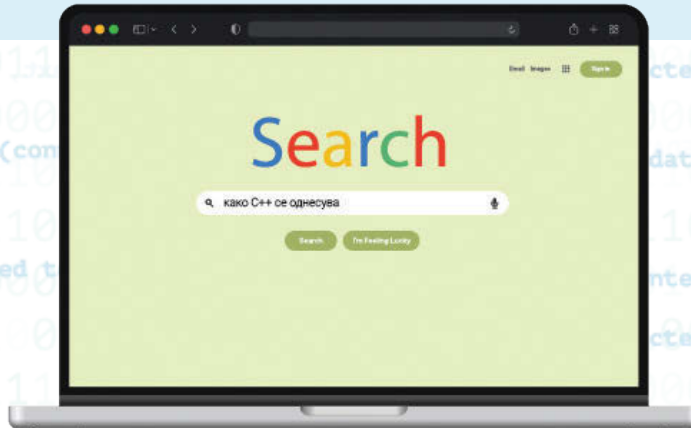
Öğrendiklerini Hatırla!

- Karşılaştırma operatörleri, değerleri karşılaştırmak için kullanılır.
- Her karşılaştırma ifadesi true veya false değerini döndürür.
- Sonuç, bool türündeki bir değişkende saklanabilir.
- Bu operatörler, koşullu yapıların temelini oluşturur.



DAHA FAZLA BİLGİ EDİNMEK İSTEYENLER İÇİN

C++ dilinde farklı türlerin (örneğin int ile double) karşılaştırılması sırasında nasıl davrandığını araştır! Karşılaştırmaları, bir sonraki derste öğreneceğimiz && (ve) ve || (veya) gibi mantıksal operatörlerle birleştiren bir program yazmayı dene!



Bu konuyu tamamladığında...

- ✓ Karşılaştırma operatörlerini (`==`, `!=`, `<`, `>`, `<=`, `>=`) kullanırsın,
- ✓ Bir ifadenin doğru veya yanlış olup olmadığını kontrol edersin,
- ✓ Program içinde basit koşullar yazarsın,
- ✓ Bir karşılaştırmadan boolean değer elde edersin.

**TEKRARLAMA SORULARI:**

1. «Küçük veya eşit» operatörü nasıl yazılır?
2. Bir sayının çift olup olmadığını kontrol eden bir program yaz!

**AŞAĞIDAKİ KONULARDA ŞUNLARI ÖĞRENECEKSİN:**

- Birden fazla koşulu mantıksal operatörleri kullanarak birleştirirsin,
- Mantıksal ifadelerin ne olduğunu ve nasıl çalıştığını açıklarsın,
- C++ dilinde `&&`, `||` ve `!` operatörlerini kullanırsın,
- İşlem önceliği kurallarını anlarsın,
- Kısa devre değerlendirmesinin (short-circuit) ne zaman uygulandığını belirlersin.

Programlamada, birden fazla koşulun aynı anda kontrol edilmesi gereken durumlarla sıkça karşılaşılır. Örneğin, bir sayının 10'dan büyük ve 20'den küçük olup olmadığı veya kullanıcının doğru şifreyi ya da doğru PIN kodunu girip girmediği kontrol edilebilir. Bu gibi durumlarda basit bir karşılaştırma işlemi yeterli değildir. Bunun yerine koşullar, mantıksal operatörler adı verilen yapılarla birleştirilir. Bu operatörler, birden fazla karşılaştırmaya bağlı olarak doğruluk veya yanlışlık değeri döndüren mantıksal ifadeler oluşturmayı sağlar.

Mantıksal operatörler, koşullarla çalışılan her programlama dilinde temel bir araçtır. C++ dilinde üç ana mantıksal operatör bulunur: mantıksal VE işlemi için `&&`, mantıksal VEYA işlemi için `||` ve DEĞİL (olumsuzlama) işlemi için `!`. Bu operatörler, basit koşulları daha karmaşık mantıksal ifadeler hâline getirmek için kullanılır. Örneğin, bir x sayısının 10 ile 20

arasında olup olmadığını kontrol etmek için $x > 10 \ \&\& \ x < 20$ ifadesi yazılır. Bu ifade, yalnızca her iki karşılaştırma da doğru olduğunda true değerini döndürür. Karşılaştırmalardan en az biri yanlışsa, tüm ifade yanlış olacaktır.

ÖRNEK 1

```
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      int sayi = 15;
6      if (sayi > 10 && sayi < 20) {
7          cout << "Sayı 10 ile 20 arasındadır." << endl;
8      }
9      return 0;
10 }
```

Koşullar birleştirilirken operatörlerin önceliğine dikkat etmek önemlidir. C++ dilinde mantıksal! operatörü en yüksek önceliğe sahiptir, ardından && gelir ve en son || değerlendirilir. Bu nedenle bir ifadede her üç operatör de kullanılırsa, önce olumsuzlama, ardından mantıksal VE işlemi, daha sonra mantıksal VEYA işlemi gerçekleştirilir. Belirsizliği önlemek için çoğu zaman parantezler kullanılır. Örneğin, $!(x < 10 \ || \ x > 20)$ ifadesi, yalnızca x sayısı 10 ile 20 aralığındaysa (10 ve 20 dâhil) true değerini döndürür. Parantezler, işlemlerin gerçekleşme sırası üzerinde kesin kontrol sağlar.

ÖRNEK 2

```
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      int x = 25;
6      if (!(x < 10 || x > 20)) {
7          cout << "x, 10 ile 20 arasındadır." << endl;
8      } else {
9          cout << "x aralığın dışındadır." << endl;
10     }
11     return 0;
12 }
```

C++ dilinde mantıksal operatörlerin önemli bir özelliği kısa devre değerlendirmesi (short-circuit evaluation) özelliğidir. Bu, && operatöründe ilk koşul yanlışsa, ifadenin tamamı zaten yanlış olacağı için ikinci koşulun hiç kontrol edilmemesi anlamına gelir. Benzer şekilde, || operatöründe ilk koşul doğruysa, ifadenin true olması için tek bir koşulun doğru olması yeterli olduğundan ikinci koşul kontrol edilmez. Bu özellik yalnızca programın verimliliğini artırmakla kalmaz, aynı zamanda aksi durumda hatalara neden olabilecek koşulların yazılmasına da olanak sağlar. Buna örnek olarak, bölme işlemi gerçekleştirilmeden önce değişkenin sıfırdan farklı olup olmadığının kontrol edilmesi verilebilir.

ÖRNEK 3



```
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     int sayi = 0;
6     if (sayi != 0 && 100 / sayi > 1) {
7         cout << "Bölüm 1'den büyüktür." << endl;
8     } else {
9         cout << "Bölme işlemi yapılmadı." << endl;
10    }
11    return 0;
12 }
```

Mantıksal ifadelerin birleştirilmesi, doğal dile yakın koşulların yazılmasını sağlar. Örneğin, kullanıcının reşit olup olmadığının ve kimlik kartına sahip olup olmadığının veya izin belgesine ya da refakate sahip olup olmadığının kontrol edilmesi gibi durumlar mantıksal operatörler aracılığıyla programlara aktarılır ve kodu daha anlaşılır hâle getirir. Ayrıca, döngülerde ve koşullarda mantıksal ifadelerin kullanılması, gerçek ve işlevsel programlar oluşturmak için büyük önem taşır. Bu ifadeler sayesinde, her otomasyonun temelinde bulunan mantık gerçekleştirilir.



SONUÇ

Mantıksal ifadeler, programların çalışmasının kesin ve koşullu bir şekilde kontrol edilmesi için gereklidir. Birden fazla koşulun birleştirilmesine ve kod içinde mantıklı kararların alınmasına olanak sağlar. &&, || ve ! operatörlerinin doğru kullanılması, ayrıca işlem önceliğine ve kısa devre değerlendirmesine dikkat edilmesi, güvenilir ve verimli programların temelini oluşturur.





DAHA FAZLA BİLGİ EDİNMEK İSTEYENLER İÇİN

Mantıksal operatörlerin karşılaştırma operatörleriyle nasıl birleştirildiğini ve if ile while yapılarında nasıl kullanıldığını araştır! Kullanıcı adı ve şifre ile giriş koşullarının uygulandığı bir program yaz ve koşulun bir bölümünün doğru olmaması durumunda ne olduğunu analiz et!



BİLGİ KONTROLÜ SORULARI VE ETKİNLİKLER



1. C++ dilindeki üç mantıksal operatör hangileridir?
2. Kısa devre değerlendirmesi (short-circuit evaluation) nedir?
3. Mantıksal operatörlerin işlem önceliği sırası nasıldır?
4. ! operatörü neden kullanılır?
5. Bir sayının pozitif ve çift olup olmadığını kontrol eden bir program yaz!
6. Kullanıcının 18 yaşında veya daha büyük olup olmadığını ya da izne sahip olup olmadığını kontrol eden bir program yaz!
7. Bir stringin boş olmadığını ve 3 harften fazla olduğunu kontrol eden bir program yaz!
8. Bir sayı için && ve || operatörlerini kullanarak 3 koşulu birleştiren bir program yaz!

Öğrendiklerini hatırla!

- Mantıksal operatörler &&, || ve !, birden fazla koşulu birleştirir.
- &&, yalnızca her iki ifade de doğru olduğunda true değerini döndürür.
- ||, en az bir ifade doğru olduğunda true değerini döndürür.
- !, koşulun doğruluk değerini tersine çevirir.
- Verimliliği artırmak için kısa devre değerlendirmesi uygulanır.

Bu konuyu tamamladığında...

- ✓ Karşılaştırma operatörlerini kullanırsın,
- ✓ Ekranı metin yazdırırsın,
- ✓ Aritmetik işlemler gerçekleştirirsin.

**TEKRARLAMA SORULARI:**

1. Koşulun olumsuzlanması için ! operatörünü kullanan bir program yaz!
2. Birden fazla && ve || operatörünün parantez olmadan birleştirilmesi durumunda ne olduğunu kontrol et!
3. Gerçek durumlarla ilgili kendi örneklerini yaz (örneğin, okula kabul koşulu)!

**AŞAĞIDAKİ KONULARDA ŞUNLARI ÖĞRENECEKSİN:**

- İki seçenekli seçimin ne anlama geldiğini ayırt edersin,
- Programlamada seçim yapısının ne zaman kullanılması gerektiğini belirlersin,
- Mantıksal bir koşulun kodun çalışması üzerindeki etkisini anlarsın,
- C++ dilinde if-else yapısını kullanarak örnekler oluşturursun,
- Farklı mantıksal koşulların sonuçlarını analiz edersin.

Günlük hayatta sık sık iki seçenek arasında kararlar verilir. Bu tür durumlarla programlama dünyasında da karşılaşılır. İki seçenekli seçim yapısı, bilgisayarın bir koşula bağlı olarak karar vermesini ve iki olası komut bloğundan birini çalıştırmasını sağlar. Bu derste, programcıya kararları uygulama olanağı veren temel mantıksal yapı ele alınmaktadır. Bu yapı

if-else yapısı olarak adlandırılır ve programın akışının koşullu yönetiminin bir parçasıdır. Bu tür koşulların kullanılması, kodu daha esnek hâle getirir ve gerçek problemlerin çözümüne uygun olmasını sağlar.

İki seçenekli seçim yapısı, programlamadaki en basit ancak çok önemli kontrol mekanizmalarından biridir. Bu yapı sayesinde programın iki alternatif arasında seçim yapması sağlanır. Bu işlem, bir mantıksal koşulun sağlanıp sağlanmadığını kontrol eden if anahtar kelimesi ile gerçekleştirilir. Koşul doğruysa bir komut bloğu çalıştırılır, aksi durumda ise else ile belirtilen ikinci komut bloğu çalıştırılır. Bu kontrol yöntemi, programın verilen verilere veya durumlara bağlı olarak farklı tepkiler vermesini sağlar. Programlardaki kararlar, önceki derste ele alınan karşılaştırma ve mantıksal ifadeler temel alınarak oluşturulur.

ÖRNEK 1

```
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      int sayi;
6      cout << "Bir sayı girin: ";
7      cin >> sayi;
8
9      if (sayi > 0)
10         cout << "Sayı pozitifdir." << endl;
11     else
12         cout << "Sayı pozitif değildir." << endl;
13     return 0;
14 }
```

Bu örnekte iki seçenek arasında bir seçim gösterilmektedir: sayının sıfırdan büyük olup olmadığı. Koşulun sonucuna bağlı olarak uygun bir mesaj ekrana yazdırılır.

Basit karşılaştırma koşullarının yanı sıra, iki seçenekli seçim yapısında mantıksal operatörler kullanılarak birden fazla koşul birleştirilebilir. Bu olanak, koşulu daha karmaşık hâle getirir ve programın verdiği kararlar üzerinde daha ayrıntılı kontrol sağlar. Örneğin, kullanıcı tarafından girilen bir değer belirli bir aralıkta olup olmadığı veya iki bağımsız koşulun aynı anda sağlanıp sağlanmadığı kontrol edilebilir.

ÖRNEK 2



```
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     int puan;
6     cout << "Puanı girin (0-100): ";
7     cin >> puan;
8
9     if (puan >= 51 && puan <= 100)
10         cout << "Geçti." << endl;
11     else
12         cout << "Geçemedi." << endl;
13     return 0;
14 }
```

Örnekte, girilen puanların geçmek için gereken minimum sınır içinde olup olmadığı kontrol edilmektedir. Bir aralık belirlemek için mantıksal && operatörünün kullanıldığına dikkat edilmelidir.

Sıkça sorulan bir soru, birden fazla if-else ifadesinin birleştirilip birleştirilemeyeceğidir. Cevap evettir – karmaşık kararların gerektiği durumlarda if yapıları iç içe kullanılabilir veya else if yapısı kullanılabilir. Ancak her zaman kodun açık ve okunabilir bir yapıda tutulması önemlidir, çünkü bu, programın mantığının daha kolay bakımını ve anlaşılmasını sağlar.

ÖRNEK 3



```
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     int sayi;
6     cout << "Bir tam sayı girin: ";
7     cin >> sayi;
8
9     if (sayi % 2 == 0)
10         cout << "Sayı çifttir." << endl;
11     else
12         cout << "Sayı tektir." << endl;
13     return 0;
14 }
```

Örnekteki program, bir sayının çift mi yoksa tek mi olduğunu kontrol etmektedir. Bu tür kontroller, programlamada sayı türünün analiz edilmesi veya belirli bir işlemin gerçekleştirilip gerçekleştirilmeyeceğinin belirlenmesi sırasında sıkça kullanılır.

Bu konuyu tamamladığında...

- ✓ Aritmetik ifadeleri hesaplırsın ve değışkenleri kullanırsın.
- ✓ cin ve cout komutlarını kullanırsın.
- ✓ Karşılaştırma operatörlerini ve mantıksal değerleri birbirinden ayırt edersin.

BİLGİ KONTROLÜ SORULARI VE ETKİNLİKLER



1. if-else yapısı neyi ifade eder?
2. Koşul sağlanmadığında kodun hangi bölümü çalıştırılır?
3. Mantıksal operatörler if-else içinde nasıl kullanılır?
4. broj % 2 == 0 ne anlama gelir?
5. Birden fazla if yerine ne zaman else if kullanmak daha uygundur?
6. Girilen sayının negatif olup olmadığını kontrol eden bir program yaz!
7. Bir öğrencinin 60 puandan fazla alması durumunda sınavı geçip geçmediğini kontrol eden bir program yaz!
8. Sıcaklığın normal aralıkta (36 ile 37,5 derece arasında) olup olmadığını kontrol eden bir program yaz!
9. Bir harfin sesli harf olup olmadığını kontrol eden bir program yaz!
10. Yılın artık yıl olup olmadığını kontrol eden bir program yaz!

Öğrendiklerini Hatırla!

- İki seçenekli seçim yapısı (if-else), mantıksal bir koşula bağlı olarak farklı kodların çalıştırılmasını sağlar. Bu yapı, program akışının kontrolünde kullanılan temel mekanizmalardan biridir ve kararları ifade etmek için kullanılır. Bu yapı sayesinde program, farklı durumlara ve girilen değerlere uyarlanabilir. Karşılaştırma operatörleri ve mantıksal ifadelerin kullanılması, güçlü koşullar oluşturmayı sağlar. if-else yapısının uygulanması, kodu gerçek problem senaryolarında daha dinamik ve işlevsel hâle getirir.

3.22

if-else Yapısı için Alıştırma Soruları



KOLAY ALIŞTIRMALAR (1 – 7)

1. Girilen sayının 100'den büyük olup olmadığını kontrol eden bir program yaz!
2. Verilen sayının negatif olup olmadığını kontrol eden bir program yaz!
3. Girilen sayının 5'e bölünüp bölünmediğini kontrol eden bir program yaz!
4. Kullanıcının 18 yaşını doldurup doldurmadığını kontrol eden bir program yaz!
5. Girilen iki tam sayının eşit olup olmadığını kontrol eden bir program yaz!
6. Kullanıcı «cumartesi» veya «pazar» girerse «Bugün hafta sonu», aksi durumda «İş günü» yazdıran bir program yaz!
7. Puanı 90'dan yüksekse öğrencinin 5 notunu alıp almadığını kontrol eden bir program yaz!



ORTA DÜZEY ALIŞTIRMALAR (8 – 14)

8. Girilen sayının hem 3'e hem de 7'ye bölünüp bölünmediğini belirleyen bir program yaz!
9. Kullanıcının girdiği sayının 10 ile 99 arasındaki aralıkta olup olmadığını kontrol eden bir program yaz!
10. Doğru şifre girilip girilmediğini kontrol eden bir program yaz!
11. (Şifrenin «tajna123» olduğunu varsayalım.)
12. Bir kişinin oy kullanma hakkına sahip olup olmadığını kontrol eden bir program yaz!
13. (Kişinin en az 18 yaşında olması ve vatandaş olması gerekir.)
14. Sayı pozitif ve çift ise «çift ve pozitif», aksi durumda «şartları sağlamıyor» yazdıran bir program yaz!
15. Üç kenarın bir üçgen oluşturup oluşturamayacağını kontrol eden bir program yaz!
16. Puan 90'dan büyükse «mükemmel», 70 ile 90 arasındaysa «iyi», aksi durumda «yeterli» yazdıran bir program yaz!



ZOR ALIŞTIRMALAR (15 – 20)

18. İki girilen sayıdan hangisinin daha büyük olduğunu belirleyen bir program yaz!
19. Girilen sıcaklığın normal sınırlar içinde olup olmadığını kontrol eden bir program yaz!
20. (36,0 ile 37,5 derece arasında olup olmadığını kontrol et.)
21. Girilen harfin büyük harf mi yoksa küçük harf mi olduğunu kontrol eden bir program yaz!
22. Girilen yılın artık yıl olup olmadığını kontrol eden bir program yaz!
23. Girilen sayının üç basamaklı ve çift olup olmadığını kontrol eden bir program yaz!
24. Girilen kullanıcı adı ve şifrenin doğru olup olmadığını kontrol eden bir program yaz!
25. (Örneğin: kullanıcı adı = «admin», şifre = «1234».)!

Bu konuyu tamamladığında...

- ✓ Basit koşullu ifadeler oluşturursun,
- ✓ Karar vermek için if ve else yapılarını kullanırsın,
- ✓ Girdi ve çıktı kullanan temel programlar oluşturursun.

**TEKRARLAMA SORULARI:**

1. 10 ile 20 arasındaki bir sayının kontrol edilmesi için ifade nasıl yazılır?
2. Örnekleri yeniden yaz ve farklı bir çıktı elde etmek için değerleri değiştir!

**AŞAĞIDAKİ KONULARDA ŞUNLARI ÖĞRENECEKSİN:**

- Bir ifadenin başka bir ifade içinde "iç içe" bulunmasının ne anlama geldiğini anlarsın,
- Basit koşullar ile iç içe koşulları birbirinden ayırt edersin,
- Daha karmaşık problemleri çözmek için iç içe koşulları kullanırsın,
- Birden fazla iç içe yapı kullanan programlar yazarsın,
- İç içe koşulların nerelerde dikkatli kullanılması gerektiğini anlarsın.

Programlamada koşullu yapılar oluşturma sürecinde, çoğu zaman bir kontrolün başka bir kontrolün içinde yapılması gereken durumlarla karşılaşılır. İç içe koşullar (nested conditions) olarak adlandırılan bu yapı, karmaşık mantıksal kararlar oluşturmayı sağlayan güçlü bir kavramdır. İç içe koşul, bir koşulun başka bir koşulun içinde yer almasıdır ve çok katmanlı mantıkla programlama yapılmasına olanak tanır. Bu tür yapıların kullanımı, gerçek hayattaki problemlerin modellenmesinde sık görülür. Çünkü bazı işlemler yalnızca birden fazla koşul belirli bir sırayla sağlandığında gerçekleştirilmelidir. İç içe koşullar, if içinde if şeklinde veya bir yapının başka bir yapıyı içerdiği if ve else if kombinasyonları şeklinde kullanılabilir.

İç içe koşullar, bir kararın daha önce verilen başka bir karara bağlı olduğu durumlarda kullanılır. Pratik örneklerde, bir sayının pozitif olup olmadığının kontrol edilmesi ve ardından çift olup olmadığının incelenmesi sık karşılaşılan bir durumdur. Bu iki kontrol birbirinden bağımsız değildir, çünkü sayı pozitif değilse ikinci kontrolün yapılmasının anlamı yoktur.

Tam olarak bu tür durumlar iç içe koşullar ile çözülür. Önce bir koşul yazılır ve bu koşulun gövdesinin içine ikinci bir koşul yerleştirilir. Böylece ikinci koşul yalnızca ilk koşul doğru olduğunda kontrol edilir. Bu yöntem gereksiz veya mantıksız kontrolleri önler. Bu şekilde mantığın düzenlenmesi, kodun daha anlaşılır olmasını ve programın çalışma akışı üzerinde daha iyi kontrol sağlanmasını mümkün kılar.

```
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      int sayi;
6      cin >> sayi;
7      if (sayi > 0) {
8          if (sayi % 2 == 0) {
9              cout << "Sayı pozitif ve çifttir." << endl;
10         }
11     }
12     return 0;
13 }
```

İç içe koşullar yalnızca iki seviyeden oluşmak zorunda değildir. Özellikle farklı özelliklerin analiz edildiği veya birden fazla değişkenin karşılaştırıldığı durumlarda, daha fazla katman şeklinde düzenlenebilirler. Ancak ne kadar fazla koşul iç içe yerleştirilirse, kodun okunması ve bakımı o kadar zorlaşır. Bu nedenle iç içe koşullar ölçülü kullanılmalı ve yalnızca mantıksal olarak gerekli olduğu durumlarda tercih edilmelidir. İç içe koşullar, kontrollerin sırasının önemli olduğu durumlarda kullanılır. Yani bazı koşulların yalnızca önceki koşullar sağlandığında kontrol edilmesi gerekir. Örneğin, bir öğrencinin sınavı geçmek için yeterli puana sahip olup olmadığı ve ardından yeterli ders saatine katılıp katılmadığının kontrol edilmesi gerekiyorsa, ikinci kontrolün yalnızca ilk koşul sağlandığında yapılması mantıklıdır.

```
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      int puan, katilim;
6      cin >> puan >> katilim;
7      if (puan >= 50) {
8          if (katilim >= 75) {
9              cout << "Öğrenci başarılı." << endl;
10         }
11     }
12     return 0;
13 }
```

Gerçek problemlerde çoğu zaman farklı senaryolar arasında seçim yapmak gerekir. Bu durumda iç içe ifadeler else dallarıyla da birleştirilebilir. Böylece hangi koşulların sağlandığına bağlı olarak farklı davranışlar oluşturulabilir. Bu kullanım, iç içe koşulların gücünü artırır, ancak mantık hatalarının oluşmaması için kodun dikkatli bir şekilde düzenlenmesini gerektirir. Çok sık olarak if-else yapısı, birden fazla iç içe koşulla birlikte kullanılır ve böylece ağaç yapısına benzeyen bir mantık oluşturulur. Bu tür durumlar, örneğin yaş grubu ve üyelik durumu gibi birden fazla kritere bağlı olarak fiyat seçimi yapılırken yaygın olarak kullanılır.

```

1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      int yas;
6      bool uyeMi;
7      cin >> yas >> uyeMi;
8      if (yas >= 18) {
9          if (uyeMi) {
10             cout << "Ücret 100 denardır." << endl;
11          } else {
12             cout << "Ücret 150 denardır." << endl;
13          }
14      } else {
15          cout << "Ücret 50 denardır." << endl;
16      }
17      return 0;
18 }

```

İç içe koşulların avantajlarının yanında bazı zorlukları da vardır. Aşırı kullanımı, kodun okunmasını zorlaştırabilir ve hata yapma olasılığını artırabilir. Bu nedenle iyi bir programlama uygulaması olarak, iç içe koşullar yalnızca mantıksal olarak gerekli olduğunda ve daha basit bir yapıyla değiştirilemediğinde kullanılmalıdır. Eğer kodun çok uzun veya takip edilmesi zor hâle geldiği fark edilirse, mantığın bazı bölümlerinin ayrı fonksiyonlara ayrılması düşünülebilir. Bu yaklaşım, kodun tekrar kullanılmasını ve programın daha düzenli olmasını sağlar. Ayrıca kodun okunabilirliğine dikkat edilmelidir. Bunun için uygun girintileme (tab kullanımı) yapılmalı ve hangi koşulun nerede başlayıp bittiğini açıklayan yorum satırları kullanılmalıdır.

BİLGİ KONTROLÜ SORULARI VE ETKİNLİKLER

1. İç içe ifade (nested if) nedir?
2. Gerçek bir durumda iç içe if ne zaman kullanılır?
3. Aşırı iç içe kullanmanın riskleri nelerdir?
4. İç içe koşullar kullanırken kodu nasıl daha okunabilir hâle getirebiliriz?
5. Bir sayının 3'e bölünüp bölünmediğini, ardından çift olup olmadığını kontrol eden bir program yaz!
6. Bir öğrenci için puan ve devam durumunu girerek geçip geçmediğine karar veren bir program yaz!
7. Yaş grubuna ve üyelik durumuna göre bilet fiyatını hesaplayan bir program yaz!
Bir öğrencinin üstün başarıya sahip olup olmadığını ve ödül alıp almadığını kontrol eden bir program yaz!

Öğrendiklerini Hatırla!

- İç içe koşullar (nested if), başka koşulların içine yerleştirilen koşullardır.
- Bu yapılar, mantığın yalnızca belirli şartlar altında ek kontroller yapılmasını gerektirdiği durumlarda kullanılır.
- Aşırı iç içe koşul kullanımı, kodun okunmasını ve bakımını zorlaştırır.
- Kod yazarken, hangi koşulun hangi bloğa ait olduğu her zaman açık olmalıdır.
- İç içe koşullar, birbiriyle ilişkili birden fazla kritere göre karmaşık kararlar verilmesini sağlar.

3.24

İç içe if-else Yapısı için Alıştırmalar



KOLAY AŞTIRMALAR

1. Girilen sayının pozitif ve 5'e bölünüp bölünmediğini kontrol eden bir program yaz.
2. 10-a bölünüp bölünmediğini, bölünüyorsa 100'den büyük olup olmadığını kontrol eden program yaz.
3. Bir sayının tek olup olmadığını ve eğer tekse tek basamaklı olup olmadığını kontrol eden bir program yaz.
4. Bir sayının 4'e bölünüp bölünmediğini, ardından 6'ya da bölünüp bölünmediğini kontrol eden bir program yaz!
5. Bir sayının pozitif ve 3'e bölünebilir olup olmadığını kontrol eden bir program yaz!
6. Bir harfin sesli harf olup olmadığını ve eğer sesli harfse büyük harf olup olmadığını kontrol eden bir program yaz.



ORTA DÜZEY ALIŞTIRMALAR

7. Sıcaklığın 0'ın altında olup olmadığını ve eğer öyleyse yağmur yağıp yağmadığını kontrol eden bir program yaz!
8. Kişinin reşit olup olmadığını ve eğer reşitse sürücü belgesi olup olmadığını kontrol eden bir program yaz!
9. Yılın artık yıl olup olmadığını ve ardından çift sayı olup olmadığını kontrol eden bir program yaz!
10. İsmi büyük harfle başlayıp başlamadığını ve ardından 5 harften fazla olup olmadığını kontrol eden bir program yaz!
11. Öğrencinin liseye kayıtlı olup olmadığını ve ardından not ortalamasının 4'ten büyük olup olmadığını kontrol eden bir program yaz!
12. Girilen sayının iki basamaklı olup olmadığını ve eğer öyleyse son rakamının çift olup olmadığını kontrol eden bir program yaz!
13. Günün cumartesi olup olmadığını ve eğer öyleyse sıcaklığın 25 dereceden yüksek olup olmadığını kontrol eden bir program yaz.



ZOR ALIŞTIRMALAR

14. Öğretmen, hangi öğrencilerin özel övgüyü hak ettiğini kontrol etmek istiyor. Her öğrenci için testlerden aldığı puan ve yaptığı ödev sayısı bilinmektedir. Öğrencinin 80 puandan fazla alıp almadığını, ardından tüm ödevlerini yapıp yapmadığını kontrol eden bir program yaz!
15. Dönem sonunda öğretmen, hangi öğrencilerin olumlu davranış notu alabileceğini belirlemek istiyor. Her öğrenci için ders puanı ve mazeretsiz devamsızlık sayısı bilinmektedir. Öğrencinin 60 puandan fazla alıp almadığını, eğer aldıysa 5'ten az mazeretsiz devamsızlığı olup olmadığını kontrol eden bir program yaz!
16. Sürücü sınav kayıt bürosunda, kişilerin şartları sağlayıp sağlamadığı kontrol edilmektedir. Her kişinin yaşı ve kimlik kartına sahip olup olmadığı bilinmektedir. Kişinin 18 yaşında veya daha büyük olup olmadığını, ardından kimlik kartı olup olmadığını kontrol eden bir program yaz!

Bu konuyu tamamladığında...

- ✓ Tek ve çok seçenekli koşullar arasındaki farkı ayırt etmek,
- ✓ İki seçenekli seçim yapılarını (if-else) kullanmak,
- ✓ Programlarda mantıksal ifadeleri tanımak,
- ✓ Karşılaştırma operatörlerini kullanmak,
- ✓ Koşullar kullanarak basit programlama problemlerini çözmek.

**TEKRARLAMA SORULARI:**

1. If ve else çoklu katmanlarda nasıl birbirine bağlanır?
2. Girilen sayının pozitif ve 100'den büyük olup olmadığını kontrol eden bir program yaz!

**AŞAĞIDAKİ KONULARDA ŞUNLARI ÖĞRENECEKSİN:**

- İki seçenekli koşul yapısından çok seçenekli seçim yapısını ayırt etmek,
- if - else if - else yapısını uygulamak,
- switch yapısının ne zaman uygun olduğunu tanımak,
- Programlarda çoklu koşulları uygulamak,
- İkiden fazla alternatifte sahip program yazmak,
- Dallanma sırasında mantıksal hatalardan kaçınmak.

Günlük hayatta problemler çözülürken veya program yazarken, çoğu zaman birden fazla seçenek arasından seçim yapmamız gereken durumlarla karşılaşırız. Örneğin, sıcaklık 0 °C'nin altındaysa kışlık mont giyilir, 0 ile 20 °C arasındaysa sonbaharlık palto, 20 °C'nin üzerindeyse ince bir gömlek giyilir. Bu, birden fazla seçenek arasından seçim yapmayı ifade eder. Programlamada bu tür davranışlar, iki seçenekli seçim yapısının mantıksal devamı olan birden fazla seçenekten seçim yapısı ile modellenir. Sadece iki yürütme yolu sağlayan if-else yapısından farklı olarak, if - else if - else veya switch yapısı birden fazla alternatifin işlenmesini sağlar. Bu tür bir mantık, programlamada ve farklı senaryoların

modellenmesinde büyük önem taşır. Bu derste, bu yapıların nasıl ve ne zaman kullanılacağı, kullanımları sırasında hataların nasıl önleneceği ve örnekler aracılığıyla dallanmış program akışlarının doğru uygulanmasına yönelik anlayışın nasıl geliştirileceği ele alınacaktır.

Gerçek programlama durumlarında, çoğu zaman birden fazla farklı koşulun kontrol edilmesi gerekir ve bu koşullardan yalnızca birinin sağlanması beklenir. Bu gibi durumlarda if - else if - else yapısı kullanılır. Bu yapı, koşulların sıralı olarak kontrol edilmesini sağlar. Program, koşulu doğru olan ilk bloğa girer ve diğer koşullar göz ardı edilir. Bu yapının, koşulların birbirini dışladığı, yani yalnızca bir dalın çalıştırılması gereken durumlarda kullanıldığına dikkat edilmelidir. Bu yaklaşım, açık ve düzenli bir kod yapısı sağlar, aynı zamanda programın okunabilirliğini ve sürdürülebilirliğini artırır.

```
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      int sayi;
6      cout << "Bir sayı girin: ";
7      cin >> sayi;
8
9      if (sayi > 0)
10         cout << "Sayı pozitifdir." << endl;
11     else if (sayi < 0)
12         cout << "Sayı negatiftir." << endl;
13     else
14         cout << "Sayı sıfırdır." << endl;
15     return 0;
16 }
```

Örnekte, sayının pozitif, negatif veya sıfır olup olmadığını kontrol etmek için if - else if - else yapısı kullanılmaktadır. Bu koşullardan yalnızca biri gerçekleşebilir ve program uygun dalı seçerek diğer mantık yapılarını yok sayar.

if - else if - else yapısı güçlü olmasına rağmen, bazen kontroller aynı değişkenin değerlerine göre yapılır. Bu gibi durumlarda daha basit ve etkili bir alternatif olarak switch yapısı kullanılır. Bu yapı, bir tam sayı veya karakter değerine bağlı olarak bir kod bloğunun seçilmesine olanak sağlar.

Buradaki temel öge case ifadesidir. case, bir seçeneği temsil ederken, hiçbir case eşleşmediğinde son seçenek olarak default kullanılır. Bu yapı, özellikle aynı değişken üzerinde çok sayıda koşul kontrol edildiğinde kodun okunabilirliğini artırır.

```
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      int gun;
6      cout << "1 ile 7 arasında bir sayı girin: ";
7      cin >> gun;
8
9      switch (gun) {
10         case 1: cout << "Pazartesi"; break;
11         case 2: cout << "Salı"; break;
12         case 3: cout << "Çarşamba"; break;
13         case 4: cout << "Perşembe"; break;
14         case 5: cout << "Cuma"; break;
15         case 6: cout << "Cumartesi"; break;
16         case 7: cout << "Pazar"; break;
17         default: cout << "Geçersiz gün!";
18     }
19     return 0;
20 }
```

Bu örnekte, haftanın günlerinden seçim yapmak için switch yapısı kullanılmaktadır. Bu, bir değişkenin değerine göre seçim yapıldığında ve her değer için uygun bir işlemin gerçekleştirildiği durumlara iyi bir örnektir.

Çok sayıda koşulun kontrol edildiği durumlarda, koşulların sırasına dikkat etmek ve çakışmaları önlemek önemlidir. Öncelikle en sınırlı (en kesin) koşullar, ardından daha geniş koşullar kontrol edilmelidir. Ayrıca sık yapılan bir hata, else bölümünün kullanılmamasıdır. Bu durum, birden fazla bloğun beklenmeyen şekilde çalışmasına neden olabilir. Bu nedenle, iyi belgelenmiş ve okunabilir kod kullanılması önerilir. Bunun yanında, koşullarda kullanılan değişkenler, mantıksal tutarlılığı sağlayacak ve hataların kolayca giderilmesine yardımcı olacak şekilde seçilmelidir. Koşulların dikkatli bir şekilde düzenlenmesiyle mantıksal hata riskleri azaltılır ve daha açık bir program akışı elde edilir.

```
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     int puan;
6     cout << "Puanı girin (0-100): ";
7     cin >> puan;
8
9     if (puan >= 90)
10         cout << "Not: 5" << endl;
11     else if (puan >= 75)
12         cout << "Not: 4" << endl;
13     else if (puan >= 60)
14         cout << "Not: 3" << endl;
15     else if (puan >= 50)
16         cout << "Not: 2" << endl;
17     else
18         cout << "Not: 1" << endl;
19     return 0;
20 }
```

Örnek, puanlara göre birden fazla not arasından seçim yapılmasını göstermektedir. Koşulların en yüksek değerden en düşük değere doğru kesin bir sırayla düzenlendiğine dikkat edilmelidir. Böylece doğru sınıflandırma sağlanır.

if - else if - else ve switch yapılarının doğru kullanımının yanı sıra, bu yapıların her birinin ne zaman kullanılması gerektiğini bilmek de büyük önem taşır. Örneğin, birçok farklı sabit değere sahip int türündeki bir değer kontrol edilirken switch yapısı daha basit ve daha hızlı çalışır. Buna karşılık, koşullar yalnızca değerlere değil, ifadelere dayandığında if - else yapısı kullanılır. Bu seçimler yalnızca söz dizimiyle ilgili değildir, programın kalitesini, hızını ve bakım kolaylığını doğrudan etkiler. Bu ayrıntıların anlaşılması, programlama öğrenen herkes için temel bir öneme sahiptir.

BİLGİ KONTROLÜ SORULARI VE ETKİNLİKLER



1. if - else if - else yapısı nedir ve ne zaman kullanılır?
2. if ve switch yapıları arasındaki farkı açıklayın!
3. Koşulların en kesin olandan en genel olana doğru sıralanması neden önemlidir?
4. if yerine switch kullanılması ne zaman önerilir?
5. Girilen ay numarasına göre hangi mevsim olduğunu gösteren bir program yaz!
6. Puan sayısına (1 – 100) göre öğrenci hakkında açıklama yapan bir program yaz!
7. Not kontrolü yaparak mesaj gösteren („Mükemmel“, „İyi“ vb.) bir program yaz!
8. Girilen taşıma türüne göre ulaşım aracını gösteren bir program yaz (1 – Otomobil, 2 – Uçak, 3 – Tren)!

Öğrendiklerini Hatırla!

- if - else if - else yapısı, birden fazla seçenek arasından yalnızca birinin seçilmesi gerektiğinde kullanılır.
- switch yapısı, bir değişkenin kesin değerine göre seçim yapılmasını sağlar.
- Koşullar her zaman dikkatli ve mantıklı bir şekilde düzenlenmelidir.
- if - else if - else yapısında yalnızca bir kod bloğu çalıştırılır.
- switch yapısında, bir sonraki case bölümünün çalışmasını önlemek için her case ifadesi break ile sonlandırılmalıdır.



DAHA FAZLA BİLGİ EDİNMEK İSTEYENLER İÇİN

Temel yapıların yanı sıra, ileri programlamada iç içe geçmiş switch ifadeleri ve mantıksal operatörlerle (&&, ||) if koşullarının birleştirilmesi de kullanılmaktadır. Bazı programlama dillerinde, daha karmaşık dallanma seçimlerine olanak sağlayan match veya pattern matching gibi yapılar da bulunmaktadır. Ayrıca fonksiyonel programlamada seçimler çoğunlukla veri yapıları olarak modellenir ve bu durum daha yüksek düzeyde soyutlama sağlar.





KOLAY ALIŞTIRMALAR

1. Girilen sayının pozitif, negatif veya sıfır olup olmadığını kontrol eden bir program yaz!
2. Girilen 1 ile 7 arasındaki sayıya göre haftanın gününü gösteren bir program yaz!
3. Öğrencinin 1, 2, 3, 4 veya 5 notuna göre uygun açıklamayı yazdıran bir program yaz!
4. Girilen sıcaklığa (°C) göre kıyafet türünü gösteren bir program yaz!
5. Girilen 1 ile 12 arasındaki sayıya göre ayın adını yazdıran bir program yaz!
6. Girilen eğitim seviyesine göre mesaj yazdıran bir program yaz (1 – ilköğretim, 2 – ortaöğretim, 3 – yükseköğretim)!
7. Depodaki su seviyesini yüzde değerine göre açıklayan bir program yaz (düşük, orta, yüksek)!
8. GMT saat farkına göre zaman dilimini belirleyen bir program yaz!
9. Yaşa göre içecek önerisi veren bir program yaz (14 yaş altı – meyve suyu, 18 yaşına kadar – alkolsüz içecek, 18 yaş üstü – kahve)!
10. Hesap türünü gösteren bir program yaz (1 – elektrik, 2 – su, 3 – telefon)!



ORTA DÜZEY ALIŞTIRMALAR

11. Mesafeye göre ulaşım ücretini hesaplayan bir program yaz (10 km'den az, 50 km'ye kadar, 50 km'den fazla)!
12. Girilen sayıya göre müzik türünü sınıflandıran bir program yaz (1 – rock, 2 – caz, 3 – pop, 4 – klasik)!
13. AQI değerine göre hava kalitesi açıklamasını belirleyen bir program yaz (0 – 50 – **mükemmel**, 51 – 100 – **iyi vb.**)!
14. Kullanıcının rolüne göre kullanıcı türünü yazdıran bir program yaz (admin, misafir, kayıtlı kullanıcı)!
15. Girilen sayıya göre trafik ışığının rengini belirleyen bir program yaz (1 – kırmızı, 2 – sarı, 3 – yeşil)!
16. Gelir kategorisine göre vergi hesaplayan bir program yaz (düşük, orta, yüksek)!
17. Okul laboratuvarında elektronik ders kitapları için kullanılan tabletler test edilmektedir. Her tablet için pil doluluk yüzdesi (0 ile 100 arasında) ölçülmektedir. Teknik görevli, pil durumunun otomatik olarak gösterilmesini istemektedir:
0 ile 20 % arası – «kritik»
21 ile 60 % arası – «orta»
61 ile 100 % arası – «dolu»
Pil yüzdesini okuyup bu aralıklara göre uygun pil durumu mesajını yazdıran bir program yaz!

Bu konuyu tamamladığında...

- ✓ Mantıksal ve karşılaştırma ifadelerini ayırt edersin,
- ✓ C++ dilinde seçim yapılarını kullanırsın,
- ✓ İki değer arasında karşılaştırma yapmanın ne anlama geldiğini anlarsın,
- ✓ Bir işlemin ne zaman tekrarlanması gerektiğini fark edersin.

**TEKRARLAMA SORULARI:**

1. Çoklu koşullar kullanılırken oluşabilecek hatalar nelerdir?
2. Girilen 1 ile 7 arasındaki gün numarasına göre günün adını gösteren bir program yaz!

**AŞAĞIDAKİ KONULARDA ŞUNLARI ÖĞRENECEKSİN:**

- Döngüyü diğer programlama yapılarından ayırt edersin,
- Döngünün ne olduğunu ve ne zaman kullanıldığını açıklarsın,
- Programlama dilindeki farklı döngü türlerini tanırsın,
- Komutların neden ve ne zaman tekrarlandığını anlarsın,
- Tekrarlama yapılarıyla basit programlar yazarsın,
- Döngünün bir kez, birden fazla kez veya hiç çalışmayacağını analiz edersin.

Birçok problem çözme durumunda, belirli işlemlerin birden fazla kez tekrarlanmasına ihtiyaç duyulur. Bir dizideki elemanların toplamını hesaplama, menü gösterme veya şifre doğrulama gibi işlemlerde bazı komutların birden fazla kez çalıştırılması gerekir. Programlamada bu durum, programın belirli bölümlerinin belirli bir koşul sağlandığı sürece tekrar tekrar çalıştırılmasını sağlayan döngüler aracılığıyla gerçekleştirilir.

Döngüler, programların akış kontrolündeki temel ve en önemli yapılardan biridir. Döngülerin mantığını anlamak ve doğru kullanmak, programlamayı öğrenmede önemli bir adımdır. Döngüler sayesinde daha esnek, daha kısa ve daha verimli programlar oluşturulabilir.

Programlamada döngü olarak adlandırılan yapı, belirli bir koşul doğru olduğu sürece aynı kod bloğunun birden fazla kez çalıştırılmasını sağlayan bir yöntemdir. Sıralı çalıştırmada her komut yalnızca bir kez ve belirli bir sırayla yürütülürken, döngü sayesinde programın bir bölümü «geri dönerek» aynı kodu tekrar çalıştırabilir. Bu tekrar işlemi önceden belirlenmiş olabilir (örneğin 10 kez) veya bir koşula bağlı olabilir, yani belirli bir duruma ulaşılan kadar devam edebilir. Birkaç farklı döngü türü bulunmaktadır. Her birinin kendine özgü kullanım amacı ve özellikleri vardır, ancak hepsinin ortak amacı aynıdır, komutların kod içinde birçok kez elle tekrar yazılmasını önlemek için komutların tekrarlanmasını sağlamaktır. while döngüsünün yapısı şu şekildedir:

```
while (koşul) {  
    // çalıştırılacak komutlar  
}
```

ÖRNEK 1



```
1  #include <iostream>  
2  using namespace std;  
3  
4  int main() {  
5      int i = 0;  
6      while (i < 5) {  
7          cout << "Merhaba!" << endl;  
8          i++;  
9      }  
10     return 0;  
11 }
```

Beş kez «Merhaba!» yazdırılması gerektiğini varsayalım. Döngü kullanılmadan bunu yapmak için beş ayrı cout komutu yazmak gerekir. Ancak döngü kullanılarak bu işlem çok daha düzenli ve verimli bir şekilde gerçekleştirilebilir.

Bu örnekte, i değişkeni döngünün her çalışmasında artırılır ve i < 5 koşulu yanlış olduğunda döngü sona erer.

Programlamada döngü kullanmanın en önemli avantajlarından biri, tekrar eden işlemlerin otomatikleştirilmesidir. Programcının kodu elle kopyalayıp tekrar yazması yerine, döngüler aynı kod bloğunun küçük değişikliklerle birçok kez kullanılmasını sağlar. Bu durum yalnızca programı daha kısa ve bakımı daha kolay hâle getirmekle kalmaz, aynı zamanda hata olasılığını da azaltır. Bununla birlikte, döngünün her tekrarı genellikle sayaç adı verilen bir değişken aracılığıyla kontrol edilir. Sayaç, her tekrarda ek mantıksal işlemler gerçekleştirilmesine olanak sağlar. Sayaç, elemanları indeksleme, değerleri toplama, koşulları kontrol etme ve birçok farklı işlem için kullanılabilir. Döngülerin iyi planlanmış şekilde kullanılması, kodun yapısını ve okunabilirliğini önemli ölçüde geliştirir.

İlk 10 doğal sayının (1'den 10'a kadar) toplamını hesaplamak istiyoruz.

```
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      int toplam = 0;
6      for (int i = 1; i <= 10; i++) {
7          toplam += i;
8      }
9      cout << "İlk 10 doğal sayının toplamı: " << toplam << endl;
10     return 0;
11 }
```

Burada for döngüsü, i sayacını otomatik olarak başlatır, koşulu kontrol eder ve sayacı günceller. Her yinelemede i, toplam değişkenine eklenir. İşlem tamamlandığında toplam toplam değeri içerir.

Döngüler yalnızca tekrar işlemlerini sağlamakla kalmaz, aynı zamanda programların dinamik davranış göstermesine de olanak tanır. Kullanıcıdan alınan girişe, bir hesaplama sonucuna veya başka bir dış etkene bağlı olarak döngü çalışmaya devam edebilir veya sonlandırılabilir. Bu tür döngülere koşullu döngüler adı verilir ve genellikle while veya do-while yapıları kullanılır. while döngüsünde koşul her çalıştırmadan önce kontrol edilirken, do-while döngüsünde ise kontrol yapılmadan önce döngü gövdesi en az bir kez çalıştırılır. Bu tür döngüler, kaç yineleme yapılacağının önceden bilinmediği durumlarda sıkça kullanılır. Örneğin, kullanıcı geçerli bir değer girene kadar veya belirli dinamik koşullar gerçekleşene kadar döngünün devam etmesi bu duruma örnektir. Bu özellik, programlama diline aynı kodu tekrar yazmadan farklı senaryolara tepki verebilme esnekliği sağlar.

Programcılar, simülasyonlardan veri analizlerine kadar karmaşık algoritmalar oluşturmak için döngüleri sıklıkla koşullar ve veri işlemleriyle birlikte kullanırlar. Uygun döngü türünü seçme, doğru koşulu belirleme ve sonsuz döngülerden kaçınma becerisi, iyi programlamanın temel unsurlarından biridir. Bazen koşulda veya sayacı güncellemesinde yapılan küçük bir hata, programın hiçbir zaman sona ermemesine neden olabilir. Bu nedenle döngüler, dikkatli ve düzenli kullanım gerektiren güçlü bir araçtır.



1. Programlamada döngü nedir?
2. Hangi döngü türlerini biliyorsun ve aralarındaki farklar nelerdir?
3. Döngülerde sayaçlar neden kullanılır?
4. while döngüsündeki koşul hiçbir zaman sağlanmazsa ne olur?
5. 1'den 50'ye kadar olan tüm çift sayıları yazdıran bir program yaz!
6. Klavyeden girilen N değerine göre 1'den N'ye kadar olan tüm sayıların toplamını hesaplayan bir program yaz!
7. Kullanıcıdan pozitif bir sayı girmesini isteyen ve kullanıcı bunu yapana kadar devam eden bir program yaz!
8. 1'den 10'a kadar olan sayıların çarpımını hesaplayan bir program yaz!

Öğrendiklerini Hatırla!

- Döngüler, kodların tekrar çalıştırılmasını sağlar.
- Üç temel döngü türü kullanılır: for, while ve do-while.
- Döngüler, işlemlerin otomatikleştirilmesi için etkili bir yöntemdir.
- Sayaçlar ve koşullar, tekrar sayısını kontrol eder.
- Yanlış oluşturulan bir döngü, sonsuz çalışmaya neden olabilir.

**DAHA FAZLA BİLGİ EDİNMEK İSTEYENLER İÇİN**

Yarışmalı programlama ve "Kunduz" türü problemlerde döngüler, büyük veri dizilerinin analiz edilmesi veya işlenmesini gerektiren problemlerin çözümünde kullanılır. Örneğin, belirli bir metinde bir harfin kaç kez tekrarlandığını saymak veya bir labirentte hareket simülasyonu gerçekleştirmek gibi işlemler döngüler aracılığıyla yapılabilir.

Döngüler ve koşulların birlikte kullanılmasıyla sıralama, arama ve matris analizi gibi birçok algoritmik problem çözülebilir. Öğrenci döngülerin mantığını ne kadar erken öğrenirse, özyineleme (rekürsiyon), veri yapıları ve optimizasyon gibi daha ileri programlama kavramlarına o kadar hızlı geçiş yapabilir.



3.28

While Döngüsü Yapısı için Alıştırma Soruları



KOLAY ALIŞTIRMALAR

- 1'den 15'e kadar olan sayıları yazdıran bir program yaz!
- 10'dan 1'e kadar olan sayıları tersten yazdıran bir program yaz!
- 20'den 40'a kadar olan tüm çift sayıları yazdıran bir program yaz!
- 5'ten başlayarak 30'a kadar her üçüncü sayıyı yazdıran bir program yaz!
- 1'den 100'e kadar 10'a bölünebilen tüm sayıları yazdıran bir program yaz!
- «While döngüleri çalışıyorum» ifadesini tam 6 kez yazdıran bir program yaz!
- 30'dan 40'a kadar olan sayıların toplamını gösteren bir program yaz!



ORTA DÜZEY ALIŞTIRMALAR

8. Klavyeden girilen N sayısına kadar olan ve 3'e bölünebilen sayıları hesaplayan bir program yaz!
9. Kullanıcıdan 10 tane tam sayı girmesini isteyen ve girilen sayılar arasındaki en büyüğünü gösteren bir program yaz!
10. 1'den 50'ye kadar olan sayılardan 7'ye bölünebilen kaç sayı olduğunu bulan bir program yaz!
11. 1'den 100'e kadar olan tüm tek sayıların toplamını hesaplayan bir program yaz!
12. Girilen sayının kaç basamaklı olduğunu bulan bir program yaz!
13. İki girilen sayı A ve B arasındaki tüm sayıların toplamını yazdıran bir program yaz!
14. Girilen sayının hem 2'ye hem de 5'e bölünüp bölünmediğini kontrol eden bir program yaz!



ZOR ALIŞTIRMALAR

15. Matematik kulübünde öğrenciler «mükemmel sayıları» öğrenmektedir. Mükemmel sayılar, pozitif bölenlerinin (kendisi hariç) toplamı kendisine eşit olan sayılardır. Örneğin, 6 için: $1 + 2 + 3 = 6$. Girilen bir tam sayıyı okuyup bu sayının mükemmel sayı olup olmadığını kontrol eden bir program yaz!
16. Girilen sayının en küçük rakamını bulan bir program yaz!
17. 1'den N'ye kadar olan sayıların karelerinin toplamını hesaplayan bir program yaz!
18. 1'den 100'e kadar olan sayılardan hem 3'e hem de 4'e bölünebilen kaç sayı olduğunu bulan bir program yaz!
19. 10'dan 999'a kadar olan tüm palindrom sayıları sayan bir program yaz!
20. Güvenlik kodu kontrolünde yalnızca çift rakamlardan (0, 2, 4, 6, 8) oluşan sayılara izin verilmektedir. Sayıda en az bir tane tek rakam bulunursa kod geçersiz kabul edilir. Bu durumu kontrol eden bir program yaz!

Bu konuyu tamamladığında...

- ✓ Programlamadaki temel döngü türlerini ayırt edersin,
- ✓ Komutların tekrar edilmesi gereken durumları fark edersin,
- ✓ Koşulların program akışının kontrolündeki anlamını anlarsın,
- ✓ C++ dilinde yazılmış basit bir döngüyü okuyup anlayabilirsin.

**TEKRARLAMA SORULARI:**

1. Her döngü while kullanılarak yazılabilir mi?
2. 100'den 1'e kadar olan tüm sayıları ters sırada yazdıran bir program yaz!

**AŞAĞIDAKİ KONULARDA ŞUNLARI ÖĞRENECEKSİN:**

- do while döngüsünün nasıl çalıştığını açıklarsın,
- do while döngüsünün uygun olduğu durumları tanırırsın,
- Bir koşul sağlanana kadar tekrar eden do while döngüsünü kullanan program yazarsın,
- Koşulu doğru şekilde oluşturarak sonsuz döngülerden kaçınırsın,
- Döngü içerisinde değişkenleri, koşulları ve değer güncellemelerini kullanırsın.

Programlamada sıkça, bir kod bloğunun koşulun doğru olup olmadığına bakılmaksızın en az bir kez çalıştırılması gereken durumlarla karşılaşılır. Bu gibi durumlarda, do-while döngüsü kullanılır. Bu döngü, kendine özgü bir yapıya sahiptir: Önce döngünün gövdesi çalıştırılır, ardından koşul kontrol edilir. Koşul doğru olmaya devam ederse döngü tekrar çalıştırılır. Bu yaklaşım, programın ilk olarak bir işlemi gerçekleştirmesinin gerektiği

durumlarda kullanışlıdır. Örneğin, kullanıcıdan bir sayı alınması, menüden bir seçenek seçilmesi veya en az bir kez gerçekleşmesi gereken bir işlemin başlatılması gibi durumlar buna örnektir. do-while döngüsünün yapısı şu şekildedir:

```
do {  
    // çalıştırılacak komutlar  
} while (koşul)
```

ÖRNEK 1

AŞAĞIDAKİ örnek, kullanıcı tarafından pozitif bir sayı girilmesini sağlamak için do-while döngüsünün nasıl kullanıldığını göstermektedir. Döngü, geçerli bir değer girilene kadar tekrar çalışmaya devam eder:

```
1  #include <iostream>  
2  using namespace std;  
3  
4  int main() {  
5      int sayi;  
6      do {  
7          cout << "Pozitif bir sayı girin: ";  
8          cin >> sayi;  
9      } while (sayi <= 0);  
10  
11     cout << "Geçerli sayı: " << sayi << endl;  
12     return 0;  
13 }
```

Bu durumda, kullanıcı sayıyı en az bir kez girme fırsatına sahiptir. Eğer girilen değer geçerli değilse (pozitif bir sayı değilse), program doğru değer girilene kadar sayıyı tekrar girmesini ister. Bu yaklaşım, geçerlilik kontrolünün gerekli olduğu durumlarda kullanıcıyla güvenli etkileşim sağlamak için de kullanılır.

do-while döngüsü genellikle kullanıcı etkileşiminin yönetildiği durumlarda kullanılır. Örneğin menülerde, kullanıcının bir seçenek belirlemesi ve ardından "çıkış" seçeneğini seçene kadar tekrar seçim yapabilmesi buna örnektir. while döngüsünden farklı olarak, koşul başlangıçta sağlanmıyorsa while döngüsü hiç çalışmayabilir. Buna karşılık do-while döngüsü, döngü gövdesindeki işlemlerin en az bir kez çalıştırılacağını garanti eder. Bu avantaj, ilk kullanıcı girişinin veya herhangi bir fonksiyon çağrısının koşulun mantıksal durumundan bağımsız olarak mutlaka gerçekleştirilmesi gereken durumlarda önemlidir. Böylece programın daha öngörülebilir ve kontrollü bir şekilde çalışması sağlanır.

AŞAĞIDAKİ örnekte üç seçenekli bir menü gösterilmektedir. Kullanıcı 3 – “çıkış” seçeneğini girene kadar seçim yapmaya devam edebilir:

```
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      int secim;
6      do {
7          cout << "==== Ana Menü ==== " << endl;
8          cout << "1. Veri gir" << endl;
9          cout << "2. Verileri göster" << endl;
10         cout << "3. Çıkış" << endl;
11         cout << "Seçiminiz: ";
12         cin >> secim;
13
14         if (secim == 1)
15             cout << "Veriler alınıyor..." << endl;
16         else if (secim == 2)
17             cout << "Veriler gösteriliyor..." << endl;
18         else if (secim != 3)
19             cout << "Bilinmeyen seçenek!" << endl;
20     } while (secim != 3);
21
22     cout << "Program sona erdi." << endl;
23     return 0;
24 }
```

Bu örnek, etkileşimli programlarda do-while döngüsünün tipik kullanımını göstermektedir. Önce menü görüntülenir, ardından kullanıcıdan giriş beklenir ve işlem, çıkış seçeneği seçilene kadar tekrarlanır. Bu yöntem, açık ve güvenli bir mantıkla etkileşimli konsol uygulamalarının gerçekleştirilmesinde klasik bir yaklaşımdır.

do-while döngüsünü diğerlerinden ayıran temel özellik, çalıştırma sırasıdır: Döngünün gövdesi koşul kontrol edilmeden önce çalıştırılır. Diğer döngülerde ise koşul her yinelemeden önce kontrol edilir. Bu nedenle do-while, koşul başlangıçta doğru olmasa bile her zaman en az bir kez çalışır. while ve for döngülerinde ise koşul başlangıçtan itibaren sağlanmıyorsa döngü hiç çalışmayabilir.

Ayrıca, koşulun hiçbir zaman yanlış duruma gelmemesi durumunda sonsuz döngü oluşabileceğine dikkat edilmelidir. Bu nedenle do-while döngüsünün gövdesinde, önceki örneklerde gösterildiği gibi giriş, seçim veya rastgele değer üretimi yoluyla döngünün sona ermesini sağlayacak açık bir mantık bulunmalıdır. Kodun okunabilirliği ve bakım kolaylığı açısından do-while, yalnızca ilk çalıştırmanın gerçekten gerekli olduğu durumlarda kullanılmalıdır.



1. do while döngüsünü while döngüsünden ayıran temel yapısal özellik nedir?
2. while veya for yerine do while kullanmak hangi durumlarda daha uygundur?
3. Bir do while döngüsünün hiç çalışmaması mümkün müdür? Açıklayınız!
4. do while neden etkileşimli menülerde kullanılır?
5. Kullanıcıdan 100'den büyük bir sayı girmesini isteyen bir program yazın. Kullanıcı daha küçük bir sayı girerse, do while kullanarak tekrar sayı istensin!
6. İki zar atma işlemini, zarların aynı sayıyı göstermesine (çift gelmesine) kadar simüle eden bir program yazın!
7. Kullanıcıya programa devam etmek isteyip istemediğini ("evet" veya "hayır") soran ve kullanıcı "hayır" girene kadar devam eden bir program yazın!
8. Kullanıcının girdiği sayıları sıfır girene kadar toplayan ve toplamı gösteren bir program yazın!

Öğrendiklerini Hatırla!

- do while döngüsü, bir komut bloğunun en az bir kez çalıştırılması istendiğinde kullanılır.
- Döngünün devam etme koşulu, döngü çalıştırdıktan sonra kontrol edilir.
- Veri girişi, etkileşimli menüler, rastgele simülasyonlar ve başlangıç kontrollerinde kullanılır.
- Sonsuz döngü oluşmaması için her zaman döngünün sona ermesini sağlayacak açık bir mantık bulunmalıdır.



DAHA FAZLA BİLGİ EDİNMEK İSTEYENLER İÇİN

Bazı programlama dillerinde, örneğin Python'da, do while döngüsü yerleşik bir yapı olarak bulunmaz. Bu gibi durumlarda, do while mantığı while True: kullanılarak ve döngüden çıkmak için içerisine yerleştirilen break koşulu ile gerçekleştirilir. Bunun yanında, tehlike oyunları simülasyonları veya video oyunlarındaki diyalog yöneticileri gibi gelişmiş simülasyonlarda da do while mantığı kullanılmaktadır. Bu tür sistemlerde programcılar, kullanıcıyla tekrar eden etkileşimleri sağlamak amacıyla do while yapısını sıklıkla grafik arayüzlerle birlikte kullanırlar.





KOLAY ALIŞTIRMALAR

1. 1'den 30'a kadar olan tüm tek sayıları yazdıran bir program yaz!
2. 10'dan 1'e kadar olan sayıları azalan sırada yazdıran bir program yaz!
3. 1'den 100'e kadar olan tüm çift sayıların toplamını hesaplayan bir program yaz!
4. Kullanıcıdan 0 girene kadar sayı girmesini isteyen bir program yaz!
5. 100'den büyük bir sayı girilene kadar "Tekrar gir!" mesajını gösteren bir program yaz!
6. 1 ile 50 arasındaki 5'e bölünebilen tüm sayıları yazdıran bir program yaz!
7. 1 ile 50 arasındaki 3'e bölünebilen tüm sayıların toplamını hesaplayan bir program yaz!
8. Girilen pozitif bir sayının rakamları toplamını hesaplayan bir program yaz!



ORTA DÜZEY ALIŞTIRMALAR

9. Girilen sayının asal sayı olup olmadığını belirleyen bir program yaz!
10. Girilen sayının rakamlarını ters sırada gösteren bir program yaz!
11. 1 ile 15 arasındaki tüm tek sayıların çarpımını hesaplayan bir program yaz!
12. Kullanıcının -1 girene kadar girdiği sayı dizisindeki en büyük sayıyı bulan bir program yaz!
13. Girilen sayının kaç basamaklı olduğunu bulan bir program yaz!
14. Kullanıcıdan sayılar alıp negatif sayı girilene kadar bu sayıların ortalamasını hesaplayan bir program yaz!



ORTA ZORLUKTA ALIŞTIRMALAR

15. Bir sayı bulmacasında "gizli sayı" 100 ile 1000 arasında saklıdır. Bu sayı için iki koşul bilinmektedir: Sayı 17'ye bölünebilmeli ve rakamları toplamı 15 olmalıdır. 100 ile 1000 arasındaki bu koşulları sağlayan ilk sayıyı (küçükten büyüğe doğru) bulan ve yazdıran bir program yaz!
16. Girilen sayının faktöriyelini hesaplayan bir program yaz!
17. Kullanıcının girdiği N sayısına kadar olan ilk doğal sayıların kareleri toplamını hesaplayan bir program yaz!
18. Öğrenciler analiz etmek için bilgisayara bir tam sayı dizisi giriyor. Sayı girişi 0 girilene kadar devam eder. 0 yalnızca bitiş işareti olarak kullanılır ve diziye dahil edilmez. Girilen farklı sayılar arasındaki ikinci en büyük sayıyı bulan bir program yaz!

Bu konuyu tamamladığında...

- ✓ Koşul ve döngü gibi temel kontrol yapılarını ayırt edersin,
- ✓ Programın belirli bir işlemi tekrar etmesi gereken durumları fark edersin,
- ✓ En az bir kez çalıştırılması gereken durumlarda do while döngüsünü kullanırsın,
- ✓ Programdaki işlemlerin sıralı yürütülme mantığını takip edersin.

**TEKRARLAMA SORULARI:**

1. Sonsuz bir do while döngüsünden nasıl kaçınılabilir?
2. Kullanıcıdan doğru belirlenmiş bir şifreyi ("tajna123") girene kadar şifre isteyen bir program yaz

**AŞAĞIDAKİ KONULARDA ŞUNLARI ÖĞRENECEKSİN:**

- Başlangıç ve bitiş değerleriyle sayma işlemleri için for döngüsünü kullanırsın,
- for döngüsünde başlatma (initiation), koşul ve güncelleme işlemlerinin nasıl çalıştığını açıklarsın,
- Sabit tekrar sayısına sahip programlar yazarsın,
- for, while ve do while döngülerini karşılaştırırsın,
- Tekrarlanan işlemler içeren gerçek yaşam problemlerini çözersin.

Programlamada belirli komutların tekrar edilmesi gereksinimi günlük olarak sıkça karşılaşılan bir durumdur. Bu tekrarlar genellikle bir işlemin kaç kez gerçekleştirileceğinin önceden bilindiği durumlarda ortaya çıkar. Böyle durumlarda, tekrar sayılarını kontrol etmeyi sağlayan bir kontrol yapısı en uygun çözümdür. Bu yapı C++'ta (ve birçok diğer programlama dilinde) for döngüsü olarak adlandırılır. Bu yapı kısa, okunabilir ve başlangıç değeri, sonlandırma koşulu ve adım (güncelleme) mantığını açık bir şekilde gösteren bir yapıya sahiptir. Bu özelliklerinden dolayı for döngüsü, dizilerin, tabloların işlenmesinde, veri girişlerinde ve belirli sayıda adımda gerçekleştirilen hesaplamalarda sıklıkla kullanılır.

Bu derste for döngüsünün yapısı, çalışma biçimi ve gerçek yaşamdan ve programlama durumlarından örnekler ele alınacaktır.

C++'ta for döngüsü, üç temel bölümü içerecek şekilde tasarlanmıştır: başlatma (initiation), devam etme koşulu ve değer güncelleme. Bu bölümlerin tamamı döngünün parantezleri içerisinde yer alır. Yapısı şu şekildedir:

```
for (başlatma, koşul, güncelleme) {  
    // tekrar edilen komutlar  
}
```

"Başlatma" bölümünde değişkenler için başlangıç değerleri verilir. Birden fazla değişkenin başlatılması gerekiyorsa, bu değişkenler virgül ile ayrılır. Aynı şekilde güncelleme bölümünde de genellikle yineleme değişkenleri (iteratörler) kullanılır, birden fazla güncelleme yapılması gerekiyorsa, bunlar da virgül ile ayrılır. Bu yapı, döngünün tüm temel bölümlerinin tek bir yerde toplanmasını sağlar ve kodun okunabilirliğini artırır. Başlatma bölümü genellikle sayacın başlangıç değerini belirler (örneğin `int i = 1;`). Koşul bölümü döngünün ne zamana kadar devam edeceğini belirler (örneğin `i <= 5;`). Güncelleme bölümü ise her yinelemede değerlerin değişmesini sağlar (örneğin `i++`).

ÖRNEK 1



1'den 5'e Kadar Sayıların Yazdırılması

Bu örnekte döngü `i = 1` değeriyle başlar ve `i` değeri 5'e küçük veya eşit olduğu sürece devam eder. Her yinelemeden sonra `i` değeri 1 artırılır. Bu şekilde ekranda 1'den 5'e kadar olan sayılar gösterilir.

Birçok durumda programlar, yüksek bir değerden daha düşük bir değere doğru ilerleyen bir sayaca ihtiyaç duyar. Bu durum geri sayım işlemlerinde, elemanların ters sırada gösterilmesinde veya bir dizinin sonuna ulaşılana kadar yapılan aramalarda gerekli olabilir. Ters sayaç kullanan for döngüsü, daha büyük bir başlangıç değeriyle tanımlanır. Döngünün sona ermesi için sayaç belirli bir değerin altına düştüğünde koşul sonlanır ve sayaç değeri azaltılarak güncellenir. Bu tür bir kullanım, for döngüsünün esnekliğini gösterir ve döngünün problemin mantığına uygun şekilde düzenlenebileceğini ortaya koyar.

```
1  #include <iostream>  
2  using namespace std;  
3  
4  int main() {  
5      for (int i = 1; i <= 5; i++)  
6      {  
7          cout << i << endl;  
8      }  
9      return 0;  
10 }  
11
```

10'dan 1'e Kadar Geri Sayım

Bu örnekte döngü $i = 10$ değeriyle başlar ve i değeri 1'e eşit veya büyük olduğu sürece devam eder. Her yinelemede $i--$ kullanılarak sayaç değeri azaltılır. Ekranda 10, 9, 8, ... 1 sayıları gösterilir. Bu tür geri sayım işlemleri, örneğin zamanlayıcılarda, alarmlarda ve kalan sürenin gösterilmesi gereken etkileşimli oyunlarda kullanılır.

Bunun yanında for döngüsü, toplama, çarpma veya toplam değerlerin hesaplanması gibi işlemler için de kullanılabilir. Bu kullanım, özellikle kullanıcıdan alınan verilerle çalışırken veya belirli sayıda yineleme gerektiren hesaplamalarda oldukça faydalıdır. Döngü yalnızca değerleri yazdırmak için değil, sayaç değerlerini işleyerek sonuçları bir değişkende saklamak için de kullanılabilir. Bu yaklaşım, toplam, ortalama, faktöriyel hesaplama ve hatta en büyük veya en küçük değeri bulma gibi işlemleri içeren problemlerde sıkça kullanılır.

```

1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      for (int i = 10; i >= 1; i--)
6      {
7          cout << i << " ";
8      }
9      return 0;
10 }
11

```

İlk 100 Doğal Sayının Toplamının Hesaplanması

Bu örnekte toplam değeri başlangıçta 0 olarak belirlenir ve her yinelemede i değişkeninin mevcut değeri toplama eklenir. Döngü tamamlandığında sum değişkeni son değeri içerir: 5050. Bu tür problemler hem başlangıç düzeyindeki programlama çalışmalarında hem de matematiksel modellerde karşımıza çıkar.

Gerçek programlama çözümlerinde for döngüsü, sıklıkla bir dizi veya veri serisindeki elemanlara erişmek için kullanılır. Bu yaklaşım, dizinin her elemanı üzerinde belirli ve kontrollü bir şekilde, önceden tanımlanmış sayıda tekrar ile işlem yapılmasını sağlar. Burada for döngüsünün yapısı oldukça uygundur, çünkü dizinin boyutunu kullanarak döngünün sona erme koşulunu belirler ve her yinelemede indis değerini otomatik olarak günceller. Döngünün bu şekilde kullanımı, veri işleme, istatistik, hesaplamaların otomatikleştirilmesi ve birçok algoritmik problemin çözümünde temel bir yöntemdir.

```

1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      int toplam = 0;
6      for (int i = 1; i <= 100; i++)
7      {
8          toplam += i;
9      }
10     cout << "Toplam: " << toplam << endl;
11     return 0;
12 }

```

BİLGİ KONTROLÜ SORULARI VE ETKİNLİKLER



1. for döngüsü nedir ve ne zaman kullanılır?
2. Sözdizimi ve kullanım açısından for ve while döngüsü arasındaki fark nedir?
3. for döngüsünün yapısı nasıldır ve üç bölümü neyi ifade eder?
4. for döngüsünde güncelleme bölümü yazılmazsa ne olur?
5. AŞAĞIDAKİ programın ne yazdıracağını açıklayın:

```
for (int i = 1, i <= 5, i++) { cout << i * i << " ",  
}
```
6. for döngüsünü kullanarak 2'den 20'ye kadar olan tüm çift sayıları yazdıran bir program yazın!
7. İlk 10 doğal sayının çarpımını hesaplayan bir program yazın!
8. 1 ile 99 arasındaki tüm tek sayıların toplamını yazdıran bir program yazın!
9. for döngüsünü kullanarak A'dan Z'ye kadar alfabeyi gösteren bir program yazın!
10. 50'den 1'e kadar geri sayım yapan ve sonunda "BİTTİ!" yazdıran bir program yazın!

Öğrendiklerini Hatırla!

- for döngüsü, tekrar sayısının bilindiği durumlarda kullanılan bir tekrar yapısıdır.
- Başlatma, koşul ve güncelleme bölümlerinden oluşur.
- Sayma, hesaplama ve dizi işlemleri gibi sayaç gerektiren görevler için idealdir.
- Artan veya azalan sırada kullanılabilir.
- Programlama ve veri analizindeki birçok gerçek yaşam probleminde uygulanır.



DAHA FAZLA BİLGİ EDİNMEK İSTEYENLER İÇİN

- C++'ta iç içe for döngüleri de bulunmaktadır. Bu döngülerde bir döngü başka bir döngünün içine yerleştirilir. Matrislerin ve tabloların işlenmesinde kullanılırlar.
- for döngüsü, belirli hesaplamaların gerçekleştirilmesi için if gibi koşullu ifadelerle birlikte kullanılabilir.
- İleri düzey programlamada, vector, map ve set gibi veri yapılarında yineleyiciler (iterator) kullanılarak dolaşma işlemleri gerçekleştirilir.
- C++11 ve daha yeni sürümlerde aralık tabanlı for döngüleri (range-based for) de kullanılabilir.





KOLAY ALIŞTIRMALAR

1. 1'den 20'ye kadar olan tüm sayıları yazdıran bir program yaz!
2. 1'den 50'ye kadar olan tüm tek sayıları yazdıran bir program yaz!
3. 1'den 100'e kadar olan sayıların toplamını hesaplayan bir program yaz!
4. 10'dan 1'e kadar olan sayıları ters sırada yazdıran bir program yaz!
5. 1'den 5'e kadar olan sayıların çarpımını hesaplayan bir program yaz!
6. 1'den 30'a kadar 3'e bölünebilen tüm sayıları yazdıran bir program yaz!
7. 1'den 10'a kadar olan tüm sayıların karesini yazdıran bir program yaz!



ORTA ZORLUKTA ALIŞTIRMALAR

8. 1'den 100'e kadar olan tüm çift sayıların toplamını hesaplayan bir program yaz!
9. Kullanıcı tarafından girilen N sayısına kadar olan ve 5'e bölünebilen tüm sayıları yazdıran bir program yaz!
10. 1'den N'ye kadar olan ve 2'ye bölünemeyen (tek) sayıların toplamını hesaplayan bir program yaz!
11. 1'den 10'a kadar olan her sayı için karesini ve küpünü yazdıran bir program yaz!
12. 1'den 100'e kadar olan sayılardan 7'ye bölünebilen kaç sayı olduğunu bulan bir program yaz!
13. Kullanıcı tarafından girilen N sayısının faktöriyelini hesaplayan bir program yaz!
14. Başlangıç değeri 3, farkı 5 olan ve 10 elemandan oluşan bir aritmetik dizi yazdıran bir program yaz!



ZOR ALIŞTIRMALAR

15. 1'den 100'e kadar olan sayılarda 3 rakamının kaç kez geçtiğini hesaplayan bir program yaz!
16. Klavyeden girilen N sayı arasındaki en küçük ve en büyük sayıyı yazdıran bir program yaz!
17. Girilen N sayının aritmetik ortalamasını hesaplayan bir program yaz!
18. 100 ile 999 arasındaki rakamları toplamı 15'e eşit olan tüm sayıları yazdıran bir program yaz!
19. 1'den 100'e kadar onlar basamağındaki rakamı 2 olan tüm sayıları yazdıran bir program yaz!
20. 1'den 100'e kadar olan sayıları yazdıran bir program yaz. Ancak 3'e bölünebilen sayıların yerine "Fizz", 5'e bölünebilen sayıların yerine "Buzz", hem 3'e hem de 5'e bölünebilen sayıların yerine "FizzBuzz" yazdır!



SONUÇ

“C++ ile Programlama” konusunda, öğrencilerin bilişim ve algoritmik düşünme alanındaki ileri düzey öğrenmeleri için temel oluşturan kapsamlı bir bölüm tamamlanmaktadır. Başlangıçta mantıksal ve algoritmik problemler ele alınarak, problemlerin adım adım analiz edilmesi ve açık yönergelere dönüştürülmesi alışkanlığı kazandırılır. Daha sonra temel bilişim kavramları ve ikili sayılar tanıtılır. Böylece bilgisayarda verilerin nasıl temsil edildiği ve işlendiği anlaşılır. Kodlama ve şifreleme konularına giriş yapılarak bilgilerin güvenli kullanımının önemi vurgulanır.

Devamında programlama kavramı, programlama dilleri ve derleyiciler üzerinde durulur. Farklı programlama dilleri arasındaki farklar incelenirken C++ ana öğrenme aracı olarak ele alınır, aynı zamanda Scratch, Java, Python, PHP ve Lisp gibi diğer dillerin programlama dünyasındaki yeri de tanıtılır. Entegre geliştirme ortamı (IDE) ile teorik kavramların uygulamalı çalışmaya nasıl dönüştürüldüğü gösterilir: Programların yazılması, çalıştırılması ve hata ayıklanması işlemleri gerçekleştirilir. Söзде kod (pseudocode) ve ardışık komut yapıları aracılığıyla çözümün önce anlaşılır bir “insan dili” ile ifade edilmesi, ardından belirli C++ komutlarına dönüştürülmesi öğretilir.

Sonraki bölümde değişkenler ve sabitler, bunların türleri, değer atama ve değerleri görüntüleme kuralları ele alınarak verilerin bellekte nasıl saklandığı konusunda temel bir anlayış oluşturulur. Aritmetik işlemler ve ifadeler, veri girişi, karşılaştırma ve mantıksal işlemler ile programların matematiksel ve mantıksal temeli oluşturulur. Bu temel üzerine program akışının kontrolü inşa edilir: iki ve daha fazla seçeneekli seçim yapıları, iç içe koşullar ve üç temel döngü türü (while, do while ve for) öğretilir. Böylece öğrencilerin tekrar ve dallanma gerektiren daha karmaşık problemleri çözebilmesi sağlanır.

Her bölümün sonunda yer alan çok sayıdaki uygulama göreviyle öğrenilen bilgiler pekiştirilir, bağımsız düşünme desteklenir ve C++ dilini kullanma konusunda güven geliştirilir. Konunun sonunda öğrenci, bir programın mantığını anlayabilecek, algoritma tasarlayabilecek, bunu söзде kod ile ifade edebilecek ve doğru çalışan, veri okuyan, işleyen ve görüntüleyen bir C++ programına dönüştürebilecek seviyeye ulaşır. Bu şekilde, sonraki konularda ve eğitim yıllarında daha ileri programlama kavramlarının ve tekniklerinin öğrenilebilmesi için sağlam bir temel oluşturulur.



KONU 3: GENEL TEKRARLAMA

1. Kendi kelimelerinle algoritmanın ne olduğunu açıkla ve günlük hayattaki sıradan açıklayıcı bir işlem den (örneğin yemek tarifi) nasıl farklı olduğunu belirt!
2. Günlük hayattan bir durum seç (okula gitme, sınava hazırlanma, alışveriş yapma) ve bunu açık bir adımlar dizisi şeklinde yaz! Giriş, işlem ve çıkış olan adımların altını çiz!
3. Üç koşullu bir mantık problemi oluştur (örneğin insan, nesne veya renklerin düzenlenmesiyle ilgili) ve en az iki olası çözüm yaz!
4. Ondalık sayı olan 37'yi ikili sayı sistemine dönüştür! İşlemi adım adım açıkla!
5. Verilen ikili sayı 101101₂'nin ondalık değerini hesapla ve sonucun nasıl elde edildiğini açıkla!
6. Günlük hayatta bilgisayarlar dışında bilgi kodlamasının kullanıldığı iki örnek ver ve her örnekte neyin kodlandığını açıkla!
7. Şifreleme (encryption) nedir? Açıkla ve bir mesajın şifrelenmesinin önemli olacağı bir örnek ver!
8. Bir programın deterministik olması ne anlama gelir? Kendi kelimelerinle açıkla ve bunun problem çözmede neden önemli olduğunu belirt!
9. Programlama dili ile doğal dil (Makedonca veya İngilizce gibi) arasındaki üç benzerliği ve üç farkı yaz!
10. Derleyici (compiler) ve yorumlayıcı (interpreter) arasındaki farkı açıkla! Hangi türün kullanıldığını bilmenin önemli olduğu bir durum belirt!
11. Entegre geliştirme ortamının (IDE) program yazmadaki rolünü açıkla! Programcının işini kolaylaştıran en az üç özelliğini belirt!
12. Üç tam sayı okuyan ve bunların aritmetik ortalamasını gösteren bir sözde kod algoritması yaz!
13. Girilen uzunluk ve genişlik değerlerine göre bir dikdörtgenin alanını ve çevresini hesaplayan ve gösteren bir sözde kod algoritması yaz!
14. AŞAĞIDAKİ değişkenlerin her biri için uygun veri türünü belirle: öğrenci yaşı, hava sıcaklığı, bilet fiyatı ve şehir adı. Nedenini açıkla!
15. Verilen x değeri için $3 \cdot x^2 - 2 \cdot x + 5$ ifadesinin değerini hesaplayan sözde kod ifadesini yaz! Bu ifadenin adım adım nasıl hesaplanacağını açıkla!
16. Bir karşılaştırma ifadesi ve bir mantıksal ifade örneği oluştur! Değerlerinin hangi durumlarda "doğru", hangi durumlarda "yanlış" olacağını açıkla!
17. Bir öğrencinin 50 puan geçme sınırına göre dersi geçip geçmediğini kontrol eden bir koşul (if-else yapısı) sözde kodu yaz! Sonuca göre uygun mesaj göster!
18. while ve do while döngüleri arasındaki farkı açıkla! Birincisinin ve ikincisinin daha uygun olduğu birer durum örneği ver!
19. 1'den 50'ye kadar olan tüm çift sayıları ekranda tek satırda gösteren bir for döngüsü algoritması yaz!
20. Hem koşul hem de döngü gerektiren günlük hayattan bir problem oluştur (örneğin birden fazla ürün alımında indirimli toplam fiyat hesaplama)! Çözüm mantığını, programa dönüştürülebilecek adımlar dizisi olarak açıkla!



KONU 4:

METİN ÇALIŞMASI

Günümüzde bilgisayarlar, metin belgeleri hazırlamak ve bunları düzenlemek için her gün kullanılmaktadır. Metin düzenleme amacıyla en yaygın kullanılan kelime işlemci programları MS Word, LibreOffice paketindeki Writer, ayrıca Word'ün çevrim içi sürümü ve Google Docs'tur. Bu programlar kullanımı kolaydır ve farklı biçim ve stillerdeki metinleri düzenlemek için birçok olanak sunar. Bu ders kitabında, ilk iki kelime işlemci programının işlevleri açıklanacaktır. Bu işlevler, büyük ölçüde çevrim içi metin düzenleyicilerinin sunduğu özelliklere benzemektedir.

Yeni Kavramlar

- Başlık, alt başlık, içindekiler, dizin, şablon sihirbazı, form, metin kutusu, seçim düğmesi, açılır menü.

Bu konuyu tamamladığında...

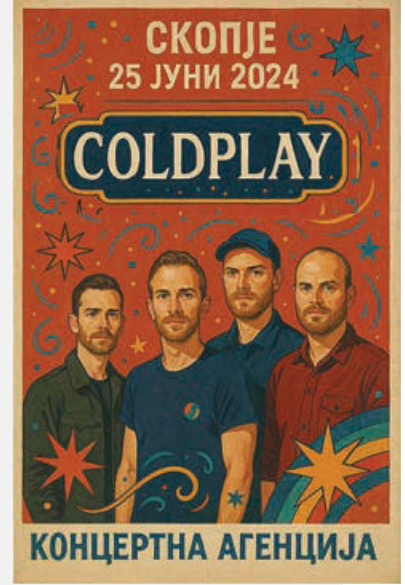
- ✓ Yazı tipi, yazı rengi ve yazı stilini değiştirirsin.
- ✓ Paragraf satır aralığını ve birden fazla paragraf arasındaki boşluğu ayarlarsın.
- ✓ Metni sütunlar hâlinde yazarsın.
- ✓ Resimler, fotoğraflar, tablolar, grafikler ve diğer grafik öğelerini eklersin.



TEKRARLAMA SORULARI:

1. En sevdiğin grubun konseri için AŞAĞIDAKİ yönergelerle göre bir afiş hazırla:

- Konserin yeri ve tarihi/saati, Kiril alfabesiyle ve 20 punto büyüklüğünde yazılmalıdır.
- Bu bilgiler afişin üst kısmında ve ortalanmış olarak yer almalıdır.
- Grubun adı ve üyelerinin isimleri Latin alfabesiyle yazılmalıdır. Daha fazla ziyaretçi çekecek şekilde yerleşimini kendi tercihinle belirle.
- Grubun adını belirgin şekilde öne çıkaracak uygun bir çerçeve ekle.
- Organizatörün adı Kiril alfabesiyle ve mavi renkte yazılmalıdır.
- Kendi seçtiğin grafik öğelerini ekle.



2. Bir video oyunu için yaklaşık on satırlık kısa bir senaryo yaz! Metin 12 punto yazı boyutunda ve 1,15 satır aralığıyla yazılmalıdır. Başlık 14 punto yazı boyutunda olmalı ve ortalanmış olmalıdır. Metni, aralarında 6 punto boşluk bulunan iki paragrafa ayır. Sayfanın kenar boşlukları 2 cm olmalıdır. Üst bilgiye video oyununun adını yaz. Alt bilgiye kendi adını ve soyadını yaz. Stile çift tıklamak



AŞAĞIDAKİ KONULARDA ŞUNLARI ÖĞRENECEKSİN:

- Farklı stiller kullanarak metin belgesi oluşturur ve düzenlersin.
- Belgelerdeki içeriği belirli gereksinimlere göre düzenlersin.
- Belgede dizinler oluşturursun.
- Şablonları ve formları kullanırsın.
- Belge koruması uygularsın.





Kaç farklı metin düzenleme yöntemi biliyorsun?
A AŞAĞIDAKİ metin hangi komutlarla düzenlenmiştir?

"Günümüzün **dijital dünyasında**, açık ve profesyonel **iletişim** kurma becerisi güçlü bir yetenektir – fikirlerinizin etkili MESAJLAR hâline gelmesini sağlayan kuralları öğrenmeye **başlayalım**."

Microsoft Word'de stillerle metin düzenleme

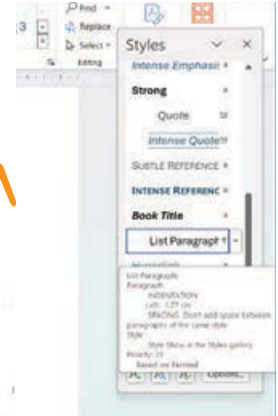
4.1.1 Metin Belgesindeki Stilleri Anlama

Metinle çalışma programlarında stil, metne uygulanan, yazı tipi türü ve boyutu, renk, paragraf hizalama, satırlar arasındaki boşluk ve diğer özellikleri içeren bir dizi düzenleme anlamına gelir. Stillerin kullanılması, belgenin düzenli bir görünüme sahip olmasını, otomatik olarak içindekiler tablosu oluşturulmasını, metnin görünümünün hızlı bir şekilde değiştirilmesini ve belge boyunca tutarlılığın sağlanmasını mümkün kılar. Aynı zamanda stillerin kullanımı, belge biçimlerinin standartlaştırılmasını, kolayca değiştirilmesini, daha iyi erişilebilirlik sağlanmasını ve belgelerin otomatik olarak düzenlenmesini sağlar.

Üç temel stil türü vardır:

- **Paragraf Stilleri**– tüm paragraflara uygulanır (örneğin: Heading 1 (Başlık 1), Normal (Normal veya Temel).
- **Karakter Stilleri**– tek tek kelimelere veya ifadelere uygulanır (örneğin: Emphasis (Vurgulu), Strong (Güçlü vurgulu).
- **Tablo Stilleri**– tüm tablolara uygulanır.

Her stil hakkında ek bilgileri, Styles grubundaki iletişim kutusu başlatıcı düğmesine tıklayarak veya Ctrl+Alt+Shift+S kısayolunu kullanarak açılan stil listesinde bulabilirsin.



Resim 1 Stiller hakkında bilgiler

4.1.2 Mevcut Stillerin Uygulanması

Microsoft Word, Başlık 1, Başlık 2, Alıntı ve diğerleri gibi önceden tanımlanmış çok çeşitli stiller sunar. Bu stiller, metin seçilerek ve Giriş (Home) > Stiller (Styles) sekmesinden uygun stil seçilerek uygulanabilir.



Resim 2 Stil türleri

ÖRNEK

- **Heading 1** – bölümlerin ana başlıkları için kullanılır.
- **Heading 2** – alt başlıklar ve alt konular için kullanılır.
- **Normal** – normal metinler için kullanılır.
- **Quote** – alıntılar veya özel metinleri vurgulamak için kullanılır.
- **Title Subtitle** – belgenin ana başlığı ve alt başlığı için kullanılır.

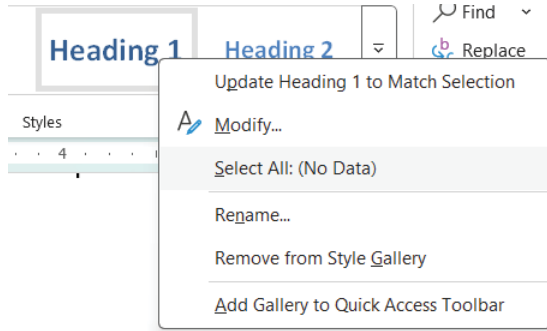
Stillerin Pratik Kullanımı:

1. Biçimlendirmek istediğin paragrafı seç!
2. Styles (Stiller) panelinden bir stil seç!
3. Seçilen stilin otomatik olarak tüm paragrafa uygulanacağını göreceksin.

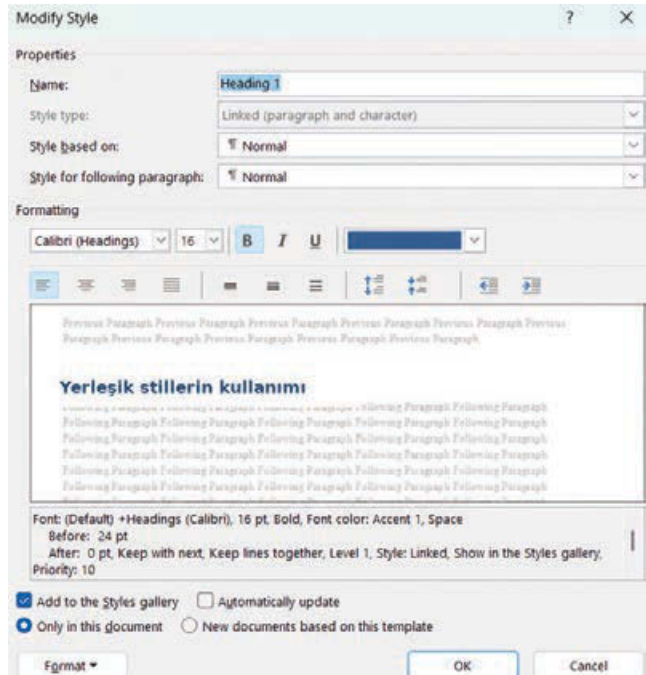
4.1.3 Mevcut Stillerin Düzenlenmesi

Stilin görünümünde küçük değişiklikler yapmak istediğin durumlarda (örneğin farklı bir yazı tipi veya renk kullanmak gibi), her paragrafı tek tek biçimlendirmene gerek yoktur.

Sadece stilin kendisini düzenlemen yeterlidir:



Resim 3 Mevcut stilin düzenlenmesi



Resim 4 Stil değiştirme penceresi

Düzenleme Adımları:

1. Değiştirmek istediğin stilin üzerine sağ tıkla (örneğin Heading 1).
2. Modify... (Değiştir...) seçeneğini seç.
3. Açılan pencerede istediğin değişiklikleri yap: yazı tipi, renk, boyut, aralık, hizalama.
4. OK ile onayladığında, o stili kullanan tüm bölümler otomatik olarak güncellenir.

Bu özellik, tasarımda hızlı değişiklik yapılması gereken çok sayfalı belgeler için özellikle kullanışlıdır.

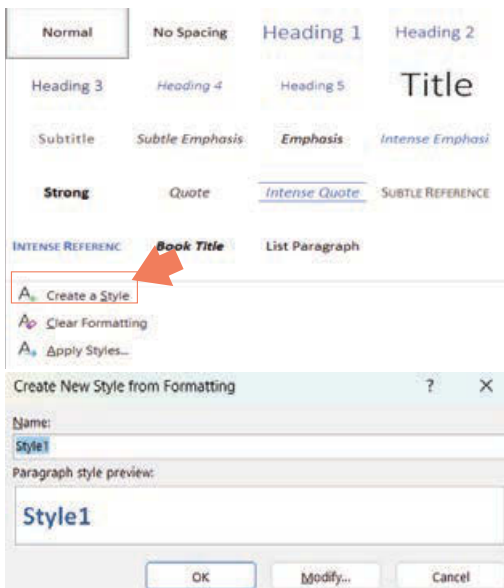
4.1.4 Kendi Stillerini Oluşturma

Belge, mevcut stillerde bulunmayan özel bir görünüme ihtiyaç duyuyorsa kendi stilini oluşturabilirsin.

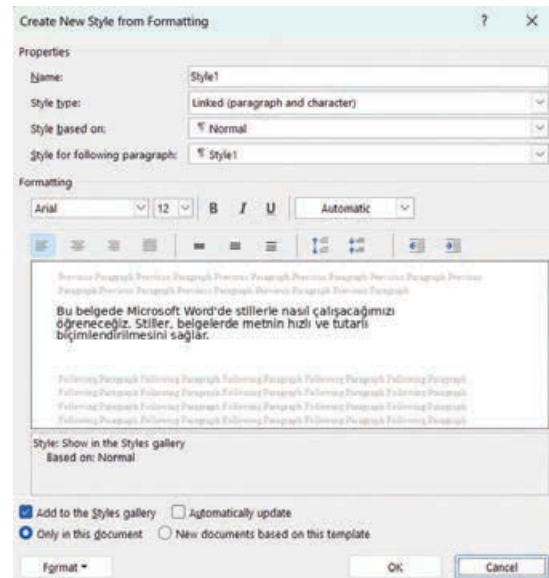
Adımlar:

1. Metni istediğin görünüme göre biçimlendir (yazı tipi, boyut, renk, aralık).
2. Metni seç ve Styles > Create a Style > New Style seçeneklerine tıkla.
3. Stile bir ad ver (örneğin "Gri alt başlık").
4. OK ile onayla ve stil, kullanılabilir stiller listesinde görünecektir.

Kendi stillerini oluşturmak, belirli standartlara sahip belgeler üzerinde çalışırken veya baskı, sunum ya da yayınlama amacıyla belgeler hazırlarken faydalıdır.



Resim 5 Yeni stil oluşturma penceresi



Resim 6 Yeni stil oluşturma penceresi

Name (Ad) alanına, yeni stilin adı yazılır ve bu ad mevcut stiller listesinde görünür. Daha sonra bu stil, belirli gereksinimlere göre Modify (Değiştir) düğmesine tıklanarak değiştirilebilir.

- Style (Stil) alanında, stil türü seçilir (paragraf, karakter, tablo, liste için).
- Based on (Temel alınan) alanında, stilin hangi mevcut biçime dayanacağı seçilir.
- Style for following paragraph (Sonraki paragraf için stil) alanında, sonraki paragrafın hangi stile sahip olacağı belirlenir.
- Formatting (Biçimlendirme) alanında, yazı tipi türü, boyutu, rengi ve stili seçilir.
- Add to Quick Style List (Hızlı Stil Listesine Ekle) düğmesi seçildiğinde, stil galeriye ve stil listesine eklenir.

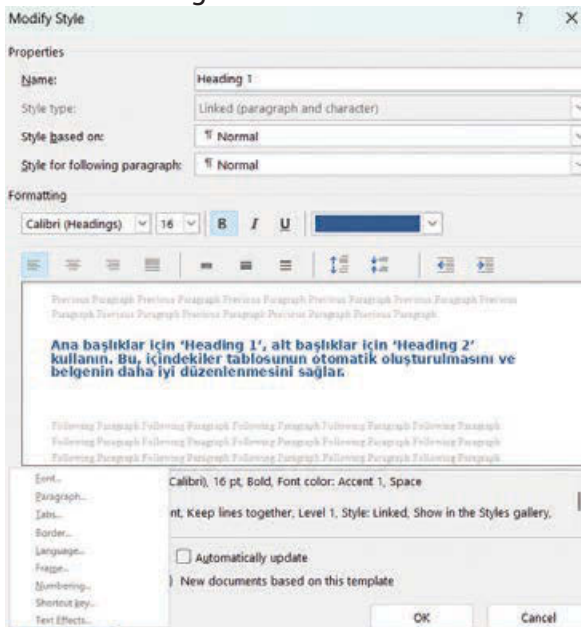
4.1.5 Başlık ve Alt Başlık

Her profesyonel olarak düzenlenmiş belgede başlık, dikkati çeken ilk unsurdur. Genellikle Title (Başlık) stiliyle biçimlendirilir ve şunları içerir:

- Büyük yazı tipi boyutu,
- Ortalanmış hizalama,
- Özel bir renk veya efekt seçimi.

Alt başlık ise başlığı ek açıklamalar veya ayrıntılarla destekler. Subtitle (Alt Başlık) stiliyle biçimlendirilir, genellikle daha küçük bir yazı boyutuna sahiptir, ancak benzer yazı tipi ve stile kullanılır.

Ayrıca ana başlıklar için Heading 1 (Başlık 1), alt başlıklar için Heading 2 (Başlık 2) stillerini kullanabilirsin. Bu kullanım, otomatik içindekiler tablosu oluşturulmasını ve belgenin daha iyi düzenlenmesini sağlar.



Resim 7 Başlık ve alt başlık düzenleme penceresi

“Öğrencilerde dijital okuryazarlığın gelişimi” başlıklı ve “Yapay zekâ çağında yönetim, erişim ve araçlar” alt başlıklı bir metin yaz. Metni, öğrenilen stil ve başlık düzenleme tekniklerini kullanarak biçimlendir.

4.1.6 Neden Stiller Kullanılmalıdı?

Stil kullanımı birçok avantaj sağlar:

- Otomatik içerik oluşturma (içindekiler tablosu),
- Görünümde tutarlılık sağlama,
- Kolay yeniden düzenleme,
- Profesyonel görünüm,
- Daha iyi gezinme (özellikle Navigation Panel (Gezinti Bölmesi) kullanıldığında).

Stillerin uygulanması özellikle raporlar, makaleler, tezler, eğitim kılavuzları veya yayınlar gibi belge türlerinde gereklidir.

Tutarlı Biçimlendirme Önerileri:

- **Elle biçimlendirme yerine stilleri kullanın.**
- **Farklı yazı tipi türlerini ve boyutlarını karıştırmaktan kaçının.**
- **Belgenin birleşik (uyumlu) bir görünümü için temaları kullanın.**

Farklı Belgelerde Stil Örnekler:

- **Raporlar:** Başlıklar, alt başlıklar, metin, alıntılar.
- **Denemeler:** Başlık, giriş, ana bölüm, sonuç.
- **Mektuplar:** Başlık, selamlama, içerik, imza.

4.1.7 SONUÇ

Microsoft Word’de stillerle çalışmak, birden fazla bölüm içeren veya resmî kullanım için hazırlanan belgeler oluşturan herkes için temel bir beceridir. Önceden tanımlanmış stiller kullanılmasının yanı sıra, kullanıcı ihtiyaçlarına göre mevcut stilleri düzenleme veya yeni, özel stiller oluşturma imkânına da sahiptir. Başlık ve alt başlık kavramlarını anlamak ve bunları doğru şekilde uygulamak, belgeyi daha zengin hâle getirir, daha anlaşılır ve görsel olarak daha çekici olmasını sağlar. Stillerin kullanılmasıyla belgeler daha düzenli ve okunması daha kolay hâle gelir.

4.1.8 Pratik Alıştırma: Microsoft Word’de Stillerle Çalışma

Alıştırmanın amacı:

- • Mevcut stilleri uygulamak
- • Stilleri ihtiyaçlara göre düzenlemek
- • Yeni stiller oluşturmak
- • Başlık ve alt başlığı doğru şekilde kullanmak.

1. Belgenin Hazırlanması

1. Yeni bir Word belgesi aç.
2. AŞAĞIDAKİ metni, hiçbir biçimlendirme uygulamadan yaz:

Yapay Zekânın Tarihi

Yapay zekâ (YZ), insan gibi düşünebilen, öğrenebilen ve problemleri çözebilen sistemler oluşturmayı amaçlayan bilgisayar bilimleri dalıdır.

Alt Alan: Makine Öğrenmesi

Makine öğrenmesi, analitik modellerin oluşturulmasını otomatikleştiren bir veri analiz yöntemidir.

Alt Alan: Derin Öğrenme

Derin öğrenme, görüntü ve ses gibi karmaşık verileri analiz etmek için çok katmanlı sinir ağlarını kullanır.

Sonuç

Yapay zekâ hızla gelişmektedir ve toplumu dönüştürme konusunda büyük bir potansiyele sahiptir.

2. Mevcut Stillerin Uygulanması

1. Yapay zekânın tarihi" ifadesini seç → Title (Başlık) stilini uygula.
2. "Alt alan: Makine öğrenmesi", "Alt alan: Derin öğrenme" ve "Sonuç" ifadelerini seç → Heading 2 (Başlık 2) stilini uygula.
3. Paragraflardaki diğer metinler için Normal stilini kullan.

3. Mevcut Stilin Düzenlenmesi

1. Heading 2 üzerine sağ tıkla → Modify...
2. Şunları değiştir:

Metin rengi: koyu mavi,

Boyut: 14 punto,

Kalın (Bold) yap,

Sola hizala.

3. OK ile onayla.

4. Kendi Stilini Oluşturma

1. "Yapay zekâ (YZ)..." ile başlayan paragrafı seç.
2. Styles (Stiller) sekmesinde Create a Style → New Style seçeneklerini seç.
3. Ad: Giriş
4. Şunları seç:
 - Yazı tipi: Georgia, 12 punto
 - İtalik
 - Satır aralığı: 1,5
 - Justify (İki yana yasla) hizalama.
5. OK ile onayla ve stili uygula.

5. Ek Görev (isteğe bağlı)

Belgenin başına otomatik bir içindekiler tablosu ekle
References > Table of Contents > Automatic Table 1



UYGULAMALI ETKİNLİK:

1. Yeni bir Word belgesi aç ve üç başlık ile birkaç paragraftan oluşan kısa bir metin yaz. "Heading 1", "Heading 2" ve "Normal" stillerini uygula!
2. "Benim stilim" adıyla yeni bir stil oluştur. Bu stil, mavi yazı tipi, 14 punto büyüklük ve ortalı hizalama kullanmalıdır. Bu stili bir paragrafa uygula!
3. "Heading 1" stilini, kırmızı renk ve kalın (Bold) yazı tipi kullanacak şekilde düzenle! Bu stili belgedeki tüm ana başlıklara uygula!
4. Uyguladığın stillere dayalı olarak otomatik bir içindekiler tablosu oluştur!



1. Microsoft Word'de stiller panelini hangi sekme içerir
 - a) Insert.
 - B) Home.
 - c) Layout.
 - d)Review.

2. Bir metne "Heading 1" stili uygulandığında ne olur?
 - a) Metin tabloya dönüştürülür.
 - b) Ana başlık için biçimlendirme uygulanır.
 - c) Metin silinir.
 - d) Resim eklenir.

3. Mevcut bir stili düzenlemenin doğru yolu hangisidir
 - a) Stile çift tıklamak.
 - b) Sağ tık > Modify.
 - c) Insert tıkla > Style.
 - d) File tıkla > Options.

4. Belgelerde stil kullanımı ne sağlar?
 - a) Sadece biçimlendirme sağlar.
 - b) Tutarlı görünüm ve otomatik içindikiler sağlar.
 - c) Daha fazla hata oluşturur.
 - d) Sadece renk değişikliği yapar.

5. Microsoft Word'de stil nedir ve neden kullanışlıdır!
6. Word'de kendi stilini nasıl oluşturabilirsin?
7. Başlık ve alt başlıklarda stil kullanmanın avantajları nelerdir?
8. Bir belgede birden fazla yerde uygulanmış olan bir stili değiştirirsen ne olur?

4.2

İçindekiler ve Dizinler

Daha uzun belgeler (ödevler, denemeler, projeler, kitaplar) üzerinde çalışırken, bu belgelerin düzenli ve kolay gezilebilir olması çok önemlidir. Bu amaçla Microsoft Word, içindekiler tablosu ve izin oluşturmak için araçlar sunar. Bunlar, metindeki stiller ve işaretlemelere dayanarak otomatik olarak oluşturulur ve güncellenir.

4.2.1 İçindekiler Tablosu Nedir?

İçindekiler tablosu, bir belgedeki tüm ana bölümlerin ve alt başlıkların listesidir ve genellikle belgenin başında yer alır. Okuyucu için bir rehber görevi görür ve istenilen bölüme hızlıca geçiş yapılmasını sağlar.

İçindekiler tablosu nasıl oluşturulur?

İçindekiler tablosu oluşturabilmek için, öncelikle belgedeki tüm başlık ve alt başlıklara başlık stilleri (Heading 1, Heading 2 vb.) uygulanmalıdır.

AŞAĞIDAKİ adımları izleyerek içindekiler tablosu oluşturabilirsin:

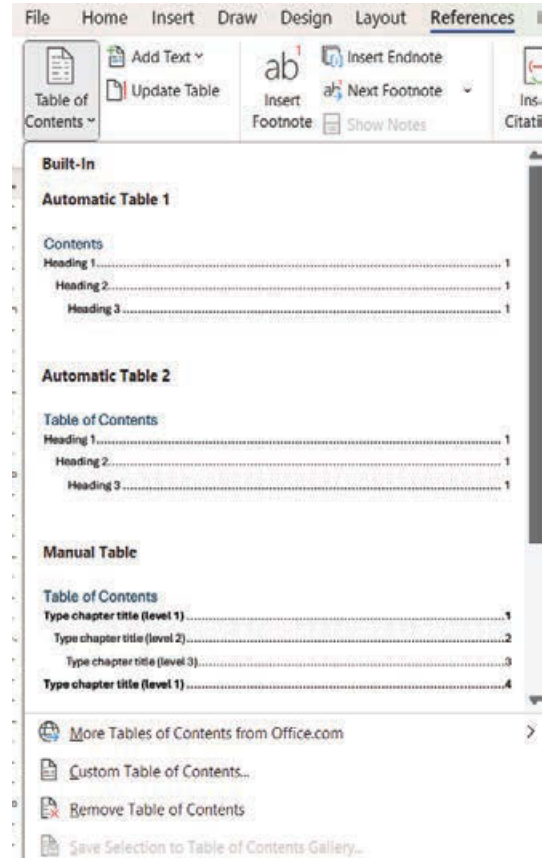
1. İmleci, içindekiler tablosunun görünmesini istediğin yere yerleştir (örneğin belgenin başına).
2. Araç çubuğundan References (Başvurular) → Table of Contents (İçindekiler) seçeneğini tıkla
3. Sunulan listeden bir içindekiler stili seç (Automatic Table 1 veya Automatic Table 2).
4. Word, başlık stilleriyle biçimlendirilmiş tüm öğelerden otomatik olarak bir içindekiler tablosu oluşturur.

Oluşturulan içindekiler tablosu, sayfa numaralarını ve hiper bağlantıları içerir. Böylece belgede istenen başlığa kolayca gidilebilir.



Öneri:

Belgenin tamamında başlıklar için stilleri tutarlı bir şekilde kullanın, böylece içindekiler tablosunu düzenlemek daha kolay olur. Ayrıca sayfalar numaralandırılmış olmalıdır. İçindekiler tablosu oluşturulurken program başlıkları bulur ve bunları ilgili sayfa numaralarıyla birlikte içindekiler tablosuna otomatik olarak ekler.



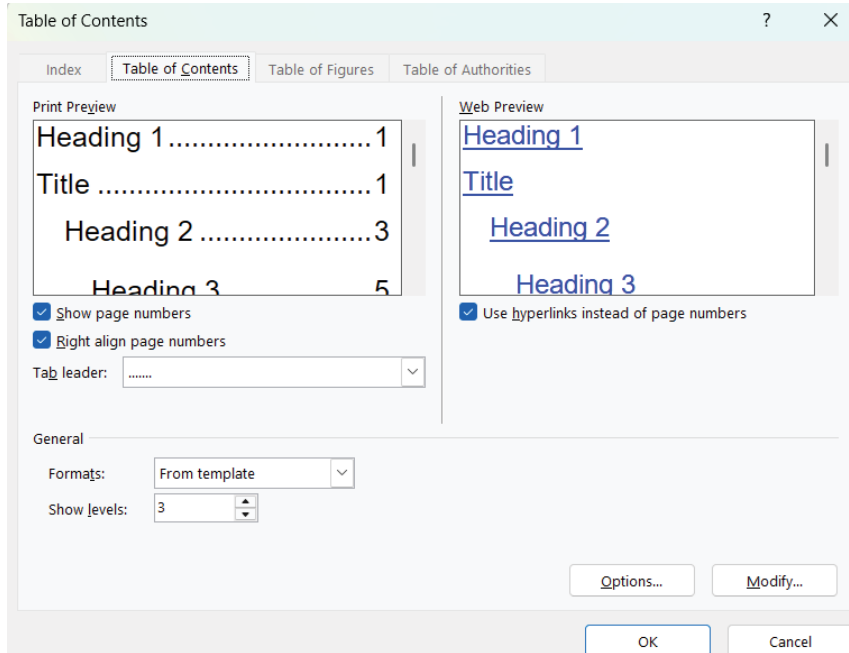
Resim 7

Belgede içindekiler tablosu oluşturma

4.2.2 İçindekiler Tablosunu Özelleştirme

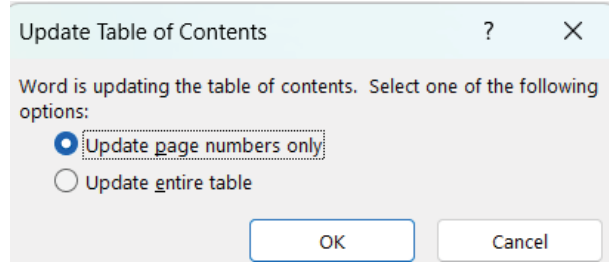
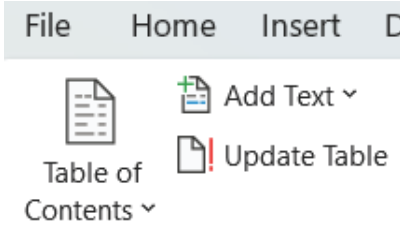
İçindekiler tablosu, ihtiyaçlara göre AŞAĞIDAKİ şekillerde özelleştirilebilir:

- Daha fazla alt başlık düzeyi eklenebilir (örneğin Heading 3).
- Görünümünü değiştirmek için References (Başvurular) → Table of Contents (İçindekiler) menüsünden Custom Table of Contents (Özel İçindekiler Tablosu) seçilir. Böylece Şekil 8'de gösterilen pencere açılır.
- Show page numbers (Sayfa numaralarını göster) seçeneği ile sayfa numaralarının gösterilip gösterilmeyeceği belirlenir.
- Right align page numbers (Sayfa numaralarını sağa hizala) seçeneği ile sayfa numaralarının sağ tarafa hizalanıp hizalanmayacağı belirlenir.
- Tab leader listesinden, sayfa numaralarına kadar uzanan kılavuz çizginin stili seçilir (boşluk, düz çizgi, noktalar veya kısa çizgiler).
- Format alanından içindekiler tablosunun biçimi seçilir.
- Show levels (Düzeyleri göster) alanında, başlıkların hangi düzeye kadar gösterileceği belirlenir.
- Belge bir web sitesinde yayımlanacaksa, içindekiler tablosundaki öğeler köprü (hiper bağlantı) olarak düzenlenmelidir. Bunun için Use hyperlinks instead of page numbers (Sayfa numaraları yerine köprüleri kullan) seçeneği işaretlenir.
- Son olarak yapılan ayarlar OK (Tamam) düğmesine tıklanarak onaylanır.



Resim 8 Belgede içindekiler tablosunu özelleştirme

İçindekiler tablosu, metinde yapılan değişikliklerde Update Table seçeneği ile güncellenebilir. Sadece sayfa numaraları güncellenecekse Update page numbers only seçilir veya tam değişiklik yapmak için Update entire table seçeneği kullanılır.



Resim 9 İçindekiler tablosunu güncelleme

İyi düzenlenmiş bir belge şunlara sahip olmalıdır:

- Açık bir yapı: Başlık → alt başlık → metin.
- Bilgi hiyerarşisi: Ana konu (Heading 1), alt konular (Heading 2), alt-alt konular (Heading 3).
- Gerektiğinde boşlukların, numaralandırmaların ve görsellerin kullanılması.

4.2.3 Dizin Nedir?

Dizin, belgede geçen anahtar kelimelerin veya kavramların, görüldükleri sayfa numaralarıyla birlikte belgenin sonunda gösterilen listesidir. Genellikle kitaplarda, ansiklopedilerde veya bilimsel çalışmalarda kullanılır.

A		G	
Alan adı.....	3	Giga.....	5
D		Güncelleme.....	1
Dosya.....	3, 4	Güvenlik.....	6
S		Y	
Sistem.....	4	Yazılım.....	5
Sürücüler.....	2		

Resim 10 Dizin

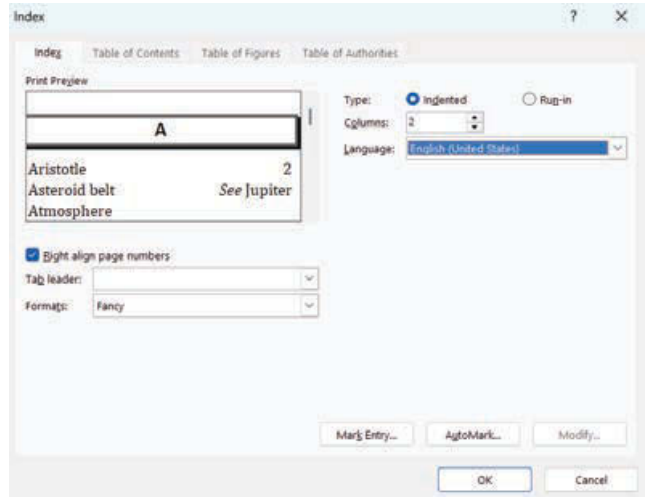
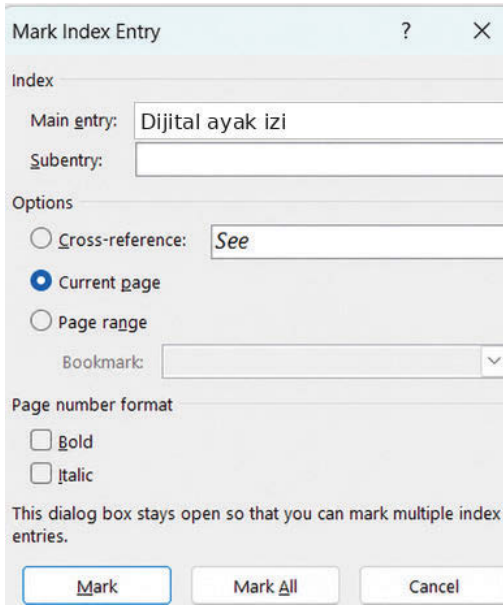
İndeks listesi veya tablosu nasıl eklenir?

- İndeksin bir parçası olması gereken kelimeyi veya kavramı seç.
- References sekmesini seç ve ardından Mark Entry seçeneğini seç.
- "Main entry" alanında işaretlenen kelime görünür.
- Birden fazla kez geçiyorsa Mark veya Mark All seçeneğine tıkla. MS Word, belgeye özel bir XE alanı (indeks ögesi) ekler. Bu alan, yeni satır işareti gibi ¶, belgenin yazdırılması sırasında görüntülenmez.

Dijital ayak izi { XE "Dijital ayak izi" } ¶

- Daha sonra imleci belgenin sonuna yerleştir.
- References → Insert Index seçeneğini seç.
- Görüntülenecek formatı, kılavuz çizgi türünü ve sütun sayısını seç, ardından OK düğmesine tıklayarak seçimi onayla.

İndeks, her terim için sayfa numaralarıyla birlikte alfabetik olarak görüntülenir.



Resim 11 İndeks oluşturma

Yeni kelimeler eklenirse indeks, Update Index seçeneği seçilerek güncellenebilir.

Bir kelimenin veya kavramın işaretlemesini kaldırmak için, XE işareti (süslü parantezler {} ve parantez içindeki kelime ile birlikte) seçilir ve Delete düğmesine tıklanır.



4.2.4 Pratik Etkinlik

Ödev: 1

1. Üç başlık ve alt başlık içeren kısa bir metin yaz.
2. Heading 1 ve Heading 2 stillerini uygula.
3. Başlangıca bir içindekiler tablosu ekle.
4. Üç kelimeyi indeks olarak işaretle ve belgenin sonuna indeks ekle.
5. Belgede değişiklikler yap ve hem içindekiler tablosunu hem de indeksi güncelle.

Ödev: 2

1. Microsoft Word'de şu başlığa sahip yeni bir belge oluştur: „Dijital okuryazarlık ve eğitimde BİT kullanımı”
2. Belgeyi AŞAĞIDAKİ bölümlerle (kısımlarla) yapılandır:

➡ Belge başlığı:

- „Dijital okuryazarlık ve eğitimde BİT kullanımı” → Heading 1
- Giriş → Heading 1

- Dijital okuryazarlık nedir? → Heading 2
- Okullarda kullanılan BİT araçları → Heading 2
- Derslerde teknoloji kullanımının avantajları ve zorlukları → Heading 2
- SONUÇ → Heading 1

Her bölüm yaklaşık dört ila beş cümlelik bir metin içermelidir.

1. Her başlığa uygun stilleri uygula (Heading 1, Heading 2).

2. Belgenin başına otomatik bir içindekiler tablosu ekle:

- References → Table of Contents → Automatic Table.

3. AŞAĞIDAKİ kelimeleri indeks için işaretleyin:

- Teknoloji, öğrenci, dijital okuryazarlık
- Her kelime için: References → Mark Entry → Mark All

4. Belgenin sonuna indeks ekle:

- References → Insert Index

5. Belgeyi şu isimle kaydet: "İndekslenmiş_belge_Adın.docx"

Öneri:

- Metne yeni bir başlık veya alt başlık eklediğinde, içindekiler tablosunu güncellemeyi unutma: İçindekiler tablosuna tıkla → Update Table → Update entire table



SONUÇ

Microsoft Word'de içindekiler ve indekslerle çalışma, metin belgelerinin daha iyi düzenlenmesini, daha anlaşılır olmasını ve profesyonel bir görünüme sahip olmasını sağlar. Başlık stillerini kullanarak, belgede yapılan değişikliklere göre güncellenen otomatik bir içindekiler tablosunu hızlı bir şekilde oluşturabiliriz. İndeks ise okuyucuların anahtar kelimeleri ve konuları kolayca bulmasına yardımcı olur. Bu araçlar, günlük çalışmalarda metin belgeleriyle daha verimli çalışmayı sağlar.

BİLGİ KONTROLÜ SORULARI



1. Word'de bir içindekiler tablosu oluşturulurken ne yapılır?
2. Başlıklar için stilleri kullanmak neden önemlidir?
3. Bir kelime indeks için nasıl işaretlenir?
4. İçindekiler tablosu ve indeks listesi ne zaman ve nasıl güncellenir?
5. Bir indeks tablosu oluşturmadan önce ne yapmalısın?
6. Belgenin hangi bölümü genellikle indeks listesini içerir?

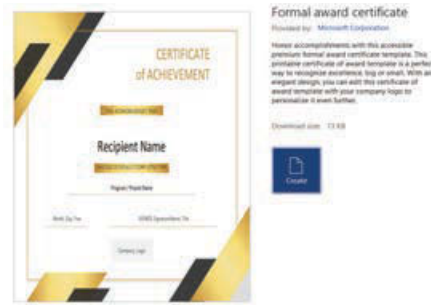
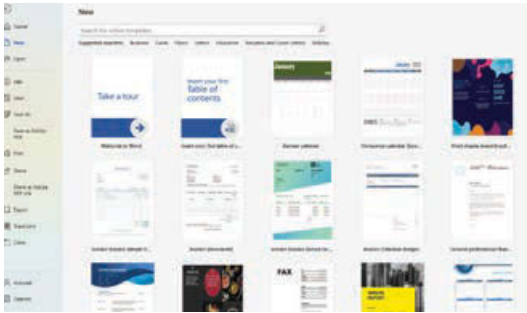
4.3

Şablonlar ve Formlar

Tekrarlanan belgelerle çalışırken — örneğin başvurular, talepler, sözleşmeler veya raporlar gibi — şablonları kullanabiliriz. Şablonlar (templates), belirli biçimleri, yapıları ve özelleştirilebilen içerikleri bulunan önceden hazırlanmış belgelerdir. Bunun dışında Word, standart bir şekilde veri girişi yapmak için kullanılan formlar oluşturma olanağı da sağlar.

4.3.1 Şablon

Yeni bir belge açarken, MS Word çok sayıda hazır şablon kategorisi sunar (örneğin, mektuplar, davetiyeler, sertifikalar, seminer çalışmaları için kapak sayfaları ve diğerleri için). File → New seçenekleri seçilerek oluşturulur, bununla sağ tarafta kategori koleksiyonu görünür. **Search** seçeneği ile çevrimiçi şablon aranabilir. Şablonlar kategoriye tıklanarak açılır, böylece şablon seçenekleri açılır, istenilen şablon seçilir ve **Create** düğmesine tıklanır. Tanımlanmış stillere sahip yeni bir belge açılır ve bu belge kişisel ihtiyaçlara veya belirli gereksinimlere göre düzenlenebilir. Normal bir belgeden farklı bir uzantıya sahiptir, .dotx ve Documents → Custom Office Templates klasöründe özel bir klasörde saklanır.



Resim 12 Şablon oluşturma (MS Word)

Adımlar: Bir şablon nasıl kullanılır?

1. Word'ü aç → File → New.
2. Pencereden ilk yardım atölyesi için bir çalıştay gündemi şablonu seç.
3. Create düğmesine tıkla ve belgeyi ihtiyaçlarına göre düzenle.
4. Düzenleme ve yazma işlemini tamamladığında, belgeyi „Çalıştay Gündemi“ adıyla kaydet.

Adımlar: Kendi şablonun nasıl oluşturulur?

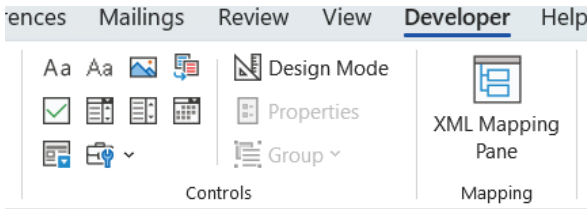
1. Yeni bir belge oluştur.
2. Metin, başlık, biçimlendirme ve tekrar kullanılan öğeleri ekle.

3. Seç: File → Save As.
4. Save as type bölümünden şunu seç: Word Template (.dotx).
5. Bir isim ver (örneğin „Rapor için şablonum“) ve kaydet.
6. Yeni belge otomatik olarak Custom Office Templates klasörüne kaydedilecektir.

4.3.2 Formlar

Formlar, isimler, tarihler, adresler ve sorulara verilen cevaplar gibi verilerin girilmesi için alanlar içeren bir belgedir ve anketler, araştırmalar, başvuru formları vb. oluşturarak veri toplamak için kullanılır. Metin kontrol alanı, seçim düğmesi, onay kutusu alanları ve veri girilebilen açılır menüler gibi önceden tanımlanmış alanlar içerir. Ayrıca, değiştirilemeyen metinler de içerir, örneğin anket soruları veya başvuru gereksinimleri (Doğum tarihini giriniz, ad ve soyad, favori müzik grubu vb.).

Formlar, yeni bir belge açılarak ve Developer sekmesi seçilerek oluşturulur. Daha sonra Controls menüsünden formun içermesi gereken alanlar seçilir. Eğer Developer sekmesi sekmeler şeridinde yoksa, şu adımlar ile ekleyebilirsiniz: File → Options bölümüne girilir ve burada Customize Ribbon seçilir. Pencerenin sağ tarafında Developer seçilir ve OK düğmesine tıklanır.



Resim 13 Formda alan oluşturma (MS Word)

1. Metin Kontrol Alanı (Text Control Field)

Bu alan, kullanıcıların forma metin girmesine olanak sağlar. Örneğin, bir kişi adını veya adresini doldurduğunda, bu işlem metin kontrol alanında yapılır. Şuradan eklenir: Developer → Controls → Rich Text Content Control veya Aa düğmesine tıklanır.

2. Seçim Düğmesi (Option Button veya Radio Button)



Bu, birden fazla seçenek arasından yalnızca birinin seçilmesi gerektiğinde kullanılan yuvarlak düğmedir. Genellikle anketlerde kullanılır (örneğin: Cinsiyet: → erkek → kadın) düğmesine tıklanarak eklenir.

3. Açılır Menü (Drop-down list)



Açılır menü, okla tıklandığında açılan seçim listesidir. Kullanıcı önceden tanımlanmış seçeneklerden birini seçebilir (örneğin: şehirler, meslek, sınıf) düğmesine tıklanarak eklenir.

4. Seçim Alanı (Check Box)



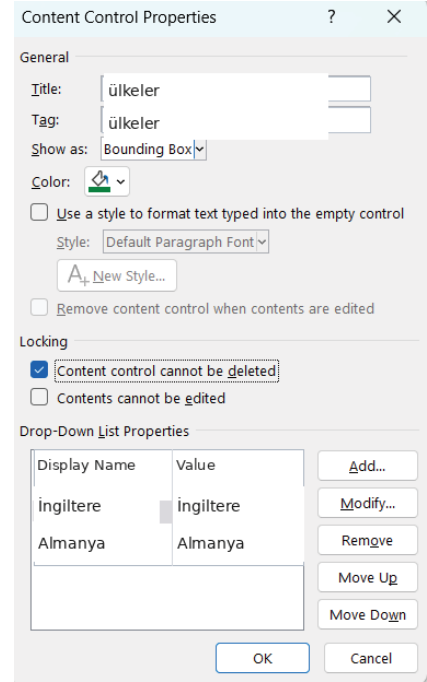
Birden fazla seçeneğin seçilmesi için kullanılır (örneğin ilgi alanları: okuma, spor, müzik) düğmesine tıklanarak eklenir.

5. Açılır Menü (Drop-down List)



Önceden tanımlanmış seçeneklere sahip açılır menüdür ve kullanıcılar sunulan seçeneklerden kendilerine en uygun olanı seçerler düğmesine tıklanarak eklenir.

Yani **Drop-Down List Content Control** seçilir, daha sonra seçenek eklemek için **Properties** seçilir ve Add düğmesine tıklanır (Şekil 13). Açılır menüye bir isim de ekleyebilirsin. Girilen seçenekleri Modify düğmesi ile değiştirebilir, Remove ile kaldırabilir ve sıralamayı Move up ve Move down düğmeleri ile değiştirebilirsin. Kullanıcının alanı silmemesi için Content control cannot be deleted seçeneğini işaretler. Eğer Content control cannot be edited seçeneği işaretlenirse, kullanıcı hiçbir seçim yapama.



Resim 14 Forma seçenek ekleme (MS Word)

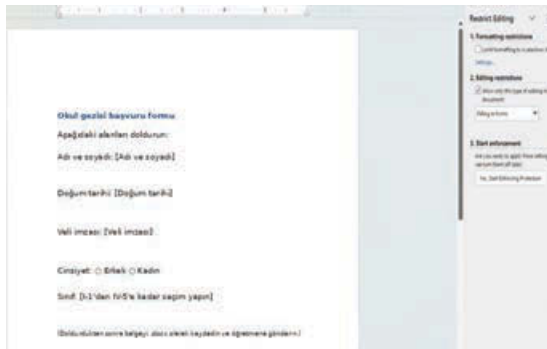
6. Tarih Seçme Alanı



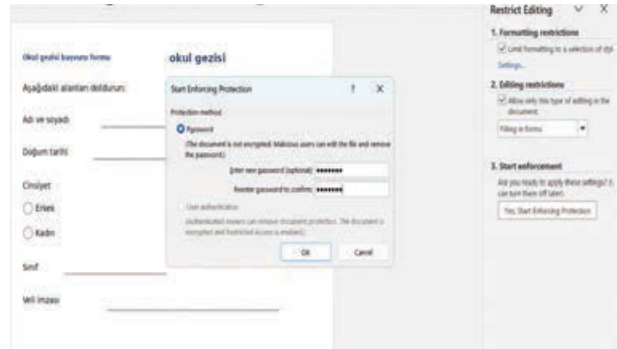
Bu alana seçilecek tarihler girilir, örneğin toplantının yapılacağı tarih, gezi günlerinin seçimi vb.

Formun Kilitlenmesi

Formu tasarladıktan sonra: Developer > Restrict Editing seç. "Editing restrictions" bölümünde Filling in forms seçeneğini seç. Böylece kullanıcılar sadece cevap yazabilirler. Daha sonra şifre belirlemek için Yes, Start Enforcing Protection seçeneğine tıkla. Bu işlem, kullanıcıların formu doldurmasına olanak sağlar ancak tasarımını değiştirmelerini engeller.



Resim 15 Şifre olmadan form kilitleme (MS Word)



Resim 16 Şifre ile form kilitleme (MS Word)

Şablon Olarak Kaydetme

File > Save As seç, daha sonra belge türü olarak Word Template (*.dotx) seç ve Save düğmesine tıkla. Bu, formu birden fazla kez kullanmanı sağlar.



SONUÇ

Word'de şablonların ve formların kullanılması, standartlaştırılmış belgelerin oluşturulmasını kolaylaştırır. Şablonlar zaman kazandırır ve tutarlı bir görünüm sağlar, formlar ise bilgilerin doğru ve düzenli bir şekilde girilmesi için yapı oluşturur. Bu araçları öğrenen öğrenci, gerçek eğitimsel ve idari durumlara uygun, ileri düzeyde belgeler hazırlayabilecek beceriye sahip olur.



UYGULAMALI ETKİNLİK

Öğrencinin okul etkinliğine başvuru formu oluşturma

1. Word'de yeni bir belge aç.
2. Başlık ekle: „Okul gezisi başvuru formu“.
3. AŞAĞIDAKİ alanları ekle:
 - ↪ Ad ve soyad → Text Field
 - ↪ Sınıf → Drop-down list (I-1'den IV-5'e kada)
 - ↪ Cinsiyet → Option Buttons (Erkek / Kadın)
 - ↪ Doğum tarihi → Text Field
 - ↪ Veli imzası → Text Field
4. Restrict Editing ile korumayı etkinleştir, böylece sadece alanların doldurulmasına izin verilsin.
5. Belgeyi şablon olarak kaydet: Save as → Word Template (.dotx)



UYGULAMALI ETKİNLİK

1. Microsoft Word'de yeni bir belge aç.
2. File > Options > Customize Ribbon üzerinden „Developer“ sekmesini etkinleştir ve „Developer“ seçeneğini işaretle.
3. AŞAĞIDAKİ öğeleri içeren bir etkinlik başvuru formu oluştur:

Form şu alanları içermelidir:

- ↪ Metin alanları: İsim, Soyisim, E-posta
- ↪ Şehir seçimi için açılır menü (ör. Üsküp, Manastır, Kalkandelen)
- ↪ Cinsiyet için seçim düğmeleri (Radio Buttons): Erkek / Kadın
- ↪ Yorum alanı (Text Area)
- ↪ İlgi alanları için seçim kutusu (Check Box): Teknoloji, Sanat, Spor

BİLGİ KONTROLÜ SORULARI



1. Şablon nedir ve ne amaçla kullanılır?
2. Form, normal bir belgeden nasıl farklıdır?
3. Metin kontrolleri ve açılır menü nerede bulunur?
4. Özel bir şablon nasıl kaydedilir?
5. Seçim düğmesi ile açılır menü arasındaki fark nedir?
6. Word'de form oluşturma araçları nerede bulunur?
 - a) File
 - b) Developer
 - c) Insert
 - d) View
7. Kullanıcıdan metin girişi almak için hangi alan kullanılır?
 - a) Drop-down
 - b) Text Control
 - c) Check Box
 - d) Table
8. Sadece bir seçeneğin seçilmesi gerektiğinde hangi düğme kullanılır?
 - a) Check Box
 - b) Drop-down
 - c) Radio Button
 - d) Text Field
9. Açılır menü ne sağlar?
 - a) Serbest metin girişi
 - b) Sunulan seçeneklerden seçim yapma
 - c) Resim ekleme
 - d) Yazdırma
10. Bir formu doldurulabilir ancak düzenlenemez şekilde kilitleyebilir miyiz?
 - a) Hayır
 - b) Evet
11. Word'de Developer sekmesinin görevini açıklayın!
12. Bir formda hangi kontrol türleri kullanılabilir?
13. Bir formun doldurma işlemi için nasıl kilitlendiğini açıklayın!

Word belgelerinin korunması, yetkisiz kişilerin belgeyi okumasını, düzenlemesini veya içeriği silmesini önlemek için önemlidir. Microsoft Word, belgeleri korumak için birkaç farklı yöntem sunar.

Koruma Türleri

1. Sadece okuma (Read-only):

Bu mod, belgenin açılmasına izin verir ancak belge sahibinin izni olmadan düzenlenmesine izin vermez. Kullanıcı, belgenin sadece okunabilir olduğu konusunda bilgilendirilir.

2. Şifre ile koruma (Password protection):

Belgeyi açmak veya düzenlemek için bir şifre belirlenebilir. Doğru şifre olmadan belge açılmaz veya değiştirilemez.

3. Düzenleme kısıtlaması (Editing restrictions):

Yalnızca belirli düzenleme türlerine izin verilmesini sağlar, örneğin form doldurma, yorum ekleme veya değişiklikleri izleme.

Koruma Ayarlama Adımları

1. Belgeyi Microsoft Word'de aç.

2. 'File' > 'Info' sekmesine git.

3. 'Protect Document' seçeneğine tıkla.

4. AŞAĞIDAKİ seçeneklerden birini seç:

- 'Always Open Read-Only'
- 'Encrypt with Password'
- 'Restrict Editing'

5. Şifre veya kısıtlama ayarlamak için talimatları takip et.

6. Belgeyi kaydet.

BİLGİ KONTROLÜ SORULARI VE ETKİNLİKLER



1. Word belgesini açmak için şifre koyulabilir mi? (Evet/Hayır)
2. 'Read-only' modu düzenlemeye izin verir mi? (Evet/Hayır)
3. Bir belgede yalnızca form doldurmaya izin verilebilir mi? (Evet/Hayır)
4. Word'de bir belgeye nasıl şifre konulduğunu açıklayın!
5. Düzenleme kısıtlamasının avantajları nelerdir?
6. Hangi durumlarda 'Read-only' modunu kullanırdınız?



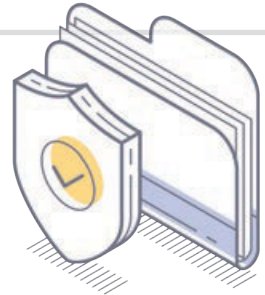
UYGULAMALI ETKİNLİK

1. "Projem" başlıklı bir belge oluşturun. Kısa bir metin yaz. Daha sonra belgeye şu koruma ayarlarını uygulayın:
2. Belge sadece okunabilir (Read-only) olacak.
3. Belgeyi açmak için şifre belirlenecek.
4. Yalnızca form doldurmaya izin verilecek.
5. Belgeyi kaydet ve şifre olmadan açmayı dene.



SONUÇ

Word'deki koruma araçlarını kullanarak belgelerimizin gizliliğini ve bütünlüğünü koruyabiliriz. Farklı koruma türlerinin nasıl ayarlanacağını bilmek ve bunları pratik durumlarda uygulamak önemlidir.



Öğrendiklerimiz

Microsoft Word'de belgelerle çalışmak sadece metin yazmaktan ibaret değildir – düzenleme, organizasyon, otomasyon ve içeriğin korunması ile ilgili becerileri de kapsar. Bu konu boyunca öğrenciler birkaç önemli araç ve kavramla tanıştılar.

Stiller

Stillerin kullanılmasıyla belgeler profesyonel bir görünüm ve tutarlılık kazanır. Stiller, başlıkların, alt başlıkların ve metinlerin hızlı biçimde biçimlendirilmesini, ayrıca otomatik olarak içerik tablosu oluşturulmasını sağlar.

İçerik ve İndeksler

Otomatik içerik tablosu (Table of Contents) ve indeksler, büyük belgelerde gezinme ve düzenleme için kullanışlıdır. Bunlar stillere dayanır ve kolay arama ile daha iyi görünürlük sağlar.

Şablonlar

Şablonlar (templates), zaman kazandıran ve belgelerde birleşik bir görünüm sağlayan önceden tanımlanmış biçimlerdir. Raporlar, mektuplar, formlar ve diğer amaçlar için farklı şablonlar kullanılır.

Formlar

Formlar, doldurulabilir alanlara sahip etkileşimli belgelerdir. Metin alanları, açılır menüler ve seçim düğmeleri gibi kontrollerin kullanılmasıyla verilerin doğru şekilde girilmesi sağlanır.

Belgelerin Korunması

Belgelerin korunması, bilgilerin güvenliği ve bütünlüğü için önemlidir. Şifre belirleyerek diğer kişilerin düzenleme yapması sınırlandırılır, belge yalnızca okunabilir hale getirilebilir.



KONU 4: GENEL TEKRARLAMA

1. Metin belgesinde “stil” neyi ifade eder? En az iki örnek veriniz (örneğin: Heading 1, Normal)!
2. Metni manuel olarak biçimlendirmek yerine stilleri kullanmanın neden yararlı olduğunu açıklayınız!
3. Öğretmen tüm başlıkların aynı stile sahip olmasını isterse belgeyi nasıl düzenlersiniz? Adımları açıklayınız!
4. Bir belgede terimlerden oluşan bir liste var. Her anahtar kelimeye yönlendiren bir indeks nasıl oluşturulur?
5. Şablon (template) ve form (form) özelliklerini karşılaştırınız. Hangi yönlerden benzerdirler ve hangi yönlerden farklıdırlar?
6. AŞAĞIDAKİ durumu analiz ediniz: Bir belgede farklı yazı tipleri, metin renkleri ve boyutları bulunmaktadır. Belgenin düzenli görünmesi için stilleri nasıl düzenlersiniz?
7. Hassas bilgiler içeren bir belgeyi koruma olmadan e-posta ile göndermek güvenli midir? Kararınızı açıklayınız!
8. Belirli bir durumda hangi koruma yönteminin daha etkili olduğunu değerlendiriniz: şifre, düzenleme kısıtlaması veya sadece okuma modu!
9. Üç farklı stil (başlık, alt başlık ve metin) kullanarak kısa bir belge oluşturunuz ve neden bu stilleri seçtiğinizi açıklayınız!
10. Bir okul raporu için kendi şablonunuzu oluşturunuz! Başlık, yazar, tarih ve indeks için alan ekleyiniz! Nasıl kullanacağınızı açıklayınız!

KONU 5: ELEKTRONİK TABLO HESAPLAMA PROGRAMI

Elektronik tablo hesaplama programları, verilerin sistematik olarak düzenlenmesini, analiz edilmesini ve görselleştirilmesini sağlayan güçlü dijital araçlardır. Esneklikleri ve işlevsellikleri sayesinde günlük yaşamın ve profesyonel hayatın farklı alanlarında – eğitim, iş dünyası, bilimsel araştırmalar ve yönetim gibi alanlar dâhil olmak üzere – yaygın olarak kullanılırlar. Bu programlar, bilgilerin kolayca işlenmesini, formüllerin oluşturulmasını ve otomatik hesaplamaların yapılmasını sağlar, ayrıca grafikler ve tablolar aracılığıyla görsel sunum imkânı verir. Matematiksel işlemlerin yanı sıra sıralama, filtreleme, koşullu biçimlendirme ve veri analizi gibi gelişmiş özellikleri de desteklerler. Böylece daha verimli çalışmaya ve bilinçli kararlar alınmasına önemli katkı sağlarlar. En yaygın kullanılan elektronik tablo hesaplama programları arasında Microsoft Excel, Google Sheets ve LibreOffice Calc bulunmaktadır. Bu ders kitabında, Microsoft 365'in bir parçası olan Microsoft Excel'in işlevleri açıklanacaktır.

Yeni Kavramlar

- Tablo, satırlar, sütunlar, hücreler, çalışma sayfası, ekleme, silme, yeniden adlandırma, çalışma sayfasını taşıma ve kopyalama, formül, işlem, grafik, grafik türleri, veri sıralama, veri filtreleme, pivot tablo, hücre koruma, çalışma kitabı koruma.

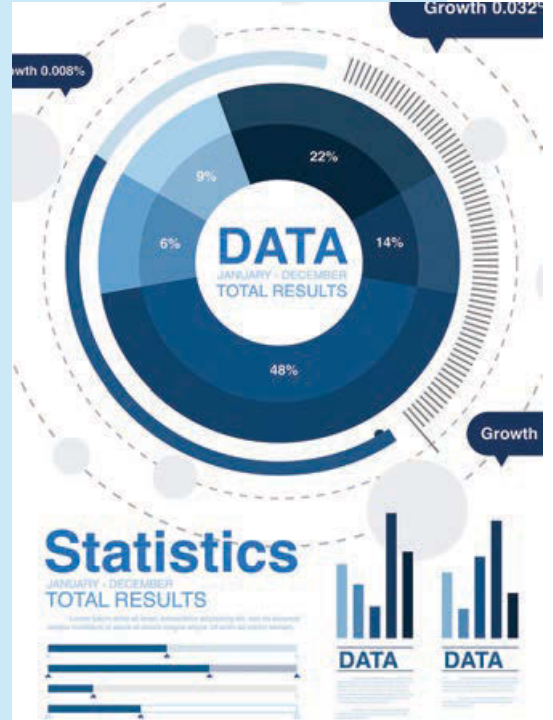
Bu konuyu tamamladığında...

- ✓ Farklı formatlarda veri girersin
- ✓ Basit tablolar oluşturur ve düzenlersin.
- ✓ Formül çubuğunu kullanarak elektronik tablo hesaplamaları için basit formüller oluşturursun.
- ✓ Uygun menü aracılığıyla tablodaki verilerle hesaplama yapmak için basit işlevleri kullanırsın.
- ✓ Farklı türlerde grafikler oluşturursun.
- ✓ Tablodaki verileri filtrelersin.
- ✓ Tablodaki verileri sıralarsın.



AŞAĞIDAKİ İÇERİKLERDE ŞUNLARI ÖĞRENECEKSİN:

- Elektronik tablo hesaplama programında tablo oluşturur ve düzenlersin.
- Veri tabanı oluşturursun.
- Tablodaki verileri hesaplamak için ileri düzey formüller ve işlevler uygularsın.
- Belirlenen kriterlere göre farklı grafik türlerini seçer ve düzenlersin.
- Belirlenen kriterlere göre tablo verilerini sıralar ve filtrelersin.
- Etkileşimli tablo oluştururken koşullu biçimlendirmeyi kullanırsın.
- Mutlak ve göreceli adreslemeyi kullanırsın.
- Pivot tabloları düzenlersin.
- Tablo verilerine koruma uygularsın.



TEKRARLAMA SORULARI:

Bir müzik festivali düzenleyen ekibin bir parçası olduğunu hayal et. Bando grupları, güzel müzikler, lezzetli yiyecekler ve bolca eğlence olacak. Görevin, Excel'de bir tablo hazırlamak ve tüm hazırlıklar için ne kadar paraya ihtiyaç duyulacağını hesaplamaktır – grupların kiralanması, ses sistemi, reklamlar, gönüllüler için yiyecekler ve festivalin mükemmel şekilde organize edilmesi için gerekli olan her şey!



Veri tabanı, kolayca erişilebilen, yönetilebilen, güncellenebilen ve analiz edilebilen organize bir veri topluluğudur. Verilerin dağınık veya düzensiz listeler halinde saklanması yerine, veri tabanı bilgilerin sistematik bir şekilde depolanmasını sağlar. Böylece verilerin kullanımında verimlilik ve güvenilirlik artar. Veri tabanları, modern teknolojinin neredeyse tüm alanlarında kullanılır — küçük kişisel uygulamalardan bankacılık, sağlık, eğitim ve e-ticaret gibi karmaşık sistemlere kadar. Veri tabanlarının temel özellikleri şunlardır: düzen, yapı, veriler arasındaki bağlantı, arama imkânı, koruma ve veri bütünlüğünün korunması.

Tablolar, veri tabanlarının temel ve en sık kullanılan unsurlarından biridir. Bilgilerin depolanması için temel yapı olarak kullanılırlar. Her tablo satır ve sütunlardan oluşur. Hücre bir alanı, satır (veya kayıt) tek bir veri ögesini, sütun ise bu ögenin özelliklerinden birini temsil eder. Sütunun adı aslında alanın adıdır. Örneğin, öğrencilerle ilgili bir veri tablosunda „Ad“, „Soyad“, „Sınıf“ ve „Devamsızlık Sayısı“ adlı sütunlar bulunabilir. Her satır ise bu alanlara ait belirli değerlere sahip bir öğrenciyi temsil eder. Dikkat edilmesi gereken nokta, tabloda boş satırların ve sütunların bulunmamasıdır.

Tablolarda farklı veri türleri bulunabilir: metin (isimler ve adresler gibi), sayılar (notlar ve fiyatlar gibi), tarihler, mantıksal değerler (evet/hayır) ve hatta diğer tablolardan gelen bağlantılı bilgiler. Bu çeşitlilik, tabloların esnek olmasını ve farklı durumlarda kullanılabilmesini sağlar. Örneğin, bir çevrim içi mağaza uygulamasında bir tablo ürünlerle, başka bir tablo kullanıcılarla, üçüncü bir tablo ise siparişlerle ilgili bilgileri saklayabilir. Bu tabloların tümü ortak değerler (anahtarlar) aracılığıyla birbirine bağlanır ve böylece ilişkisel veri tabanlarında güçlü bir veri ağı oluşturulur. Bu ders kitabında daha çok tablolar ve hesaplamalarla çalışmaya odaklanacağız. Veri tabanında ilişkilerin nasıl oluşturulduğunu ise gelecek eğitim yılında öğreneceğiz.

Tablolar, bilgilerin yönetilmesi açısından büyük önem taşır, çünkü açık bir yapı sağlar, hızlı arama ve filtreleme imkânı sunar ve verilerin tutarlı kalmasını garanti eder. Tablolar olmadan veri tabanı, anlamı ve kontrolü olmayan düzensiz bir veri yığını olurdu. Bu nedenle tabloları ve bunların doğru kullanımını anlamak, veri tabanlarının nasıl çalıştığını ve gerçek dünyada nasıl kullanıldığını öğrenmek isteyen herkes için temel bir konudur.

HATIRLAYALIM:

Tablo

Tablo, verilerin düzenli bir şekilde gösterilmesini sağlayan ve satır ile sütunlardan oluşan bir yapıdır. Bilgilerin sistematik olarak gösterilmesi ve analiz edilmesi için kullanılır. Tablolar metin, sayılar, formüller veya diğer verileri içerebilir.

Satırlar

Tablodaki satırlar, soldan sağa doğru uzanan yatay çizgilerdir. Her satır bir kaydı veya birbiriyle ilişkili verilerden oluşan bir bütünü temsil eder. Örneğin bir satır, bir ürünle veya bir öğrenciyle ilgili tüm verileri içerebilir. Satırlar sayılarla (1, 2, 3, ...) gösterilir ve her yeni kayıt genellikle yeni bir satıra yazılır.

Sütunlar

Sütunlar, çalışma sayfasının üst kısmında harflerle gösterilen dikey bölümlerdir (A, B, C, ...). Her sütun aynı türden verileri içerir. Örneğin bir sütun yalnızca isimler, başka bir sütun notlar, üçüncü bir sütun ise tarihler için kullanılabilir.

Hücreler

Hücre, bir satır ile bir sütunun kesişimidir ve verilerin girildiği temel birimdir. Her hücrenin sütun ve satıra göre belirlenen kendi adresi vardır. Örnek: A1, birinci sütun ve birinci satırdaki hücredir, C3 ise üçüncü sütun ve üçüncü satırdaki hücredir.

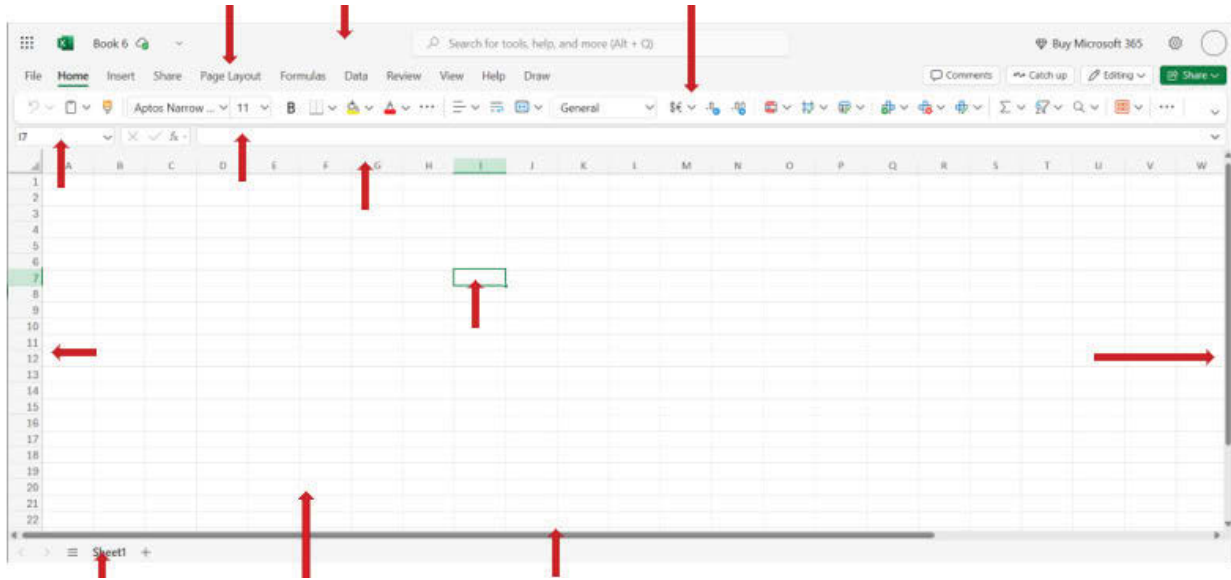
Çalışma kitabı (Workbook)

Elektronik tablo hesaplama programında hazırlanan çalışma belgesine çalışma kitabı (Workbook) denir. Çalışma kitabı, çalışma sayfalarından (Worksheet) oluşur.

Çalışma sayfası (Worksheet)

Çalışma sayfası, çalışma kitabındaki bir sayfadır. Her çalışma sayfası satır ve sütunlardan oluşan bir ızgara içerir ve basit listelerden karmaşık hesaplama ve analizlere kadar farklı amaçlar için kullanılır. Her çalışma kitabı bir veya daha fazla çalışma sayfası içerebilir. Bu sayfalar, alt kısımda Sheet1, Sheet2... şeklinde sekmelerle gösterilir. Bu sayfalara veriler (sayılar, metinler, formüller) girilir, işlenir ve kaydedilir. Çalışma sayfalarını ekleyebilir ve silebilirsiniz. Çalışma sayfalarındaki veriler birbirinden tamamen bağımsız olabilir, ancak birbirleriyle bağlantılı da olabilir.

Elektronik tablo hesaplama programındaki bir çalışma penceresinin unsurları nelerdir? Resim 1'de işaretlenmiş bölümlerin adlarını yazınız!



Resim 1 Çalışma penceresi

Terimler Başlık çubuğu, Araç çubuğu, Çalışma sayfası, Dikey kaydırma çubuğu, Etkin hücre, Sütun adı, Satır adı, Çalışma tablosu, Yatay kaydırma çubuğu, Formül çubuğu, Menü çubuğu (komutlar), Etkin hücrenin / aralığın / adlandırılmış nesnenin adresi.

5.1.1 Çalışma Sayfalarıyla İşlemler

Daha büyük projeler üzerinde veya çok miktarda veri içeren görevlerde çalışırken, tüm verileri aynı çalışma sayfasına girmek karmaşaya neden olabilir ve anlaşılabilirliği zorlaştırabilir. Bunun yerine verileri ayrı çalışma sayfalarında düzenlemek daha uygundur. Böylece her sayfanın kendine ait açık bir amacı veya içeriği olur, ancak tüm sayfalar aynı belgeye ait olmaya devam eder.

Bu şekilde bir organizasyonla çalışma daha basit ve verimli hâle gelir – veriler üzerinde daha iyi bir genel bakış elde edilir, bilgiler daha kolay yönetilir ve bütünlük duygusunu kaybetmeden yalnızca üzerinde çalıştığınız bölüme odaklanabilirsiniz. Aynı zamanda hata yapma riski azalır ve daha doğru analiz yapılmasına olanak sağlanır.

Birden fazla çalışma sayfasıyla çalışmak, sayfa ekleme, taşıma, yeniden adlandırma, kopyalama veya silme gibi farklı işlemlerin yapılmasını gerektirir. Bu işlemlere çalışma sayfalarıyla yapılan işlemler denir.



HATIRLAYALIM!

Yeni çalışma sayfası ekleme

- Son çalışma sayfasının yanındaki artı (+) simgesine tıkla (sol altta). Yeni sayfa etkin sayfanın hemen yanında görünecek ve "Sheet2", "Sheet3" vb. adını taşıyacaktır.

Çalışma sayfasını silme

- Silmek istediğin sayfanın adına sağ tıkla (alttaki sekmelerde). Menüden Delete seçeneğini seç. Seçimini OK ile onayla.

Çalışma sayfasını yeniden adlandırma

- Sayfanın adına çift tıkla, yeni adı yaz ve Enter tuşuna bas veya
- Sayfaya sağ tıkla > Rename > yeni adı yaz > seçimini OK ile onayla.

Çalışma sayfasını taşıma

- Taşımak istediğin sayfaya tıkla ve basılı tut, ardından sayfayı (drag and drop) sola veya sağa sürükle. İstedikçe konuma geldiğinde tıklamayı bırak.

Çalışma sayfasını kopyalama

- Çalışma sayfasına sağ tıkla > açılan pencereden Copy Sheet seç > sayfayı nereye kopyalamak istediğini seç > Paste Sheet (Excel 365 Online).
- Sayfaya sağ tıkla > Move or Copy > açılan pencereden sayfayı nereye kopyalamak istediğini seç > Create a copy kutusunu işaretle > OK düğmesine tıkla.
- Kopya, orijinal sayfayla aynı ada sahip olur, ancak sonuna bir numara eklenir (örneğin "Sheet1 (2)").

5.1.2 Tabloyu Düzenleme

Hücrelerdeki içeriğin tamamen görüntülenemeyecek kadar uzun veya geniş olması sık karşılaşılan bir durumdur. Böyle durumlarda satır ve sütunların boyutlarını ihtiyaçlarına göre ayarlayabilmek çok önemlidir.

Excel'de satır ve sütunların boyutlarını değiştirmek için birkaç seçenek vardır.

Sütunların genişliği birkaç teknikle değiştirilebilir:

- Fare işaretçisi sütunun sağ kenarına (sütun harflerinin arasına) getirilir. İmleç uygun şekli aldığı anda  sol tıklanır ve istenilen yöne sürüklenir.
- Home → Format → Column Width... seçilir. Column Width alanına istenilen değer yazılır.
- Sütunun adına sağ tıklanır ve menüden Column Width seçilir.
- Satırların yüksekliği de birkaç teknikle değiştirilebilir:
- Fare işaretçisi satırın alt kenarına getirilir. İmleç uygun şekli aldığı anda  sol tıklanır ve istenilen yöne sürüklenir.
- Home → Format → Row Height... seçilir. Row Height alanına istenilen değer yazılır.
- Satırın adına sağ tıklanır ve menüden Row Height seçilir.

Birden fazla sütunun genişliği veya birden fazla satırın yüksekliği aynı anda ayarlanabilir. Bunu yapmak için öncelikle boyutlarını değiştirmek istediğin tüm sütunları

veya satırları seçmelisin. Daha sonra, tek bir sütunun veya satırın boyutunu değiştirme işleminde uygulanan aynı yöntemi kullanabilirsin.

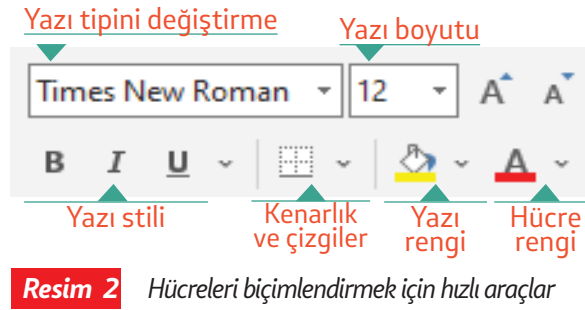


NOT:

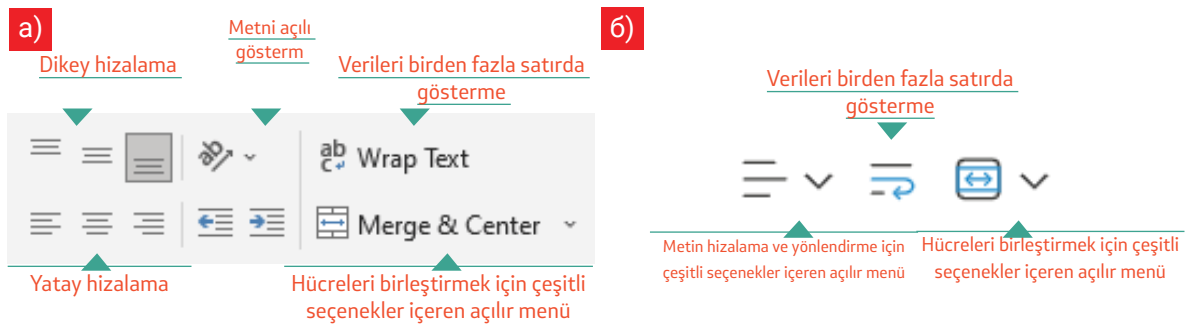
Bir hücredeki sayısal verinin yanında kare (#) işareti görünüyorsa, bu hücrede verinin görüntülenmesi için yeterli "yer" olmadığı ve sütunun genişletilmesi gerektiği anlamına gelir.

5.1.3 Hücreleri Düzenleme

Verilerin açık, düzenli ve işlevsel olması için hücreleri düzenlemek çoğu zaman gereklidir. Bu işlem, biçimin değiştirilmesini, yazı tipinin ve yazı tipi görünümünün değiştirilmesini, hücrelerdeki verilerin hizalanmasını, renk ve kenarlıkların eklenmesini, birden fazla hücrenin birleştirilmesini ve benzeri işlemleri kapsar. Hücreler doğru şekilde düzenlendiğinde tabloların okunması ve analiz edilmesi daha kolay olur



Resim 2 Hücreleri biçimlendirmek için hızlı araçlar



Resim 3 Metin ve hücreleri düzenleme araçları (a) masaüstü sürümü (b) Excel 365 çevrimiçi

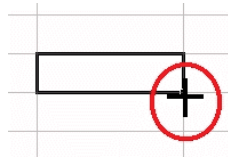
Tablonun, yani hücrelerin düzenlenmesini Format Cells penceresi üzerinden de yapabilirsin (Number: bölümünde veri türünü seçip düzenleyebilirsin, Alignment: bölümünde verileri hizalayabilirsin, Font: bölümünde yazı tipini seçip düzenleyebilirsin, Border: bölümünde ise hücre kenarlıklarının stilini, kalınlığını ve rengini seçebilirsin).

NOT: Hücrelerin hazır biçimle biçimlendirilmesi Cell Styles seçilerek yapılır (Home sekmesinde). Tabloların hazır biçimle biçimlendirilmesi ise Format As Table seçilerek yapılır (Home sekmesinde).

5.1.4 Otomatik doldurma

Otomatik doldurma, önceki hücrelerdeki örneğe dayanarak bir hücre dizisinin otomatik olarak doldurulmasını sağlayan çok kullanışlı bir işlemdir. Otomatik doldurma kullanımı, tabloların hazırlanması sırasında çalışmayı kolaylaştırır ve zaman kazandırır.

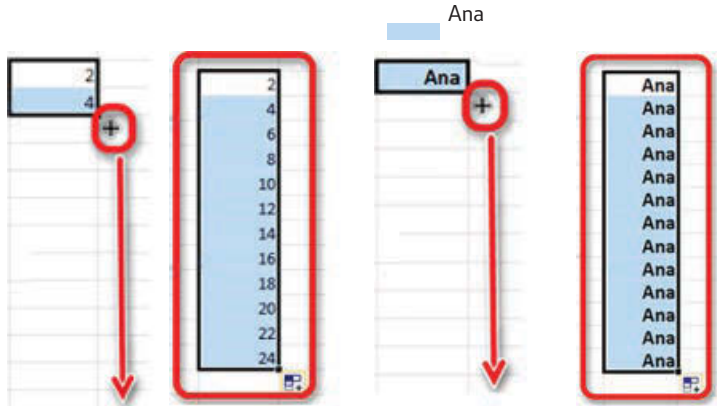
Otomatik doldurma en kolay şekilde, otomatik doldurma tutamacının herhangi bir yöne sürüklenmesiyle yapılır. Otomatik doldurma tutamacı, hücrenin kenarlığının sağ alt köşesinde bulunan küçük siyah kare şeklindeki işarettir. Fare işaretçisi üzerine getirildiğinde "+" şeklini alır. Sol fare düğmesini basılı tutup hücreleri doldurmak istediğin yöne doğru sürüklemen yeterlidir.



Resim 4 Otomatik doldurma

Aslında otomatik doldurma işlemi şu şekildedir:

- Ek hücreleri doldurmak için temel olarak kullanılacak bir hücre seçilir. 1,2,3,4,5... gibi bir seri için ilk iki hücreye 1 ve 2 yazılır. 2,4,6,8... gibi bir seri için 2 ve 4 yazılır. 2,2,2,2... gibi bir seri için yalnızca ilk hücreye 2 yazılır. Metin verisi yalnızca ilk hücreye girilir.
- Girilen sayılar/metin seçilir ve otomatik doldurma tutamacı istenilen yöne doğru sürüklenir.



Resim 5 Otomatik doldurma işlemi

- Gerekirse otomatik doldurma seçeneğine tıklanarak  istenilen seçenek seçilebilir.

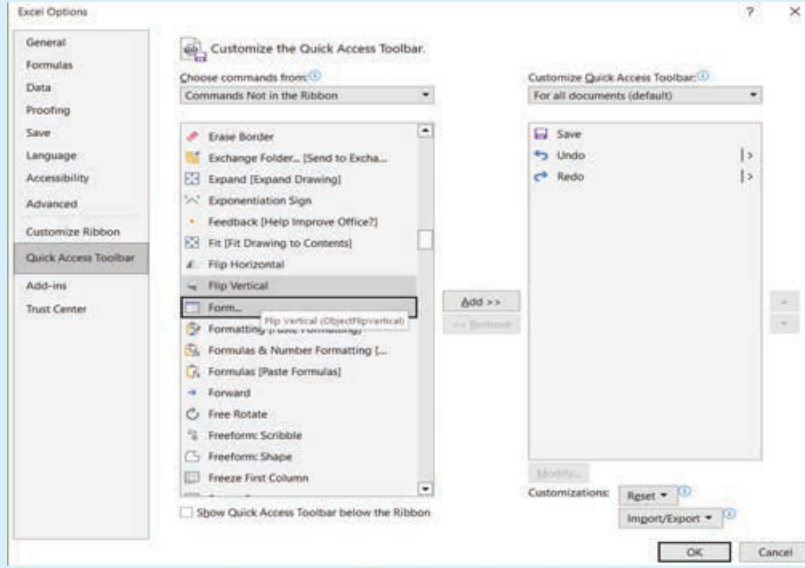


DAHA FAZLA BİLGİ EDİNMEK İSTEYENLER İÇİN

Veri tabanına veri girişi için form oluşturma

Veri tabanları genellikle çok miktarda veri içeren tablolar içerir. Bu verilerin elle girilmesi yorucu ve zaman alıcıdır, insan hatalarına açıktır ve daha önce girilmiş verilerin yanlışlıkla değiştirilme ihtimali vardır. Tabloya veri giriş işlemi, bir form oluşturularak daha kolay ve düzenli hâle getirilebilir. Ancak öncelikle ilgili aracın Hızlı Erişim Araç Çubuğu'na eklenmesi gerekir.

Form aracını araç çubuğuna ekleme



Resim 6 Excel Options penceresi

Adımlar:

1. File → Options → Quick Access Toolbar seçeneklerini aç.
2. Choose commands from penceresinden Commands Not in the Ribbon seçeneğini seç.
3. Alt panelde Form seçeneğine tıkla.
4. Add düğmesine tıkla.
5. Form, sağ panele aktarılacaktır.
6. OK ile onayla.

Şimdi Hızlı Erişim Araç Çubuğu'nda yeni bir Form aracı görünecektir.



Resim 7 Form aracı

Form oluşturma

Formu kullanarak veri girmeden önce tabloyu "akıllı" Excel tablosuna dönüştürmen gerekir. Bunu, tabloyu seçip Insert → Table seçeneklerini kullanarak yapabilirsin. Create Table penceresinde "My table has headers" seçeneğinin işaretli olup olmadığını kontrol et.

	A	B	C	D	E
1	Ürün adı	Kategori	Stok miktarı	Fiyat	Tedarik tarihi
2	Televizyon	elektronik	32	25,000.00 den	24-Ocak-2025
3	Dizüstü bilgisayar	elektronik	45	42,000.00 den	24-Ocak-2025
4	Cep telefonu	elektronik	16	28,000.00 den	20-Mart-2025
5	Tablet	elektronik	56	18,000.00 den	11-Mart-2025
6	Tiştirler	giyim	220	480.00 den	5-Mart-2025

Resim 8 Excel tablosu örneği

Adımlar:

1. Tablondaki herhangi bir hücreye tıkla.
2. Üst şeritte yeni eklenen "Form" düğmesine tıkla. Belirlenen alanlara veri girebileceğin bir pencere açılır.
3. Verileri gir ve yeni bir kayıt eklemek için "New" düğmesine tıkla. Veriler arasında gezinmek için "Find Prev" ve "Find Next" seçeneklerini kullan. Bir veriyi silmek istiyorsan "Delete" düğmesine bas .

Sheet10

Ürün adı: Televizyon

Kategori: Elektronik

Stok miktarı: 32

Fiyat: 25000

Tedarik tarihi: 24 Oca 2025

1 of 13

New

Delete

Restore

Find Prev

Find Next

Criteria

Close

Resim 9 Örnek tabloya veri girişi formu



BİLGİ KONTROLÜ SORULARI



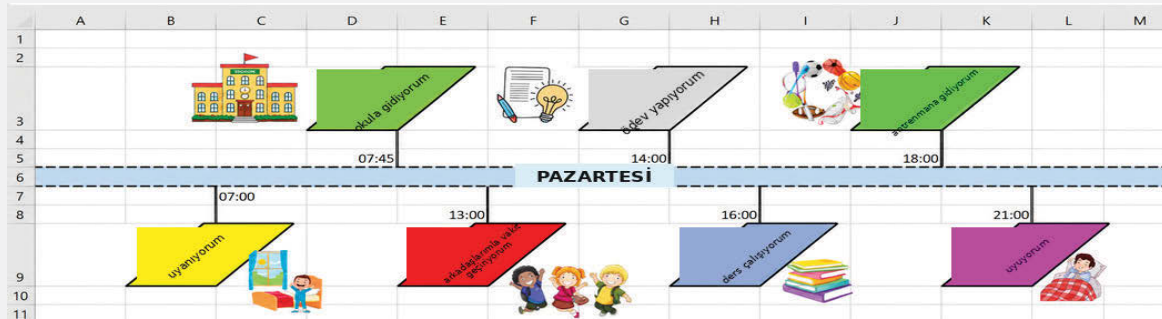
1. Excel'de çalışma sayfası nedir ve nerede bulunabilir?
2. Elektronik tablo programında Sanjah hücrelerden birine veri giriyor. AŞAĞIDAKİlerden hangisi Sanjah'nın veri girdiği hücrenin adresi olabilir?
a. AB1 b. 1ZA c. 1A1B d. A1B1
3. Bildiğin çalışma sayfası işlemleri nelerdir ve bunların nasıl gerçekleştirildiğini açıkla?

4. Excel belgesinde birden fazla çalışma sayfası kullanmak neden yararlıdır?
5. Excel’de “satır” ne anlama gelir? “Sütun” nedir?
6. Birden fazla sütunun genişliğini aynı anda değiştirme işlemini açıkla!
7. Bir hücreye hangi veri türleri girilebilir?
8. Elektronik tablo verileriyle çalışırken hücreleri doğru şekilde düzenlemek neden önemlidir?
9. Birden fazla hücreyi tek bir hücrede birleştirme adımları nelerdir?
10. Otomatik doldurma nedir ve veri girerken işini nasıl kolaylaştırabilir?



UYGULAMALI ETKİNLİK:

1. Bir metin işleme programında “En Sevdiğim 5 Film / Kitap / Şarkı” başlıklı bir tablo oluştur (sana yakın olan bir kategoriye seç). Excel’de 5 satırdan oluşan ve senin en iyi 5 seçimini temsil eden bir tablo oluştur. Tablo 5 sütundan oluşmalıdır: | Sıra | Başlık | Tür | Puan (1–10) | Yorum | |
- a) Tablo başlığı (örneğin: “En Sevdiğim 5 Film”) şu özelliklerde olmalıdır: Arial Black, 16 punto, tablonun üzerinde ortalanmış (birleştirilmiş hücreler), mavi renkte.
- b) Sütun başlıkları (Sıra, Başlık, Tür, vb.): kalın, ortalanmış, Calibri veya Verdana yazı tipi, 12 punto, arka plan: açık gri.
- c) Yorum sütunundaki metin italik ve 11 punto olmalıdır.
- d) Tabloya kenarlık ekle.
2. Kendi seçtiğin bir konu için kronolojik bir zaman çizelgesi oluştur.



Resim 10 Günlük hayatımdan bir gün” kronolojik zaman çizelgesi örneği

5.2

Formüllerin ve İşlevlerin İleri Düzeyde Kullanımı

Notlardan oluşan bir tablon olsaydı, en yüksek ortalamaya sahip öğrenciyi en kolay nasıl bulabilirdin? Aylara göre giderlerin bulunduğu bir tablon varsa, yalnızca ocak ayında harcanan toplam tutarı nasıl bulabilirdin? Çok fazla veriye sahip olduğunda, bunları tablolarda kolayca düzenlemene, hızlı hesaplamalar yapmana ve sonuçları anlaşılır bir şekilde göstermene hangi program olanak sağlar? Tabloda “elle” hesaplama yapmadan otomatik hesaplamalar elde etmek için ne kullanırsın?

5.2.1 Elektronik Tablo Programındaki Kavramlar



HATIRLAYALIM:

- **Formül** bir hücreye girilen ve programın tablodaki veriler üzerinde hesaplama yapmasını sağlayan bir ifadedir. Formülle programa hangi değerleri alacağını, bunları nasıl ilişkilendireceğini (örneğin toplayacağını veya çıkaracağını) ve sonucu nereye göstereceğini söyleriz. Formüller, sayılar, hücre adresleri, matematiksel işaretler ve sonuç elde etmek için kullanılan işlevleri içerebilir.
- **Formül örnekleri:**
 - $= (5+6)*9$
 - $= (B1+B2)-B12$
 - $= 9+10+B6$
 - $= ((B10)*10)/(B2+4)$
 - İşlevler, programa önceden tanımlanmış ve yerleşik olan formüller olarak açıklanabilir.

Bir işlevin öğeleri, işlevin adı ve argümanlarıdır.

$= \text{İşlevAdı} (\text{argumenti})$

İşlevAdı, yapılması gereken işlemi açıklayan kelimedir. Örnekler: SUM – toplama işlemi için, AVERAGE – ortalama hesaplamak için, MIN – minimum, en küçük değeri hesaplamak için, MAX – maksimum, en büyük değeri hesaplamak için kullanılır.

Argümanlar, kullandığımız işlevin türüne ve ulaşmak istediğimiz sonuca bağlı olarak farklı değerler olabilir. Bunlar metin, hücre adresleri, hücre aralıkları, metin, mantıksal değerler veya diğer işlevler olabilir. Örnekler:

- SUM(A1:A10) argümanı, toplama yapılacak sayıları içeren A1:A10 hücre aralığıdır.
- AVERAGE(10, 20, 30) argümanları, ortalaması hesaplanacak sabit değerlerdir.
- MAX(B2, B3, B4) argümanları, B2, B3 ve B4 olmak üzere üç ayrı hücre adresidir ve bunların içindeki en büyük sayısal değer gösterilir.
- MAX(SUM(A1:A3), 100) argümanları, SUM(A1:A3) işlevi ve 100 sabitidir.

...

Operatörler, formüllerin ve işlevlerin ayrılmaz bir parçasıdır ve formül veya işlev içerisinde belirli matematiksel, mantıksal veya metinsel işlemlerin yapılmasını sağlar. Farklı operatör türleri vardır:

- Aritmetik operatörler

+ (toplama), - (çıkarma), * (çarpma), / (bölme), ^ (üs alma).

Örnek: A1 hücresinde 8, B1 hücresinde 4 sayısı varsa, =A1+B1 formülü 12 sonucunu verir – çünkü + operatörü iki sayıyı toplar.

- Karşılaştırma operatörleri > (büyük), < (küçük), = (eşit), >= (büyük veya eşit), <= (küçük veya eşit), <> (eşit değil).

Örnek: A1 hücresinde 10, B1 hücresinde 15 sayısı varsa, =A1<B1 formülü TRUE sonucunu verir – çünkü 10, 15'ten küçüktür.

- Metin operatörü & (metinleri birleştirme).

Örnek: A1 hücresinde "Dijital", B1 hücresinde "dünya" kelimesi bulunuyorsa, =A1 & " " & B1 formülü "Dijital dünya" sonucunu gösterir (kelimeler arasına boşluk eklenmesini sağlar).

- Başvuru operatörleri: (aralık için kullanılır, yan yana bulunan bir grup hücreyi seçmek istediğimizde kullanılır).

Örnek: A1 ile A6 arasındaki hücrelerde sayılar varsa ve hepsini toplamak istiyorsan =SUM(A1:A6) yazarsın. (ayırma için kullanılır, birbirine bitişik olmayan birkaç hücre veya hücre aralığı seçmek istediğimizde kullanılır).

Örnek: =SUM(A1, C1:C3) işlevi, A1 hücresindeki değeri ve C1 ile C3 arasındaki tüm hücrelerin değerlerini toplar. Boşluk (kesişim için kullanılır)

Örnek: =SUM(A1:A5 A3:C4) sonucunda A3 ve A4 hücrelerindeki değerlerin toplamı elde edilir, çünkü bunlar iki aralıkta ortak bulunan hücrelerdir.

...

Hücre aralığı kavramından birkaç kez söz ettik. Peki, bu kavram aslında ne anlama gelir?

Hücre aralığı, verilerle çalışmak için seçilen bir hücre grubudur.

Hücreler:

- Aynı satırda bulunabilir. Örneğin B2:D2 (bu aralıkta B2, C2 ve D2 hücreleri bulunur).
- Aynı sütunda bulunabilir. Örneğin A1:A3 (bu aralıkta A1, A2 ve A3 hücreleri bulunur).
- Dikdörtgen bir alan oluşturabilir. Örneğin A1:C3 (bu alanda A1, A2, A3, B1, B2, B3, C1, C2 ve C3 hücreleri bulunur).

...

Hücre adresi, çalışma sayfasındaki bir hücrenin konumunu belirten benzersiz tanımlayıcıdır. Sütunun harfinden (A, B, C...) ve satırın numarasından (1, 2, 3...) oluşur

Örnek:

- A1 → A sütunu, 1. satır
- D5 → D sütunu, 5. satır

Formül kopyalandığında veya taşındığında adresin nasıl değiştiğine göre birkaç hücre adresi türü vardır:

- Göreceli adres, tablodaki bir hücrenin veya hücre aralığının konumunu gösterir ancak sabit değildir. Göreceli adres içeren bir formül başka bir hücreye kopyalandığında veya taşındığında, adres yeni konuma göre otomatik olarak değişir. Sütun harfinin veya satır numarasının önünde herhangi bir işaret kullanılmaz.

Örneğin:

D2 hücresinde =B2*C2 formülü bulunuyor. Formülü D2 hücresinden D3 hücresine kopyalarsan, formül otomatik olarak =B3*C3 şeklinde değişir. Bu, göreceli adreslerin formülü nereye yerleştirdiğine bağlı olarak "komşu" hücrelerin içeriğini hesaplamak üzere otomatik olarak uyarlanması anlamına gelir.

D2				
	A	B	C	D
1	Ürün	Miktar	Fiyat	Toplam
2	Sırt çantası	3	2,700.00 den.	8,100.00 den.
3	Çadır	1	9,300.00 den.	
4	Pusula	1	4,000.00 den.	
5	Termos	2	1,200.00 den.	
6	El feneri	5	950.00 den.	

Resim 11 Göreceli hücre adresinin kullanımına örnek

- Mutlak adres, bir hücrenin sabit adresidir ve formül başka bir hücreye kopyalandığında veya taşındığında değişmez. Her zaman belirli bir hücreyi gösterir. Mutlak adres, sütun harfinin ve satır numarasının önüne \$ işareti eklenerek gösterilir. Örneğin \$A\$1, her zaman A sütunundaki 1. satırda bulunan hücreyi gösteren mutlak adrestir.

ÖRNEK: B2 hücresinde =A2+\$B\$7 formülü bulunmaktadır. Formülü B2 hücresinden B3 hücresine kopyalarsan, formül otomatik olarak =A3+\$B\$7 şeklinde değişir. Yani \$B\$7 adresi sabit kalır ve her zaman B7 hücresindeki değeri kullanır. Buna karşılık A2 adresi, formülün kopyalandığı yere göre A3, A4 vb. şeklinde değişir.

Formül (işlev) kopyalandığında veya taşındığında hücre adresinin nasıl değiştiğine göre birkaç farklı adres türü vardır:

	A	B	C
1	Ürün fiyatı	Toplam fiyat	
2	500	670	
3	1200	1370	
4	630	800	
5	850	1020	
6			
7	Teslimat	170	

Resim 12 Mutlak adresin kullanımına örnek

- Karışık adres, mutlak ve göreceli adresin birleşimidir. Bu, adresin bir bölümünün sabit, diğer bölümünün ise formül kopyalandığında değiştiği anlamına gelir.

ÖRNEK: \$A1 → A sütunu sabittir, 1. satır değişir, A\$1 → 1. satır sabittir, A sütunu değişir.

5.2.2 Karmaşık Fonksiyonlar

Şimdiye kadar tablolarda basit hesaplamalar yapmak için SUM, AVERAGE, MIN, MAX gibi temel işlevleri kullanmayı öğrendin. Peki, tablodan daha ayrıntılı bilgiler elde etmen gerektiğinde ne yaparsın? Örneğin:

- Kaç öğrencinin notu 5?
- Yalnızca yiyecek için toplam ne kadar para harcanmış?
- Bir öğrencinin aldığı nota göre başarılı olup olmadığını nasıl kontrol ederiz?

Bu tür soruların cevaplarını COUNT, COUNTIF, SUMIF ve IF gibi gelişmiş işlevler verir.

Bunlarla şunları yapabiliriz:

- Bir şeyin kaç kez tekrarlandığını sayabiliriz (COUNTIF).
- Bir koşula göre değerleri toplayabiliriz (SUMIF).
- Bir formülde seçim veya karar verebiliriz (IF).
- Toplamda kaç tane sayısal değer olduğunu sayabiliriz (COUNT).



HATIRLAYALIM:

Fonksiyonu hücreye doğrudan yazarak girebilirsin.

Adımlar:

1. Fonksiyonu girmek istediğin hücreye tıkla (hücreyi seç).
2. Bir fonksiyon girdiğini belirtmek için = işaretini yaz.
3. Fonksiyonu gir. Örneğin: =SUM(A1:A10).
4. Uygulamak için Enter tuşuna bas.
5. Fonksiyonu Formulas menüsü üzerinden de girebilirsin.

Adımlar:

1. Fonksiyonu girmek istediğin hücreye tıkla.
2. Menü çubuğunda Formulas sekmesine tıkla.
3. Bir fonksiyon kategorisi seç. (Kategoriler: finansal, mantıksal, metinsel, tarih ve saat, başvuru, matematiksel ve trigonometrik fonksiyonlar.)
4. Belirli bir fonksiyon seç.
5. Fonksiyonun kullandığı argümanları gir.
6. Uygulamak için Enter tuşuna bas.

...

COUNT Fonksiyonu

COUNT fonksiyonu, seçilen aralıktaki sayısal değerler içeren hücrelerin sayısını hesaplar. Yani yalnızca sayı içeren hücreleri sayar, boş hücreleri, metinleri ve uygun olmayan diğer değerleri dikkate almaz.

Fonksiyonun Formatı:

=COUNT (aralık1, aralık2, ...) şeklindedir.

Burada argüman, yalnızca sayısal değerlerin sayılacağı bir veya daha fazla hücre aralığı olabilir.

Örnek: Bir mağazada farklı ürünlerin günlük satışlarının bulunduğu bir tablo tutulmaktadır. A sütununa ürünlerin adları, B sütununa ise miktarları girilmiştir. Bazı alanlar boş olabilir veya ürün o gün satılmadıysa metin içerebilir.

O gün kaç farklı ürün satılmıştır?

B9	X	✓	f _x	=COUNT(B2:B7)
	A	B	C	D
1	Ürün	Miktar		
2	Sırt çantası	5		
3	Çadır	2		
4	Kamp lambası	tekst		
5	Uyku tulumu	7		
6	Pusulula			
7	Mendiller	12		
8				
9	Cevap	4		

Resim 13

Farklı ürünlerin günlük satışları için ÖRNEK Tablo

Cevap: =COUNT(B2:B7). 4 farklı ürün satılmıştır.

Not: Yalnızca sayısal değer içeren hücreleri sayan COUNT fonksiyonundan farklı olarak, COUNTA fonksiyonu metin, tarih, mantıksal değerler ve diğer içerikleri de dikkate alır.

COUNTIF Fonksiyonu

COUNTIF fonksiyonu, belirli bir hücre aralığında belirli bir koşulun kaç kez gerçekleştiğini sayar. Yani kaç hücrede belirli bir sayı bulunduğunu, kaç hücrede belirli bir metin olduğunu veya kaç hücrenin belirli bir değerden büyük ya da küçük olduğunu sayar.

Fonksiyonun formatı:

=COUNTIF(aralık, koşul)

Burada aralık, tabloda verilerin arandığı bölümdür, koşul ise hücrelerin karşılaması gereken kriterdir.

Örnek:

- =COUNTIF (A1:A10, ">50") – Bu formül, A1:A10 aralığında değeri 50'den büyük olan hücreleri sayar.

- =COUNTIF(B1:B20,"Tamamlandı") – B1:B20 aralığında "Tamamlandı" sözcüğünün kaç kez geçtiğini sayar.
- =COUNTIF(C1:C15,">=10") – C1:C15 aralığında 10'a eşit veya 10'dan büyük değere sahip hücreleri sayar.
- =COUNTIF(D1:D30,"") – D1:D30 aralığındaki boş hücreleri sayar.

ÖRNEK: Meyve sütununda "Elma" sözcüğü kaç kez geçmektedir?

	A
1	Meyve
2	Elma
3	Muz
4	Elma
5	Limon
6	

Resim 14 Meyve" sütunu

Cevap: = COUNTIF(A2:A5, "Elma"). Şekil 14'te verilen örnek için cevap 2 kezdir.

...

SUMIF Fonksiyonu

SUMIF fonksiyonu, belirli bir aralıkta bulunan ve verilen koşulu sağlayan hücrelerin değerlerini toplar (sunar). Bu, fonksiyonun tüm hücreleri değil, yalnızca belirlediğin kriteri uyan hücreleri topladığı anlamına gelir.

Fonksiyonun biçimi:

=SUMIF(aralık, koşul, toplam_aralığı)

Burada aralık, koşulun kontrol edildiği hücre aralığıdır, koşul, aralıktaki her hücrenin karşılaması gereken kriterdir, toplam_aralığı (isteğe bağlı) ise değerlerin toplanacağı aralıktır. Belirtilmezse, ilk aralıktaki hücreler toplanır.

Örnek:

- =SUMIF(A1:A10,">50") – A1:A10 aralığındaki yalnızca 50'den büyük sayıları toplar.
- = SUMIF(A1:A10, 50) → 50'ye eşit olan sayıları toplar.
- = SUMIF(B1:B10, "Satıldı") → Değer "Satıldı" ise toplar

Örnek: Verilen tablodan (Şekil 15) satılan sırt çantalarının sayısını toplayınız.

	A	B
1	Ürün	Satılan miktar
2	Sırt çantası	5
3	Çadır	2
4	Sırt çantası	3
5	Pusula	4
6		
7	Satılan sırt çantaları	

Resim 15 Belirli bir ürünün satışlarını toplamak için tablo örneği

IF Fonksiyonu

IF fonksiyonu, bir koşulun mantıksal olarak kontrol edilmesini sağlar ve koşulun sağlanıp sağlanmadığına (TRUE veya FALSE) bağlı olarak farklı değerler döndürür. Bu, fonksiyonun bir durumun doğru olup olmadığını kontrol ettiği, ardından doğruysa bir değer, yanlışsa başka bir değer döndürdüğü anlamına gelir.

Fonksiyonun biçimi:

=IF(mantıksal_koşul, doğruysa_değer, yanlışsa_değer)

Burada mantıksal_koşul, kontrol ettiğimiz ifadedir, doğruysa_değer – koşul sağlandığında elde edilen sonuç, yanlışsa_değer – koşul sağlanmadığında elde edilen sonuçtur.

Örnek:

- =IF(A1>50, "büyük sayı", "küçük sayı") – A1 hücresindeki değer 50'den büyükse sonuç olarak „büyük sayı“ görüntülenir, aksi durumda „küçük sayı“ görüntülenir.
- =IF(A1>0, "Pozitif", "Negatif") – hücredeki değer sıfırdan büyükse sonuç olarak „pozitif“ metni görüntülenir, aksi durumda „negatif“ görüntülenir.

Örnek: Resim 16'da ürünleri ve depodaki mevcut miktarlarını gösteren bir tablo verilmiştir. „Mesaj“ sütununda, 1 veya daha fazla adet varsa „Stokta“, 0 adet varsa „Sipariş ver“ yazmalıdır. Hangi fonksiyonu kullanırsınız?

	A	B	C
1	Ürün	Miktar	Mesaj
2	Çadır	5	
3	Pilli lamba	0	
4	Sırt çantası	2	
5	Uyku tulumu	4	
6	Pusula	12	

Resim 16 Ürünler ve depodaki mevcut miktarlarını gösteren tablo.

	A	B	C	D	E	F
1	Ürün	Miktar	Mesaj			
2	Çadır	5	Stokta			
3	Pilli lamba	0	Sipariş ver			
4	Sırt çantası	2	Stokta			
5	Uyku tulumu	4	Stokta			
6	Pusula	12	Stokta			

Resim 17 Görevin çözümü

Cevap =IF(B2>0, « Stokta»,
«Sipariş ver»)



SONUÇ

Bir elektronik tablo programında bir işlev kullandığında, önce yapmak istediğin hesaplamanın adını (örneğin: SUM, IF) belirtirsin, ardından parantez içinde argümanları, yani tam olarak neyin hesaplanması gerektiğini girersin. Argümanlar, işlevin farklı veri türleriyle çalışabilmesine esneklik kazandırır. Formüllerde ve işlevlerde operatörler önemli bir rol oynar 236



– bunlar hesaplamalar (örneğin: +, -, *, /), karşılaştırmalar (örneğin: =, >, <) ve metin işlemleri (örneğin: birleştirme için &) için kullanılır.

Bir fonksiyon kullandığında, önce hangi hesaplamayı yapmak istediğini fonksiyonun adıyla, ardından neyin hesaplanacağını parantez içindeki argümanlarla belirtirsin. Argümanlar, fonksiyonların farklı verilerle esnek bir şekilde çalışmasını sağlar.

Operatörler, hesaplamalarda, karşılaştırmalarda ve veriler üzerinde işlem yapmada önemli bir rol oynar.

Hücre aralığı, birbiriyle bağlantılı hücrelerden oluşan bir gruptur. Formüllerde ve fonksiyonlarda aynı anda birden fazla hücredeki verileri analiz etmek veya hesaplamak için kullanılır.

Formüllerle çalışırken göreceli adres ile mutlak adres arasındaki farkı bilmek önemlidir. Örneğin B2 göreceli bir adrestir ve formül kopyalandığında değişir. \$B\$2 ise mutlak bir adrestir ve formül başka bir yere kopyalansa bile aynı kalır. Bu, hesaplamalar üzerinde daha fazla kontrol sahibi olmanı sağlar.

COUNT, COUNTIF, SUMIF ve IF gibi gelişmiş fonksiyonlar, belirli koşullara göre sayma, toplama ve mantıksal kontrol yapmayı sağlar. Bu nedenle veri analizinde özellikle kullanışlıdır.

BİLGİ KONTROLÜ SORULARI VE ETKİNLİKLER



1. Hesap tabloları programında formül ile fonksiyon arasındaki fark nedir?
2. Hesap tabloları programında fonksiyonların sözdizimi (formatı) nasıl görünür?
3. Fonksiyonda argüman nedir ve bir fonksiyon ne tür argümanlara sahip olabilir?
4. =AVERAGE(A1:A10) formülü verilmiştir. Bu formülde A1:A10 neyi ifade eder?
5. Operatör nedir ve hangi türleri vardır?
6. Sürekli ve ayrılmış aralığı tanımlamak için hangi semboller kullanılır?
7. A1'den D10'a kadar olan hücre aralığı nasıl gösterilir?
8. Mutlak ve göreceli adres kavramlarını açıklayınız! Mutlak adresleri ne zaman, göreceli adresleri ne zaman kullanmalısınız?
9. =A1*2 formülünü (B1 hücresinden) B2 hücresine kopyalarsanız ne olur?
10. Karışık adres (\$A1 veya A\$1), mutlak ve göreceli adresten nasıl farklıdır?
11. COUNT, COUNTIF, SUMIF ve IF fonksiyonlarının kullanımına örnekler veriniz!



UYGULAMALI ALIŞTIRMALAR

1. „Ürünler“ adlı bir tablo oluşturun. Ürünler sütunu ve Temel Fiyat sütunu olsun! Her ürüne, ayrı bir hücrede bulunan 20 denarlık sabit bir teslimat ücreti eklenir.

	A	B	C	D
1	Ürün	Temel fiyat	Teslimat ek ücreti	Son fiyat
2	Elma	50	20	
3	Muz	40		
4	Portakal	60		
5				

Resim 18 „Ürünler“ tablosu

2. D sütununda, ek ücreti ekleyerek son fiyatı hesapla! Temel fiyat için göreceli adres kullan (kopyalama sırasında değişmesi için) ve C1'deki ek ücret için mutlak adres kullan (kopyalama sırasında değişmemesi için)!
3. Bir tabloya farklı değerler (sayılar, metinler, boş hücreler) gir! Bir fonksiyon yardımıyla kaç tane sayısal değer olduğunu belirle.
4. Bir tabloda öğrencilerin isimleri ve bugünkü Bilişim Teknolojileri dersi için durumları („Mevcut“ veya „Devamsız“) girilmiştir. Derste kaç öğrencinin mevcut olduğunu say!
5. Verilen tabloyu oluşturun ve Marko'nun kaç sipariş verdiğini ve siparişlere ne kadar para harcadığını hesaplayın!

	A	B
1	Müşteri	Toplam sipariş
2	Marko	1000
3	Ana	850
4	Marko	750
5	Ivan	600

Resim 19 „Siparişler“ tablo

6. Bir tablo oluşturun ve „Öğrencinin adı“ (en az 10 öğrenci) ile „Bilişim Teknolojileri dersi notu“ sütunlarını ekleyin! Bir işlev yardımıyla Bilişim Teknolojileri dersinden 5 alan öğrenci sayısını bul!
7. Bir hafta boyunca her derse günlük kaç saat çalıştığını kaydedeceğin bir tablo oluşturun (A sütunu – dersler, B sütunu – çalıştığın saat sayısı)! Matematik dersine ayrılan toplam saat sayısını hesapla!
8. Bir tabloda çalışanların tamamladığı görev sayıları verilmiştir. Çalışan 10 veya daha fazla görev tamamlamışsa „Evet“, aksi durumda „Hayır“ görüntülenmelidir.
9. Bir tabloya sıcaklık değerlerini gir. 0 °C'nin altındaysa „Sıcaklık“ sütununun yanına «Soğuk», 30 °C'nin üzerindeyse «Sıcak», aksi durumda «İlman» yazmalıdır.

5.3

Grafiklerle İleri Düzey Çalışma

Elektronik tablolar programlarındaki çalışma sayfaları, çoğu zaman anlaşılması veya açıklanması zor olan çok sayıda veri içerebilir. Verileri bir grafik yardımıyla görsel olarak gösterdiğinde işler çok daha anlaşılır hâle gelebilir. Yapman gereken tek şey verileri seçmek ve bir grafik türü seçmektir. Farklı grafik türleri vardır (sütun, çizgi, pasta...), bu nedenle seçimin, verileri en etkili şekilde gösterecek grafik türü olmalıdır. Eğer grafik anlaşılır değilse, çok kullanışlı olmayacaktır.



TEKRARLAMA SORULARI:

Hafta boyunca parktaki ziyaretçi sayısı

Verileri ziyaretçi sayısına göre girin:

Gün	Ziyaretçi sayısı
Pazartesi	120
Salı	150
Çarşamba	200
Perşembe	180
Cuma	220

Sütun grafik oluşturun ve en fazla ziyaretçinin ne zaman olduğunu analiz edin.

Tekrarlama çalışması için olası çözümlerden biri, Şekil 20’de gösterilen grafikdir.



Resim 20 "Hafta boyunca parktaki ziyaretçi sayısı" görevi için grafik

5.3.1 Grafiğin Öğeleri

Grafiği anlayabilmek için başlık, eksenler, gösterge ve görsel olarak sunulan veriler gibi temel unsurlarını tanımak önemlidir.

1. Grafik genellikle iki eksene sahiptir:

- **X- eksen (yatay)** Kategorilerin, tarihlerin, günlerin vb. gösterildiği eksendir.
- **Y- eksen (dikey)** Değerlerin (sayılar, yüzdelere, ölçümler vb.) gösterildiği eksendir.

1. Grafik Başlığı (Chart Title) Grafikteki verilerin açık bir tanımını vermelidir.

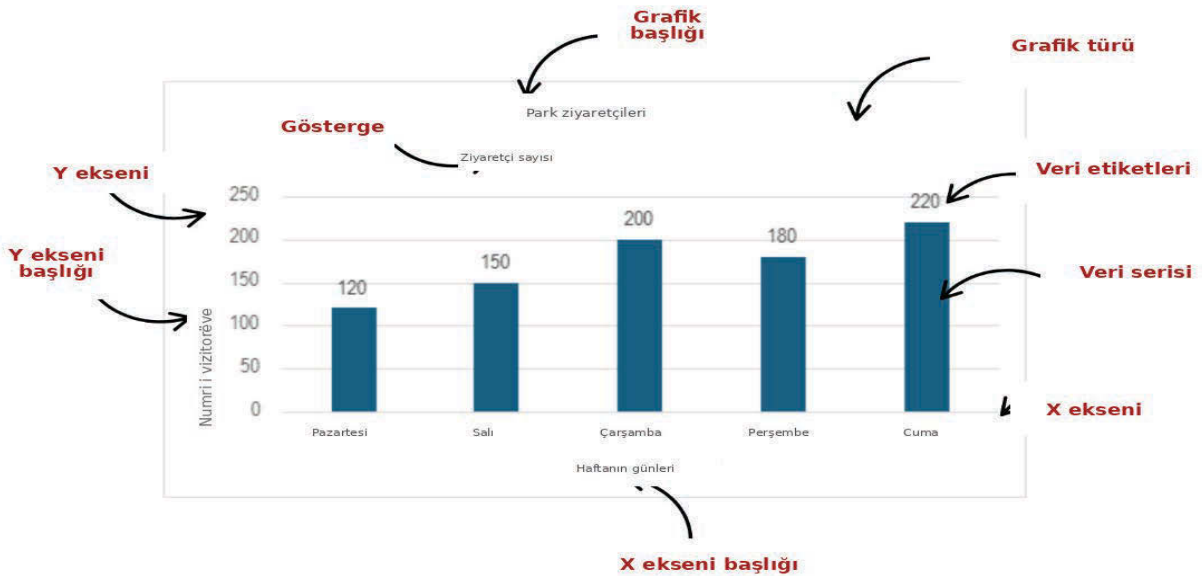
2. Eksen Başlıkları Axis Labels Grafikteki verilerin açık bir tanımını vermelidir.

3. Gösterge (Legend) Eksenlerdeki kategorileri veya değerleri belirten metindir.

4. Veri Serileri (Data Series) Sütunlar, çizgiler, noktalar ve benzeri şekillerde gösterilen bir-biriyle ilişkili veri gruplarıdır.

5. Kılavuz Çizgileri (Gridlines) Yatay veya dikey olabilen yardımcı çizgilerdir ve değerlerin okunmasını kolaylaştırır.

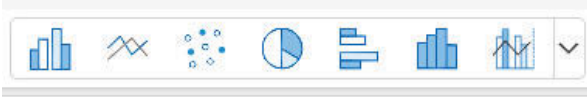
6. Veri Etiketleri (Data Labels), Sütunların, çizgilerin veya diğer grafik öğelerinin üzerinde ya da içinde bulunan ve kesin değeri gösteren sayılardır.



Resim 21 Grafiğin öğeleri

5.3.2 Grafiğin Ek Olarak Düzenlenmesi

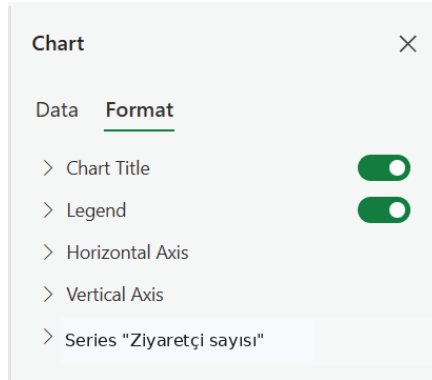
Grafiği oluşturduktan sonra fikrini değiştirebilir ve başka bir grafik türünün verileri daha iyi gösterdiği sonucuna varabilirsiniz. Böyle bir durumda, verileri yeniden girmeden grafiğin stilini veya türünü kolayca değiştirebilirsiniz. Bunun için grafiğe tıklaman, menüden Chart seçeneğini seçmen ve ardından sunulan grafik türleri listesinden birini seçmen yeterlidir.



Resim 22 Grafik türünü seçme araçları

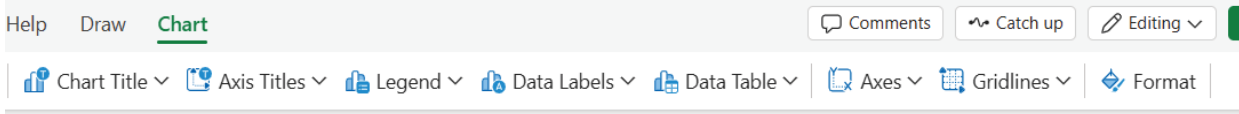
Hazır bir stil kullanmak yerine, grafiğin her bir bölümünü (sütun, çizgi, eksen, kılavuz çizgisi, gösterge vb.) ayrı ayrı düzenleyebilirsiniz. Excel 365'te bunu iki şekilde yapabilirsiniz:

- Öğeye çift tıklamak – biçimlendirme paneli açılır, burada stili, rengi, boyutu ve diğer parametreleri değiştirebilirsiniz.



Resim 23 Grafik öğelerini düzenleme araçları

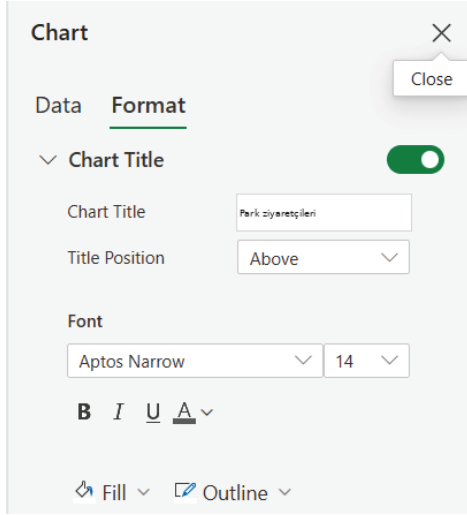
2. „Chart“ Menüsünün Kullanılması



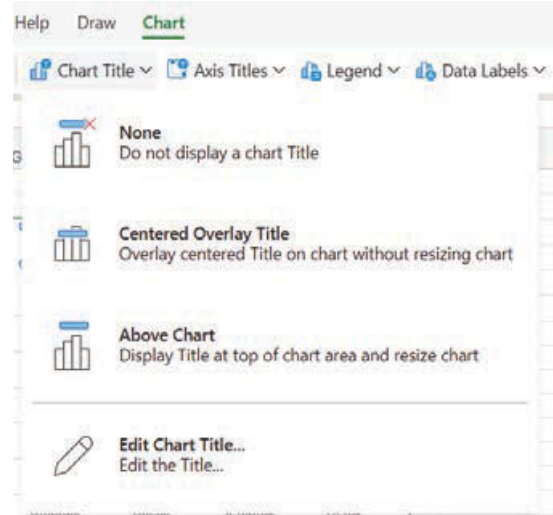
Resim 24 Grafiği düzenleme araçları – “Chart” menüsü

Grafik Başlığının Düzenlenmesi

3. Başlığa iki kez tıklayın ve yeni metni girin. Format > Chart Title bölümünden yazı tipini, boyutunu, rengini ve hizalamasını değiştirebilirsiniz (Resim 25).
4. Chart > Chart Title seçeneğini kullanırsanız, başlığı farklı konumlara yerleştirebilirsiniz (Resim 26).



Resim 25 Başlığın düzenlenmesi



Resim 26 Başlığın konumlandırılması

Eksenlerin Düzenlenmesi

- X eksenine veya Y eksenine iki kez tıklayarak eksenleri biçimlendirme seçeneğini açabilirsiniz.
- Değer aralığını, eksen adını, ölçeği vb. değiştirebilirsiniz.
- Chart > Axes menüsünden eksenleri etkinleştirebilir veya devre dışı bırakabilirsiniz.

Göstergeğin Düzenlenmesi

- Chart > Format > Legend üzerinden göstergeyi açabilir/kapatılabilir, taşıyabilir, biçimlendirebilir veya yazı tipini değiştirebilirsiniz.
- Chart > Legend üzerinden göstergeyi sola, sağa, üste veya alta yerleştirebilirsiniz.

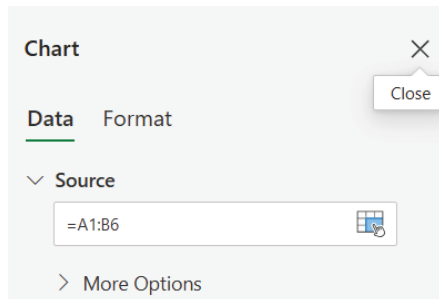
Veri Serilerinin Ayarlanması

- Sütunlara/çizgilere iki kez tıklayın > Chart > Format > Data Series bölümünden renkleri, stili ve kalınlığı değiştirebilirsiniz.

Veri Kaynağının Yeniden Seçilmesi

Grafiğin veri aralığını değiştirmek istiyorsanız:

- Grafiğe tıklayın, Chart > Select Data > Source yolunu izleyin.



Resim 27 Başlığın konumlandırılması

Değer Etiketlerinin Düzenlenmesi

- Değer etiketlerine iki kez tıklayın > Chart > Format > Data Labels bölümünden sayı biçimini, yazı tipini ve etiketlerin içeriğini değiştirebilirsiniz.
- Chart > Data Labels üzerinden veri değerlerinin gösterilip gösterilmeyeceğini seçebilirsiniz.



SONUÇ

Grafikler, verilerin görsel olarak sunulması için güçlü bir araçtır, bu sayede bilgilerin daha kolay anlaşılması ve analiz edilmesi sağlanır. Elektronik tablo programları ayrıca grafiklerin gelişmiş biçimde düzenlenmesine olanak tanır: grafik türünün değiştirilmesi, başlığın, eksenlerin, göstergenin ve veri serilerinin düzenlenmesi ve değer etiketlerinin ayarlanması mümkündür. Ayrıca, veri kaynağının yeniden seçilmesiyle grafikler yeniden oluşturulmalarına gerek kalmadan güncellenebilir.

BİLGİ KONTROLÜ SORULARI



1. Grafik nedir?
2. Grafik türü seçerken nelere dikkat etmek önemlidir?
3. Bir grafiğin temel öğeleri nelerdir?
4. X ekseninin, Y ekseninin rolü nedir?
5. Grafiğin göstergesi neyi ifade eder?
6. Değer etiketleri (data labels) nedir?
7. Daha önce oluşturulmuş bir grafiğin türünü nasıl değiştirebilirsin?
8. Grafiğin başlığını nasıl düzenleyebilirsin?
9. Göstergeyi nasıl taşıyabilir veya biçimlendirebilirsin?
10. Bir grafik için veri kaynağını yeniden nasıl seçebilirsin?
11. Grafiğin görsel olarak açık ve okunabilir olması neden önemlidir?
12. Verilen verileri en iyi şekilde gösterecek grafik türüne nasıl karar verirsın?
13. Değerler sütunların üzerinde gösterilmiyorsa grafik anlaşılır olur mu? Açıklayınız!

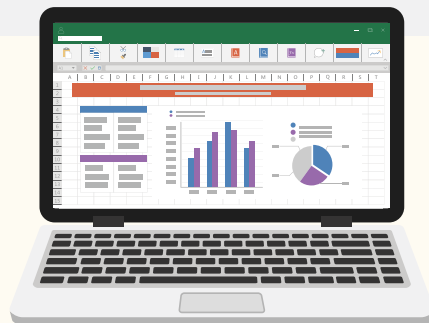


UYGULAMALI ALIŖTIRMA:

AŖAĖIDAKİ veriler bir hafta boyunca ölçölen sıcaklıkları göstermektedir:

Gün	Sıcaklık (°C)
Pazartesi	12
Salı	15
Çarşamba	17
Perşembe	14
Cuma	13
Cumartesi	16
Pazar	18

1. Verileri tabloya girin.
2. Sütun grafik oluşturun.
3. GrafiĖe başlık ekleyin.
4. Eksen başlıklarını („Günler“, „Sıcaklık“) ekleyin.
5. DeĖer etiketlerini (data labels) her sütunun üstünde gösterin.
6. Sütunların rengini deĖiştirin – maksimum sıcaklığı gösteren sütunu kırmızı, minimum sıcaklığı gösteren sütunu yeşil, diĖer sütunları ise mavi renge boyayın.
7. Bir veri satırı daha ekleyin ve grafiĖi genişletin.
8. Belgeyi kaydedin.
9. Grafik türünü dairesel grafiĖe (pie chart) dönüştürün.
10. Daire dilimlerinin etiketlerinde yüzdeleri gösterin.
11. Sonucunuzu paylaşın: Verileri hangi grafik türü daha açık bir şekilde göstermektedir?



5.4

Sıralama

Verileri sıralama, verilerin belirli bir düzene göre düzenlenmesi anlamına gelir. Bu, en temel ve en kullanışlı işlevlerden biridir. Sıralama, alfabetik olarak, sayısal olarak (küçükten büyüğe veya büyükten küçüğe), kronolojik olarak (en eski tarihten en yeni tarihe veya en yeni tarihten en eski tarihe), kullanıcı tarafından belirlenen bir düzene göre veya hücre rengi, yazı tipi rengi ya da simgeler dâhil olmak üzere biçime göre yapılabilir. Verileri sıralamanın temel amacı, verilerin daha iyi anlaşılmasını, düzenlenmesini ve bulunmasını sağlamaktır – sıralanmış veriler çok daha hızlı ve kolay bir şekilde aranabilir.

Dikkat etmen gerekenler: Sıralama yaparken tablonun tamamını seçmelisin – tablonun geri kalanını seçmeden yalnızca bir sütunu sıralarsan, satırlar birbirine karışır ve bu da yanlış verilere yol açar (örneğin, bir öğrencinin notu başka bir öğrencinin yanında görünebilir). Boş satır veya sütunlardan ve birleştirilmiş hücrelerden kaçınmalısın, bunlar sıralama sırasında sorunlara neden olabilir. Sütun başlıklarının eklenmesi önerilir.

HATIRLAYALIM:

Sıralama, Home > Sort & Filter seçilerek etkinleştirilir. Bununla:

- Metinleri alfabetik sıraya göre (Sort A to Z) veya ters alfabetik sıraya göre (Sort Z to A) sıralayabilirsin.
- Sayıları küçükten büyüğe (Sort Smallest to Largest) veya büyükten küçüğe (Sort Largest to Smallest) sıralayabilirsin.
- Tarih ve zamanı kronolojik olarak – en eskiden en yeniye (Sort Oldest to Newest) veya en yenden en eskiye (Sort Newest to Oldest) sıralayabilirsin).

...

Custom Sort (özel sıralama) seçeneği, verileri aynı anda birden fazla sütuna göre veya hücre rengi, yazı tipi rengi ve koşullu biçimlendirme uygulanmışsa simge gibi özel ölçütlere göre sıralamaya olanak tanır. Bu, özellikle alfabetik veya sayısal olarak yapılan standart sıralamanın yeterli olmadığı durumlarda kullanışlıdır.

Column	Sort On	Order
Sort by	Cell Values	Sort Ascending

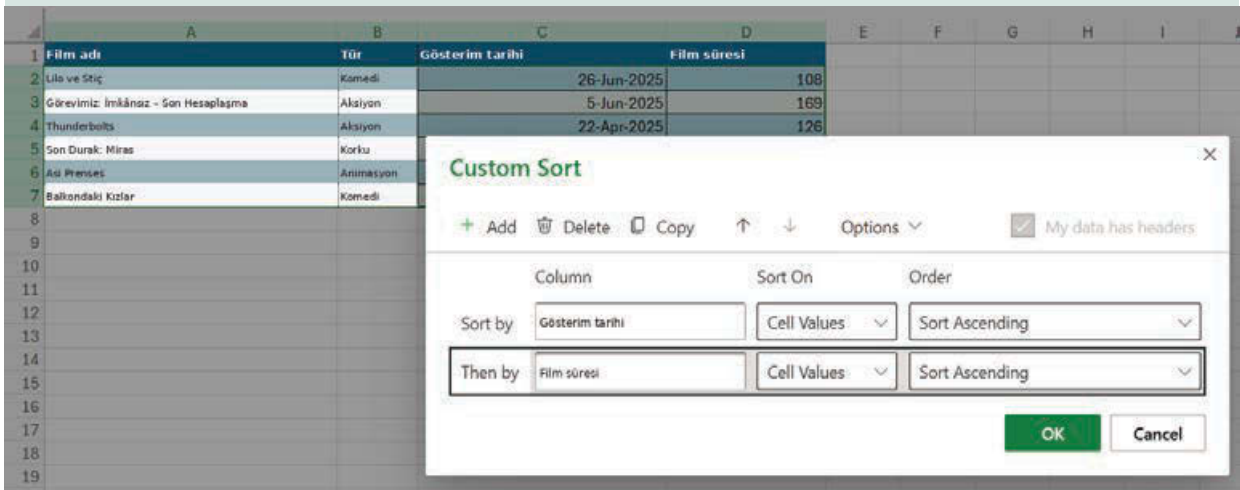
Resim 28

Custom Sort penceresi

Custom Sort penceresinde şunları seçmelisin:

- Sıralama yapacağın sütunu (Column alanında),
- Sıralama ölçütünü (Cell Values – hücrenin içeriğine göre, Cell Color – hücrenin rengine göre, Font Color – yazı tipi rengine göre veya Conditional Formatting Icon – koşullu biçimlendirme uygulandıysa simgeye göre)
- Sıralama düzenini (artan/azalan) seçtikten sonra OK seçilir.

Birden fazla ölçüt eklemek istiyorsan bunu, Custom Sort iletişim penceresinin sol üst köşesinde bulunan Add seçeneğine tıklayarak yapabilirsin. Yeni ölçütte Then by seçeneğinde sıralamanın nasıl yapılacağını aynı şekilde belirlersin. İlk belirtilen ölçütün önceliğe sahip olduğunu unutma. Sıralamanın gerçekleştirilmesi için son olarak ayarları OK seçeneğini kullanarak onaylamalısın.



Resim 29 İki ölçüte göre sıralama örneği

Bir ölçütü silmek için Delete seçeneğini seçin.

Sütun başlıklarınız varsa, diğer satırlarla karışmasını önlemek için My data has headers kutusunu işaretlemeyi unutmayın.



SONUÇ

Verilerin sıralanması, bilgilerin etkili bir şekilde yönetilmesi için en önemli ve en kullanışlı işlevlerden biridir. İster tek bir sütuna göre basit bir sıralama, ister birden fazla ölçüte göre gelişmiş özel bir sıralama söz konusu olsun, bu araç kullanıcıların verileri hızlı bir şekilde düzenlemelerine ve mantıklı, anlaşılır bir biçimde analiz etmelerine olanak tanır.

BİLGİ KONTROLÜ SORULARI VE ETKİNLİKLER



1. Sıralama nedir ve neden önemlidir?
2. İlişkili tüm sütunlar seçilmeden veriler sıralanırsa ne olur?
3. Excel’de Custom Sort nedir? Kendi sözlerinle açıklayınız!
4. Verileri aynı anda birden fazla sütuna göre nasıl sıralayabilirsin? Adımları açıklayınız!
5. Sıralama yapmadan önce tabloda başlıkların (headers) bulunması neden önemlidir?
6. Sıralama sırasında ortaya çıkabilecek yaygın sorunlardan bazıları nelerdir ve bunlar nasıl çözülebilir?
7. Sıralamayı ne zaman ve neden kullanacağına dair kendi örneğini vererek açıklayınız.



UYGULAMALI ALIŞTIRMA:

1. AŞAĞIDAKİ sütunları içeren bir tablo oluşturun: Öğrencinin adı, Öğrencinin soyadı, Doğum tarihi, Yazma süresi (öğrencinin bir metin programında „informatika“ kelimesini yazması için geçen süre). Tabloyu önce Yazma süresine, ardından Doğum tarihine göre sıralayın.
2. Okulunuzun, öğrenci takımlarının popüler video oyunlarında yarıştığı bir e-spor turnuvası düzenlediğini düşünün. Her oyuncu, antrenman maçlarındaki performansına göre belirli sayıda puan kazanmıştır ve maçlar haftanın farklı günlerine planlanmıştır.
 - AŞAĞIDAKİ etkinliklerin her birinden sonra sıralanmış tabloyu yeni bir çalışma sayfasına kopyalayın:
 - Oyuncuları puan sayısına göre, en yüksekten en düşüğe doğru sıralayın.
 - Önce Takıma, ardından Puana göre sıralayın (iki sütuna göre sıralama).
 - Tabloyu Maç Günü’ne göre sıralayın, günleri şu sıraya göre düzenleyin: Pazartesi, Salı, Çarşamba, Perşembe, Cuma.
 - Takım B’de en fazla puana sahip oyuncu kimdir?

Resim 30

-spor turnuvası

	A	B	C	D
1	Oyuncu adı	Puanlar	Maç günü	Takım
2	Ana	78	Cuma	Takım B
3	Jovan	85	Pazartesi	Takım A
4	Kristina	85	Çarşamba	Takım A
5	Aleksandar	65	Cuma	Takım B
6	Elena	90	Salı	Takım A
7	Stefan	72	Perşembe	Takım B

3. Günümüzde YouTube, özellikle gençler arasında eğlence, eğitim ve yaratıcı ifade için en popüler platformlardan biridir. AŞAĞIDAKİ tabloda YouTube'daki bazı tanınmış kanallar gösterilmektedir. Tabloda kanalın adı, kategorisi, abone sayısı (milyon olarak) ve platformda faaliyete başladığı tarih ile ilgili bilgiler yer almaktadır.

	A	B	C	D
1	Kanal	Kategori	Abone sayısı (milyon)	Başlangıç tarihi
2	T-Series	Müzik	266	13.03.2006
3	Dude Perfect	Spor/Eğlence	61	16.03.2009
4	TED-Ed	Eğitim	22	22.03.2012
5	Lozano Music	Müzik	0.29	08.02.2012
6	MKD Education	Eğitim	0.04	25.10.2020
7	Slatkaristika Off.	Müzik	0.31	20.09.2013
8	Fitness so Aleks	Sağlık/Fitness	0.065	03.07.2019
9	Kids Learn MK	Eğitim	0.08	18.03.2019

Resim 31 Ünlü YouTube kanallar

Her bir etkinlikten sonra sıralanmış tabloyu yeni bir çalışma sayfasına kopyalayın:

1. „Abone sayısı (milyon)” sütununa göre – en çoktan en aza doğru sıralayın.
2. „Abone sayısı (milyon)” sütununa göre – artan düzende sıralayın.
3. „Başlangıç tarihi”ne göre – en eski kanaldan en yeni kanala doğru, ardından „Kategori”ye göre sıralayın.



5.5

Filtreleme

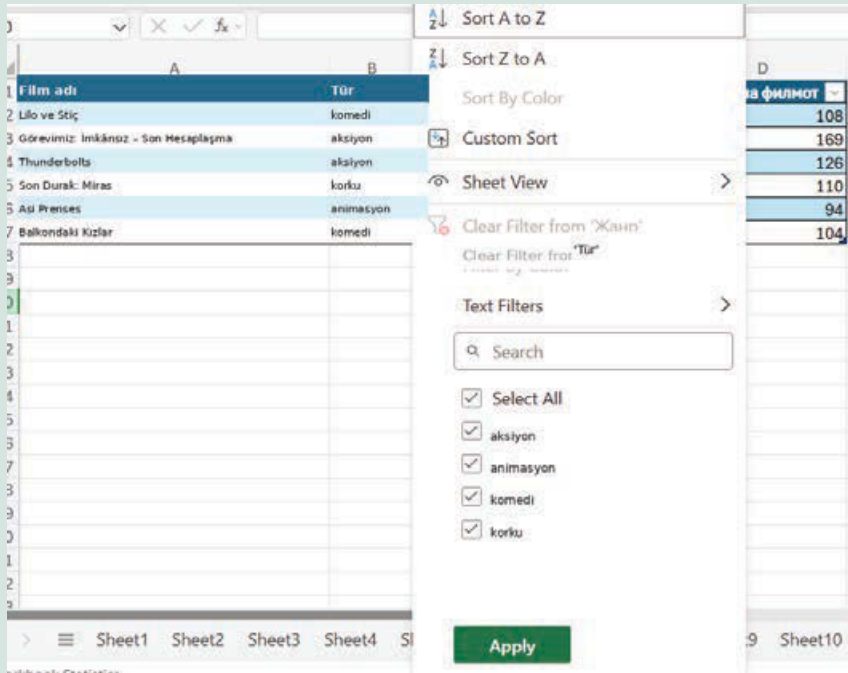
Verilerin filtrelenebilirliği, bir veri tabanında yalnızca belirli ölçütleri karşılayan verilerin gösterilmesini („geçirilmesini“) sağlarken, diğer verileri gizler (silmez). Filtrelerin kullanılmasıyla, yeniden düzenleme yapmaya gerek kalmadan bir alt tablo düzenlenebilir, biçimlendirilebilir ve yazdırılabilir. Filtrelemenin temel amacı, verilerle çalışmayı kolaylaştırmak, belirli bilgileri bulmak ve özgün verileri değiştirmeden veya silmeden verileri daha basit bir şekilde analiz etmektir. Hesaplama programlarında farklı veri türleri filtrelenebilir: metin (örneğin adlar, şehirler, kategoriler), sayılar (örneğin satış tutarları, miktarlar, yüzdeler), tarihler (örneğin sipariş tarihleri, doğum tarihleri), mantıksal değerler (TRUE/FALSE).

HATIRLAYALIM:

Belirli Değerlerle Veri Filtreleme

Adımlar

1. Filtrelemek istediğimiz verilerin üzerine tıklanır.
2. Home → Sort & Filter → Filter seçilir.
3. Tablonun ilk satırında, her sütunda bir ok görünür. Ok tıklandığında, sütunda bulunan tüm verilerin ve bunların önünde onay kutularının yer aldığı bir liste açılır.



Resim 32 Belirli değerlerle veri filtreleme

4. Onay kutusunda yalnızca gerekli verilerin seçili (işaretli) kalması gerekir (onay kutusuna tıklanarak veriler seçilir).

5. Apply seçilerek onaylanır.

Ek Koşul Belirleyerek Veri Filtreleme

Adımlar:

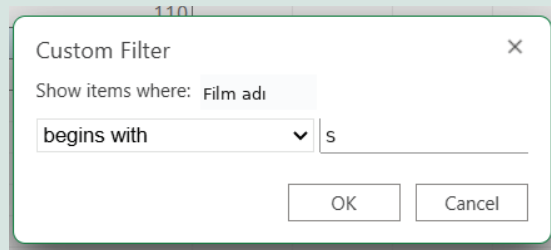
1. Filtrelemek istediğimiz verilerin üzerine tıklanır.
2. Home → Sort & Filter → Filter seçilir.
3. Tablonun ilk satırında, her sütunda bir ok görünür. Ok tıklandığında, metin/sayı/tarih filtresinin (Text Filters/Number Filters/Date Filters) seçilebileceği bir liste açılır.
4. Ek listeden, filtrelemenin yapılacağı seçenek belirlenir (örneğin: metin için – ... ile başlar, sayı için – ... değerinden küçüktür, tarih için – ... tarihinden öncedir).
5. Koşulun belirlenmesini tamamlamak için bir iletişim kutusu açılır.
6. OK seçilir.
7. And (VE) ve Or (VEYA) düğmeleri kullanılarak karmaşık koşullar oluşturulur.

ÖRNEK:

Bir tablonun şu verileri içerdiği varsayalım: Film adı, tür, gösterim tarihi ve filmin süresi. "H" harfiyle başlayan filmlerin adlarını görüntüleyin.

	A	B	C	D
1	Film adı	Tür	Gösterim tarihi	Film süresi
2	Thunderbolts	Aksiyon	22-Nisan-2025	126
3	Görevimiz: İmkânsız – Son Hesaplaşma	Aksiyon	5-Haziran-2025	169
4	Son Durak: Miras	Korku	6-Haziran-2025	110
5	Asi Prenses	Animasyon	16-Haziran-2025	94
6	Lilo ve Stiç	Komedi	26-Haziran-2025	108
7	Balkondaki Kızlar	Komedi	27-Haziran-2025	104

Resim 33 Film gösterimlerini inceleme tablo



Resim 34 Filtre ölçütlerinin belirlendiği pencere

	A	B	C	D
1	Film adı	Tür	Gösterim tarihi	Film süresi
5	Son Durak: Miras	Korku	6-Haziran-2025	110
8				

Resim 35 Filtrelenmiş tablo

Filtrenin Kaldırılması

1. Filtrenin uygulandığı sütundaki oka tıklanır.
2. Listedeki Clear Filter From „sütun adı“ seçilir.
3. Tüm veriler görüntülenir.

Çalışma sayfasındaki tüm filtrelerin kaldırılması

4. Home > Sort & Filter > Clear seçilir.



SONUÇ

Filtreler sayesinde kullanıcılar büyük tabloların karmaşıklığını azaltabilir ve yalnızca ilgili bilgilere odaklanabilir. Bu, verimliliği artırır, çalışmayı kolaylaştırır ve bilgilerin sunulması ile karar verme sürecinde açıklığı artırır. Ancak filtreler kullanılırken önemli verilerin gözden kaçırılmamasına ve filtrelenmiş verilerin genel durumu doğru biçimde yansıtmasına dikkat edilmelidir.

BİLGİ KONTROLÜ SORULARI



1. Excel’de filtreleme nedir?
2. Bir sütuna filtre uygulandığını nasıl anlarız?
3. Excel’de hangi veri türleri filtrelenebilir? (En az 3 tane belirtiniz.)
4. Filtre koşullarını karşılamayan verilere ne olur – silinir mi?
5. Değeri 5.000’den büyük olan satırları nasıl filtreleyebilirsiniz?
6. Bir tabloda uygulanmış tüm filtreleri nasıl kaldırabilirsiniz?
7. Aynı anda birden fazla sütuna filtre uygulayabilir misiniz? Nasıl yapıldığını açıklayınız!

8. Metne göre filtre uyguladığında hangi seçenekleri kullanabilirsin (örneğin: „ile başlar“, „içerir“...)?
9. 100'den fazla satırın bulunduğu bir tabloda „Not“ sütunu (1'den 5'e kadar) vardır. Yalnızca notu 4 ve 5 olan satırları nasıl filtrelersin?
10. „Fiyat“ sütunun var. Yalnızca fiyatı 1.000 ile 3.000 denar arasında olan ürünleri nasıl görüntülersin?
11. Filtre uygularken boş satırların veya hücrelerin bulunması durumunda hangi sorunlar ortaya çıkabilir?
12. Verilerin daha iyi analiz edilmesi için filtrelemeyi sıralama ile nasıl birleştirebilirsin?
13. Filtre kullanmadan önce sütun başlıklarını belirlemek neden önemlidir?



UYGULAMALI ALIŞTIRMA:

1. „Gençlerin Favori Ürünlerine Yaptıkları Harcamalar“ adlı bir tablo oluşturun. Sütun adları Ad, Kategori, Ürün, Fiyat (denar), Satın Alınan Ay ve Puan (1–5) olacak şekilde tabloyu oluşturun ve görselde gösterildiği gibi doldurun.

	A	B	C	D	E	F
1	İsim	Kategori	Ürün	Fiyat	Satın alma ayı	Puan (1-5)
2	Mila	Kulaklık	JBL Tune 510BT	3,200.00 den.	Ocak	5
3	Aleks	Oyun ekipmanları	Razer Mouse Viper	4,500.00 den.	Şubat	4
4	Sara	Akıllı cihaz	Xiaomi Smart Band 8	2,800.00 den.	Ocak	5
5	Melek	Oyun ekipmanları	Xbox kontrolcüsü	4,200.00 den.	Mart	3
6	Hana	Kulaklık	Sony WH-CH510	3,900.00 den.	Şubat	4
7	Osman	Akıllı cihaz	Apple AirTag	1,800.00 den.	Mart	5
8	Ana	Oyun ekipmanları	RGB Gaming Keyboard	5,500.00 den.	Ocak	5
9						

Resim 36 Gençlerin favori ürünlerine yaptıkları harcamalar tablosu

Her bir etkinlikten sonra filtrelenmiş tabloyu yeni bir çalışma sayfasına kopyalayın:

- 1) Kulaklık satın alan öğrencileri filtreleyin.
- 2) Yalnızca puanı 5 olan ürünleri görüntüleyin.
- 3) Şubat ayında satın alınan ürünleri filtreleyin.
- 4) „Oyun ekipmanları“ kategorisindeki tüm ürünleri görüntüleyin.

- 5) 4.000 denardan daha pahalı olan ürünleri filtreleyin.
- 6) Tabloyu kendi verilerinizle tamamlayın, ad, kategori, ay veya puana göre farklı **filtreler uygulayın ve satın alma alışkanlıkları hakkında kısa bir analiz yapın.**

2. „Gençlerin Sosyal Medyadaki Etkinlikleri“ adlı bir tablo oluşturun. Sütun adları Ad, Platform, Etkinlik Türü, Günlük Süre (dk), Favori İçerik olacak şekilde tabloyu oluşturun ve görselde gösterildiği gibi doldurun.

	A	B	C	D	E
1	İsim	Platform	Etkinlik türü	Günlük süre (dk)	Favori içerik
2	İva	Instagram	Hikâye paylaşma	90	Moda
3	Marko	YouTube	Video izleme	120	Oyun
4	Elena	TikTok	Video oluşturma	80	Dans
5	Marija	TikTok	Kaydırma	60	Komedi
6	Stefan	Instagram	Beğeni ve yorumlar	45	Fitness
7	Ana	YouTube	Video izleme	100	Bilim
8	Emin	facebook	Beğeni ve yorumlar	30	Moda
9	Ayşe	Instagram	Hikâye paylaşma	80	Oyun

Resim 37 Sosyal medyada gençlerin etkinlikleri tablosu

Belirtilen etkinliklerin her birinden sonra filtrelenmiş tabloyu yeni bir çalışma sayfasına kopyalayın:

- 1) Tabloyu kendi verilerinizle tamamlayın.
- 2) TikTok kullanan öğrencileri filtreleyin.
- 3) Platformda günde 50 ile 85 dakika arasında zaman geçirenleri görüntüleyin.
- 4) Platformda günde 100 dakikadan fazla zaman geçirenleri görüntüleyin.
- 5) Platform olarak „Instagram“, etkinlik olarak „Hikâye paylaşma“ seçerek filtreleyin.
- 6) Bu çalışmayı dijital alışkanlıklar hakkında tartışma yapmak için kullanın.



5.6

Rapor Oluřturma – Pivot Tablo

Pivot tablo, büyük miktardaki verileri hızlı bir şekilde özetlemeyi, gruplandırmayı ve analiz etmeyi sağlayan gelişmiş bir araçtır. Manuel olarak saymak veya hesaplamak yerine, pivot tablo birkaç tıklamayla istatistik elde etmeyi sağlar – kaç ürün satıldığı, hangi ayda en fazla gelir elde edildiği, bir ankette her cevabın kaç oy aldığı vb.

Önemli: Öncelikle verileri hazırlamalısınız – tablolarda boş alanlar bulunmamalı ve her sütunun bir adı olmalıdır.

	A	B	C	D	E	F	G	H	I
1	ID	Ürün	Kategori	Miktar	Birim Fiyat	Toplam Satış	Ay	Şehir	Çalışan
2	1	Tiřört	Giyim	15	800	12000	Ocak	Üsküp	Marija Stojanova
3	2	Spor ayakkabı	Giyim	10	2500	25000	Şubat	Manastır	Aleksandar Janev
4	3	Telefon	Elektronik	5	25000	125000	Ocak	Üsküp	Ivan Petrov
5	4	Dizüstü bilgisayar	Elektronik	2	45000	90000	Mart	Ohri	Elena Naumovska
6	5	Çanta	Aksesuar	7	3000	21000	Şubat	Manastır	Marija Stojanova
7	6	Kulaklık	Elektronik	12	1500	18000	Nisan	Kalkandelen	Bojan Krstev
8	7	Mont	Giyim	8	4000	32000	Mart	Kumanova	Aleksandar Janev
9	8	Eldiven	Aksesuar	3	900	2700	Mart	Üsküp	Veles
10	9	Sırt çantası	Aksesuar	6	2200	13200	Nisan	Üsküp	Elena Naumovska
11	10	Akıllı saat	Elektronik	3	7000	21000	Şubat	Kalkandelen	Ivan Petrov
12	11	Gözlük	Aksesuar	4	2800	11200	Ocak	Manastır	Marija Stojanova
13	12	Tiřört	Giyim	10	800	8000	Nisan	Kumanova	Bojan Krstev
14	13	Spor ayakkabı	Giyim	9	2500	22500	Mart	Üsküp	Aleksandar Janev
15	14	Telefon	Elektronik	6	24000	144000	Nisan	Manastır	Elena Naumovska
16	15	Dizüstü bilgisayar	Elektronik	3	45000	135000	Ocak	Ohri	Ivan Petrov
17	16	Kulaklık	Elektronik	10	1500	15000	Şubat	Üsküp	Bojan Krstev

Resim 38 „Satış“ tablosu

Adımlar:

1. Tablonun herhangi bir yerine tıklanır.
2. Insert > PivotTable seçilir.
3. PivotTable penceresi açılır. Burada veri aralığı (Table/Range) seçilir ve ardından yeni tablonun konumu belirleni (New worksheet – yeni çalışma sayfası (önerilir), Existing Worksheet – mevcut çalışma sayfası). Seçim OK düğmesine tıklanarak onaylanır.

Create PivotTable

Choose the data that you want to analyze

Select a table or range

Table/Range:

Choose where you want the PivotTable report to be placed

☒ New Worksheet

☐ Existing Worksheet

Location:

OK Cancel

Resim 39 Create PivotTable

PivotTable Fields

Choose fields to add to the report and drag them between the areas below:

Search

Filters

Rows

Columns

Values

Toplam satış toplamı

Resim 40 „Satış“ Tablosundaki pivot tablo alanları

4. PivotTable hâlâ boşdur. Sağ tarafta sütun adları ve alanların sürüklenebileceği dört bölümden oluşan bir panel gösterilir:
5. Rows (Satırlar): Gruplamak istediğin kategoriler (örneğin, kategori).
6. Columns (Sütunlar): Başka bir değere göre dağılım oluşturmak için kullanılır (örneğin, ay).
7. Values (Değerler): Hesaplamak istediğin değer (toplam, ortalama, sayı vb.).
8. Filters (Filtreler): Tabloyu belirli ölçütlere göre filtrelemek için kullanılır.

Bu örnekte, nihai sonucun kategorilere ve aylara göre toplam satışları gösteren bir PivotTable olması gerekiyorsa, sağ panel AŞAĞIDAKİ şekilde görünecektir:

Elde edilen PivotTable AŞAĞIDAKİ biçimde olacaktır:

3	Toplam satış toplamı	Ay				
4	Kategori	Nisan	Ocak	Mart	Şubat	Genel Toplam
5	Aksesuar	13200	11200	2700	21000	48100
6	Elektronik	162000	260000	90000	36000	548000
7	Giyim	8000	12000	54500	25000	99500
8	Genel Toplam	183200	283200	147200	82000	695600

Resim 41 „Satış“ tablosundan Pivot Tablo

Kaynak tablodaki verileri değiştirdikten sonra PivotTable'ı güncellemek (yenilemek) için yöntemlerden biri şudur: PivotTable'ın herhangi bir yerinde sağ tıklayın ve açılan menüden Refresh (Yenile) seçeneğini seçin. Bunun ardından tablo, kaynaktaki en güncel verileri otomatik olarak alacaktır.



SONUÇ

Pivot tabloların kullanımı, özellikle esneklikleri, hızları ve gelişmiş formüllere ihtiyaç duymadan büyük miktarda bilgiyi işleyebilme yetenekleri nedeniyle veri analizi sürecinde büyük önem taşır. Ancak bunların verimliliği, başlangıç verilerinin kalitesine ve kullanıcının doğru parametreleri belirleme becerisine bağlıdır. Her pivot tablo analizine iyi hazırlanmış ve temiz verilerle başlanması, satırlar ve sütunlar için anlamlı kategorilerin kullanılması ve analiz için açık bir amacın belirlenmesi önerilir.

Pivot tablolar, bilgileri kısa ve analitik bir şekilde sunarak özellikle iş bağlamında karar verme yeteneğini önemli ölçüde geliştirir. Finansal analizde, giderlerin takibinde, müşteri memnuniyeti analizinde veya operasyonel planlamada kullanılıp kullanılmadığı fark etmeksizin, veriye dayalı bilinçli karar vermeye katkıda bulunurlar.

BİLGİ KONTROLÜ SORULARI



1. Pivot tablolar en çok ne amaçla kullanılır?
2. Pivot tablolarla analiz yapmak için hangi tür veriler uygundur?
3. Bir pivot tablo oluşturmadan önce verilerin sütun başlıklarına sahip olması neden önemlidir?
4. Pivot tabloda hangi hesaplama fonksiyonları kullanılabilir (örneğin toplam, ortalama)?
5. Kaynak tablodaki veriler değiştirilirse pivot tabloya ne olur?
6. Veriler değiştirildikten sonra pivot tablo nasıl güncellenir (refresh)?
7. Pivot tablolarla çalışırken en sık yapılan hatalardan bazıları nelerdir?



UYGULAMALI ALIŞTIRMA:

- Okulunda birçok etkinlik düzenlenmektedir – her ay atölye çalışmaları, yarışmalar, spor günleri ve yaratıcı projeler gerçekleştirilmektedir. Bazı öğrenciler yarışmalara katılır, bazıları yeni beceriler öğrenir, bazıları ise organizasyona yardımcı olur. Öğretmenler kurulu, öğrencilerin etkinliklere katılımını ve bu etkinliklerin etkisini daha iyi anlamak istemektedir. Sana, Excel’de PivotTable (Özet Tablo) kullanarak etkinliklerin analizini yapma görevi verilmektedir.

Kullanacağınız kaynak tabloda şu sütunlar yer almaktadır:

- Öğrencinin adı
- Sınıf
- Etkinlik türü – Atölye çalışması, yarışma, spor günü, sunum vb.
- Etkinlik tarihi – Etkinliğin gerçekleştiği tarih
- Süre (dakika) – Etkinliğin ne kadar sürdüğü

Pivot tablo ile AŞAĞIDAKİ gibi sorulara cevap verilmelidir:

- En yaygın etkinlik türü hangisidir?
- Hangi sınıflar en aktiftir?
- Etkinliklerin ortalama süresi ne kadardır?
- Hangi öğrencilerin katılım sayısı en fazladır?

	A	B	C	D	E
1	Öğrenci adı	Sınıf	Etkinlik türü	Etkinlik tarihi	Süre (dakika)
2	Ana Petrova	VIII-1	Yarışma	12.03.2024	90
3	Jovan Stojkov	IX-2	Atölye	15.03.2024	120
4	Eva Miloševska	VIII-2	Spor Günü	20.03.2024	60
5	Petar Jovanov	IX-2	Atölye	25.03.2024	90
6	Ana Petrova	VIII-1	Yarışma	05.04.2024	95
7	Eva Miloševska	VIII-2	Sunum	10.04.2024	45
8	Jovan Stojkov	IX-2	Yarışma	12.04.2024	100

Resim 42 "Öğrenci Etkinlikleri" tablosu

Kendi kendini değerlendirme için çözümler

- Rows:** Etkinlik Türü
Values: Count of Etkinlik Türü
- Rows:** Sınıf
Values: Count of Öğrenci Adı
- Rows:** Etkinlik Türü (veya Sınıf)
Values: Average Sürenin Ortalaması (dakika)
- Rows:** Öğrenci Adı
Values: Count of Etkinlik türü



NOT:

Values alanındaki işlevi (örneğin Sum, Count, Average) değiştirmek için, işlev adının sağındaki oka tıklayın, Value Field Settings... seçeneğini belirleyin ve buradan istediğiniz işlevi seçin.

Gerçek hayatta tablolarla çalışırken sık sık önemli veriler girilir, bunlar notlar, finansal raporlar, programlar veya planlar olabilir. Ancak birisi yanlışlıkla bir formülü silerse ne olur? Ya da birisi sayı yerine metin gibi geçersiz bir değer girerse? Bu tür hataları önlemek için veri koruma ve veri denetimi araçları bulunmaktadır.

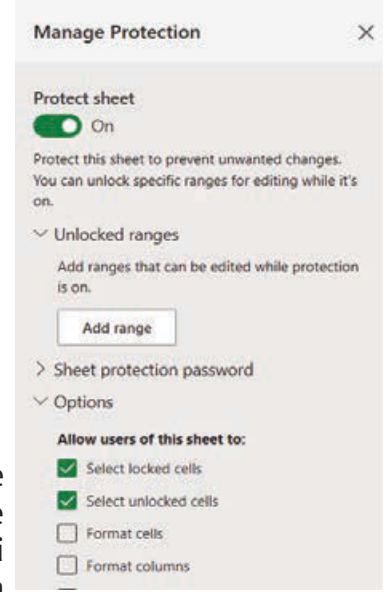
Excel 365'te Protect işlevi, çalışma sayfalarını, hücreleri veya belgenin tamamını yetkisiz değişikliklere karşı korumak için kullanılır.

Tüm çalışma Sayfasının Korunması

Adımlar:

1. Review sekmesini seçin.
2. Protection → Manage Protection seçeneğine tıklayın.
3. Bir parola girin (isteğe bağlı).
4. Kullanıcıların neleri değiştirebileceğini seçin.
(Örneğin, hücreleri biçimlendirme veya seçme.)

Önemli: Excel belgesi PDF, CSV veya HTML gibi başka bir biçime aktarıldığında, Excel'de ayarladığınız koruma diğer biçime aktarılmaz. Belgenin korumalı olmasını istiyorsanız, belgeyi Excel (.xlsx) biçiminde kaydetmelisiniz. Belge OneDrive'daysa File → Save a Copy seçeneğini kullanabilirsiniz.



Resim 43

Manage Protection

NOT: Excel 365'in çevrim içi sürümünde koruma seçenekleri masaüstü sürümünden biraz farklıdır. Burada tüm çalışma sayfasını koruyabilir veya Manage Protection aracılığıyla düzenlemeyi sınırlandırabilirsiniz. Add range seçeneği sayesinde, çalışma sayfası kilitliken hangi bölümlerin değiştirilebileceğini belirleyebilirsiniz.

Excel'de tüm hücreler varsayılan olarak "Locked" (Kilitli) durumdadır, ancak çalışma sayfası koruması etkinleştirilmeden bunun herhangi bir etkisi yoktur. Bu nedenle yalnızca bazı hücreleri kilitlemek istiyorsanız, önce tüm hücrelerin kilidini açmalı, ardından yalnızca gerekli hücreleri kilitlemelisiniz.

Tüm Hücrelerin Kilidini Açma

Adımlar:

1. Tüm hücreleri seçin (Ctrl+A).
2. Sağ tıklayın → Format Cells → Protection sekmesini seçin.
3. Locked seçeneğinin işaretini kaldırın.

Hücreleri Kilitleme

1. Değiştirilmesini istemediğiniz hücreleri seçin.
2. Sağ tıklayın → Format Cells → Protection sekmesini seçin.
3. Locked seçeneğini işaretleyin.
4. Review → Protect Sheet seçeneğine giderek korumayı etkinleştirin.

Artık "Locked" olarak işaretlediğiniz hücreler değiştirilemez, diğer hücreler ise düzenlenebilir durumda kalır!

Not: Formülleri ve işlevleri gizlemek istiyorsanız, işlem hücreleri kilitleme ile aynıdır, ancak Locked yerine Hide seçeneğini işaretlemelisiniz.

Tüm Çalışma Kitabının Korunması

Adımlar:

1. File → Info seçeneğine gidin.
2. Protect Workbook →
3. Encrypt with Password seçeneğine tıklayın.
4. Bir parola girin ve onaylayın.

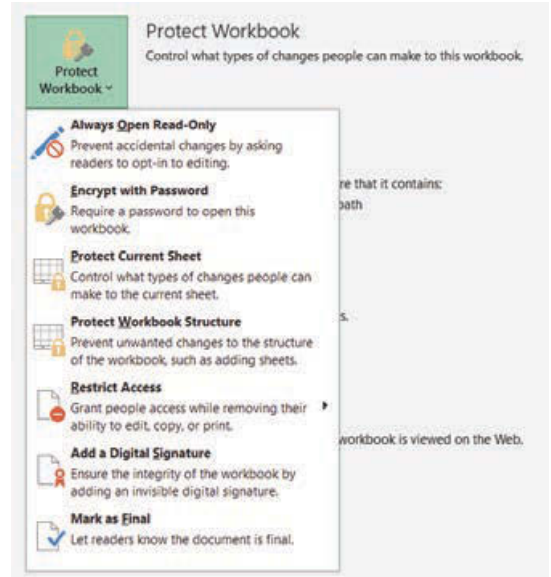
Veri doğrulama

Excel'de öğrencilerin 1 ile 5 arasında notlar girmesi gereken bir tablo oluşturduğunuzu düşünün. Ancak bir öğrenci yanlışlıkla 7, başka bir öğrenci ise sayı yerine metin girmiştir. Bu tür hatalar hesaplamalarda yanlış sonuçlara yol açabilir. İşte veri doğrulama (Data Validation) bu soruna çözüm sağlar.

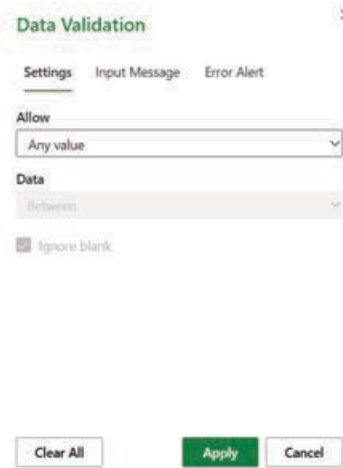
Veri doğrulama, belirli hücrelere hangi değerlerin girilebileceğine ilişkin kurallar belirlenmesini sağlar. Örneğin yalnızca 1 ile 10 arasındaki sayılara, yalnızca bugünün tarihinden sonraki tarihlere izin verebilir veya önceden belirlenmiş değerlerden oluşan bir açılır liste oluşturabilirsiniz. Böylece hatalar önlenir, verilerin doğruluğu kontrol edilir ve bilgilerin işlenmesi kolaylaştırılır.

Veri doğrulama, Data → Data Validation seçilerek etkinleştirilir.

Allow bölümünde, seçilen hücreye veya hücre aralığına hangi tür verilerin girilmesine izin verileceği belirlenir. İzin verilen değerler şunlardır:



Resim 44 Protect Workbook penceresi



Resim 45 Data Validation penceresi

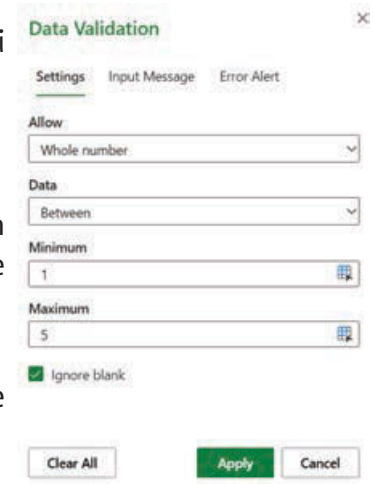
Seçenek	Açıklama
Any value	Her şeye izin verir (herhangi bir sınırlama yoktur).
Whole number	Yalnızca tam sayılara izin verir (ondalık sayı olmadan).
Decimal	Ondalıkli sayılara izin verir.
List	Önceden tanımlanmış bir listeden seçim yapılmasına izin verir.
Date	Yalnızca tarih girilmesine izin verir.
Time	Yalnızca saat girilmesine izin verir.
Text length	Girilebilecek karakter sayısını sınırlar.
Custom	Formül ile özel bir kural belirlemeye izin verir.

Doğrulama ayarlama (tam sayı örneği)

Adımlar:

- 1 ile 5 arasında bir tam sayı girilmesini istediğiniz hücreleri seçin.
- Data > Data Validation seçeneğine gidin.
- Allow bölümünden Whole number seçeneğini belirleyin.
- Data Validation penceresinde Data bölümünden between seçeneğini belirleyin. Minimum alanına 1, Maximum alanına ise 5 yazın.
- Seçimi Apply düğmesine tıklayarak onaylayın.

Artık bu hücrelere tanımlanan aralık (1 ile 5) dışında bir sayı girmeye çalıştığınızda bir hata mesajı görüntülenir.



Resim 46 Tam sayı için Data Validation penceresi

Doğrulama Ayarlama

(Liste örneği, değerlerin doğrudan girilmesi)

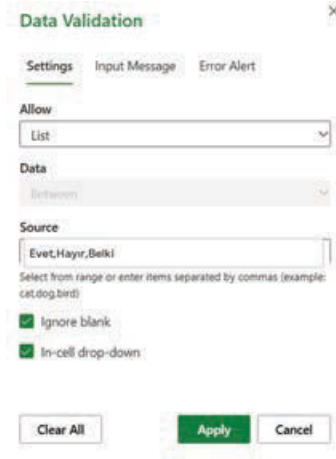
Öğrencilerin önceden belirlenmiş birkaç seçenekten birini, örneğin "Evet", "Hayır" veya "Belki" seçeneklerinden birini seçmeleri gereken bir tablo oluşturduğunuzu düşünün. Herkesin seçenekleri kendi istediği şekilde yazması ve veri girişinde hatalar oluşması yerine, önceden hazırlanmış seçenekleri bir açılır menü olarak sunabilirsiniz.

Adımlar:

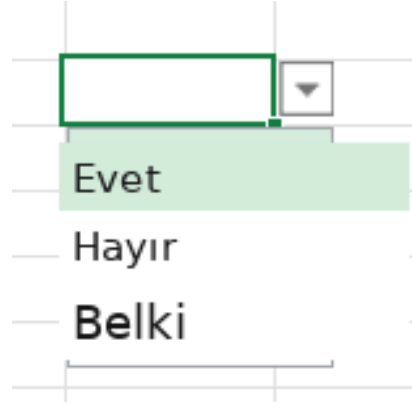
1. Açılır liste eklemek istediğiniz hücreyi veya hücre aralığını seçin.
2. Data > Data Validation seçeneğini belirleyin.
3. Açılan pencerede:
 - Allow (İzin ver) alanından List (Liste) seçeneğini belirleyin.
 - Source (Kaynak) alanına seçenekleri doğrudan virgüllerle ayırarak girin.

4. Seçimi onayla (Apply).

Şimdi seçilen alanda (veya hücre aralığında), hücrelerin sağ tarafında bir ok işareti görünecektir. Bu oka tıkladığında bir açılır liste açılır ve buradan sunulan seçeneklerden birini seçmen gerekir.



Resim 47 Tam sayı için Data Validation penceresi



Resim 48 Hücrede açılır menü



NOT:

Data Validation penceresinde iki sekme daha vardır:

- Input Message, veri doğrulamasının ayarlandığı hücreye kullanıcı tıkladığında görünen bilgilendirme mesajını yazmak için kullanılır.
- Error Alert, biri veri doğrulama kurallarına göre izin verilmeyen bir değer girdiğinde görüntülenmesini istediğin mesajı yazabileceğin bölümdür.



SONUÇ

Bir tablo oluşturduğunu ve öğrencilerin “Evet”, “Hayır” veya “Belki” gibi önceden tanımlanmış birkaç cevaptan birini seçmeleri gerektiğini düşün. Herkesin kendi istediği şekilde yazması ve veri girişinde hataların oluşması yerine, önceden onlara açılır menüden seçim yapma imkânı sunabilirsin.

BİLGİ KONTROLÜ SORULARI



1. Excel'de hücreleri kilitlemek ne anlama gelir ve hücreler hemen korunmuş hâle gelir mi?
2. Yalnızca bazı hücrelerin düzenlenmesine izin vermek istiyorsan yapman gereken ilk adım nedir?
3. Hücreleri kilitledikten sonra korumayı etkinleştirmek için yapman gereken ikinci adım nedir?
4. Excel 365'te bir çalışma sayfası nasıl korunur?
5. „Sayfa koruması“ ile „çalışma kitabı koruması“ arasındaki fark nedir?
6. Bir çalışma sayfasını korurken hangi seçenekleri etkinleştirebilir veya devre dışı bırakabilirsin?
7. Diğer kullanıcıların yalnızca belirli hücelere veri girmesine izin verilebilir mi?
8. Bir çalışma sayfasını veya çalışma kitabını korurken parola belirlemek zorunlu mudur?
9. Korunmalı bir çalışma sayfasında kilitli bir hücreyi düzenlemeye çalışırsan ne olur?
10. Bir çalışma sayfasının veya çalışma kitabının koruması nasıl kaldırılır (devre dışı bırakılır)?
11. Veri doğrulama nedir?
12. Veri doğrulama kullanmanın amacı nedir?
13. Doğrulama penceresindeki „Allow“ açılır menüsünde hangi seçenekler bulunur?
14. Bir değerler listesi girerken „Source“ alanında ne bulunmalıdır?
15. Birisi veri doğrulamasına göre izin verilmeyen bir değer girerse ne olur?
16. Aynı veri doğrulamayı birden fazla hücreye aynı anda uygulayabilir misin? Nasıl?



UYGULAMALI ETKİNLİK



1. AŞAĞIDAKİ verilere sahip bir Excel tablonuz bulunmaktadır:

Ad	Soyadı	Not 1	Not 2	Toplam
Marko	Yanev	5	4	9
Elvir	Baku	4	5	9
İvisa	Stoyanov	3	4	7

- "Not 1" ve "Not 2" sütunları dışındaki hücreleri kilitleyin!

2. Ürünler tablosundaki boş alanları uygun şekilde, gizleyeceğiniz formülleri kullanarak doldurun!

Ürün	Birim fiyat	Miktar	Toplam
Ekmek	40	2	
Süt	60	3	
Meyve suyu	90	1	
Şeker	50	2	
Toplam			

3. B sütununda (B2 ile B10 arasındaki hücrelerde), şu seçeneklerden oluşan bir açılır liste oluşturun: I. sınıf, II. sınıf, III. sınıf, IV. sınıf!

4. C sütununa öğrencilerin yaşlarını girin! Yalnızca 14 ile 19 arasındaki tam sayıların girilmesine izin ver.

5. D sütununa kayıt tarihini girin! Yalnızca 01.09.2024 ile 30.09.2025 arasındaki tarihlerinin girilmesine izin ver.



KONU 5 İÇİN TEKRAR

2. Verileri hesaplamak için Excel'de kullanılan üç temel formül veya fonksiyon belirtiniz (örneğin: SUM, AVERAGE, IF)!
3. Excel'de „göreceli adresleme“ kavramının ne anlama geldiğini açıklayınız ve „mutlak adresleme“ ile arasındaki farkı belirtiniz!
4. Öğrencilerin isimleri ve notlarından oluşan bir tablo oluşturunuz! Her öğrenci için ortalama hesaplamak üzere bir formül uygulayınız!
5. Aynı tabloda, 2,50'nin altındaki tüm notları kırmızı renkle işaretlemek için koşullu biçimlendirme uygulayınız!
6. İsim, Ders, Not, Tarih sütunlarından oluşan bir veri tabanınız var. Yalnızca notu ≥ 4 olan öğrencileri filtreleyiniz ve sonuçları nasıl elde ettiğinizi açıklayınız!
7. Ay boyunca notlardaki değişimi göstermek için hangi grafik türünün (çizgi, sütun veya pasta grafik) en uygun olduğunu analiz ediniz. Seçiminizi gerekçelendiriniz!
8. Gider ve gelirlerden oluşan bir tablo verilmiştir. Bütçenin pozitif veya negatif olup olmadığını kontrol etmek için IF fonksiyonunun en iyi seçim olup olmadığını değerlendiriniz! Kararınızı açıklayınız!
9. Önemli finansal veriler içeren bir tablonuz var. Hangi koruma türünün daha gerekli olduğuna karar veriniz: hücreleri kilitleme, çalışma sayfasını koruma veya çalışma kitabının tamamını koruma! Gerekçenizi açıklayınız!
10. En az 20 kayıt içeren bir veri tabanından bir PivotTable (Özet Tablo) oluşturunuz (örneğin: satışlar, notlar, giderler)! Kategorilere göre toplamaları gösterecek şekilde düzenleyiniz!
 - AŞAĞIDAKİleri içeren kendi raporunuzu hazırlayınız:
 - Veri tabanı olarak bir tablo,
 - En az 3 gelişmiş fonksiyonla (IF, VLOOKUP/XLOOKUP, COUNTIF) hesaplamalar,
 - Seçtiğiniz bir grafik,
 - Hızlı arama için filtreler,
 - Bir PivotTable (Özet Tablo),
 - En önemli hücreler için temel koruma.