YAZAR

Mr. Andrijana Tomovska

BİLİŞİM TEKNOLOJİLERİ

DOKUZ YILLIK İLKÖĞRETİMDE 7. SINIF DERS KİTABI

2021

Yazar: Andrijana Tomovska, Mr.

Yayıncı: Kuzey Makedonya Cumhuriyeti
Eğitim ve Bilim Bakanlığı

Hakemler: Prof. Dr. Marika Apostolova Trpkovska
Zaklina Naumovska
Senar Fazliya

Çizer: Elena Trpcevska

Lektör: Slagjana Ginovska

Türkçeye çeviren: Mr. İlker Ali

Mesleki redaksiyon: Doç. Dr. Aybeyan Selimi

Lektör: Ahmetnafi İbrahim

Grafik ve teknik tasarım: Boban Zdravkovski Andreevski

Ulusal Ders Kitapları Komisyonu tarafından 16.03.2020 tarihi ve 26- 521/1 sayılı kararıyla dokuz yıllık ilköğretimde 7. sınıflar için Bilişim Teknolojileri konulu ders kitabı kabul edilmiştir.

Basimevi: Evropa 92, Koçana

Tiraz: 661

CIP - Каталогизација во публикација
Национална и универзитетска библиотека “Св. Климент Охридски”, Скопје

373.3.016:004(075.2)=512.161

TOMOVSKA, Andrijana
Bilişim teknolojileri dokuz yillik ilköğretimde 7. sinif ders kitabi / Andrijana Tomovska ; [çeviri İlker Ali]. - Üsküp : Makedonya cumhuriyeti eğitim ve Bilim Bakanliği, 2021. - 162 стр. : илустр. ; 30 см

Библиографија: стр. 160

ISBN 978-608-226-935-1

COBISS.MK-ID 54303749

ÖNSÖZ

Eğitim Geliştirme Bürosu tarafından oluşturulup tanımlanan ve Kuzey Makedonya Cumhuriyeti Eğitim ve Bilim Bakanlığı tarafından müfredata uygun olarak onaylanan, bu Bilişim Teknolojileri ders kitabı dokuz yıllık zorunlu ilköğretimde VII. Sınıf için oluşturulmuştur. Resmi müfredata göre ders kitabının beş konusu vardır.

İlk konu olan "Elektronik Tablolama Programı" nda öğrenciler Microsoft Office Excel 2016'nın arayüzü tanıtılacak ve elektronik tablolar oluşturup düzenleyebilecek, formül ve fonksiyonları kullanarak matematiksel hesaplamalar sergileyebilecekler ve son olarak verilerin grafiksel gösterimini yapabilecekler. Elektronik tablolama programının bu işlevlerinin pratik uygulaması için uygun sorular ve pratik ödevler oluşturuldu.

Bir sonraki konu, öğrencilerin uygun bir çözüm bulmak için belirli ödevlerin yardımıyla mantıksal düşünmeyi geliştirecekleri "Mantıksal yarışma ödevlerini çözerek, bilgi kavramlarını anlamak" konusudur. Mantıksal yarışma ödevlerini çözerken, öğrenciler mantıksal düşünmeyle çözme ve Bilişim Teknolojileri kavramlarıyla çözme arasındaki bağlantıyı kavrayacak ayrıca bunların bilgisayarlarla çalışmasındaki yerlerini öğreneceklerdir.

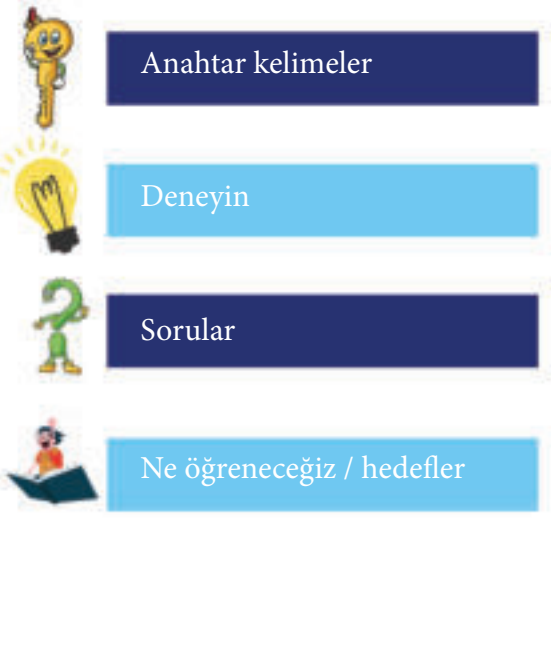
Üçüncü konu olan "Görsel ortamda ileri programlama" bölümünde, belirli bir etkinliği gerçekleştirecek olan mantık bloklarının yardımıyla bir önceki konuya bir bağlantı yapılacaktır. Bu şekilde, Scratch görsel programlama ortamını kullanarak basit programlar, etkileşimli oyunlar, hikayeler ve benzerleri yaratacaklar.

Dördüncü konu, "Standart bir yapısal programlama dili aracılığıyla programlama" konusuna ayrılmıştır. Bu amaçla entegre programlama ortamı Code::Blocks ve programlama dili C++ kullanacağız. Bir dizi örnek ve pratik görev aracılığıyla, öğrenciler program oluşturma konusunda beceri kazanacaklar.

Son konu olan "Çevrimiçi yaşam" da öğrenciler web günlükleri, yani internetin etik kullanımı ve güvenliği kurallarına saygı duymaya özel önem veren bloglar oluşturabilecekler.

Bu ders kitabında, pratik çalışmalar yeni öğretim birimlerinin benimsenmesi ve ustalaşması için temel oluşturur ve bu nedenle öğrencinin belirtilen konulardaki bilgi ve

becerilerini genişletmesini sağlayacak sorular ve görevler içeren özel bir bölüm hazırlanmıştır. Her öğretim ünitesi, öğrencilerin öğrenirken karşılaşacakları yeni anahtar kavramların altını çizmekle başlar ve öğretim ünitesinin en önemli unsurlarını gösteren bir değerlendirme listesi olarak ezberle ile biter. Her dersten sonra, öğrenilenleri tekrar etmeye yarayan sorular listelenir. Dersler boyunca öğrenciler, özel çerçevelere yerleştirilmiş ve özel olarak grafiksel işaretlenmiş hatırlatıcılar, ipuçları ve notlarla karşılaşacaklar. Ek olarak, ders kitabında kullanılan grafik öğelerin anlamlarının bir listesi verilmiştir.



Her konunun sonunda, konularla ilgili sorular ve ödevler içeren özel bir ek olarak "Tekrar edelim! Hadi pratik yapalım!" Bölümü oluşturulmuştur.

Ders kitabının sonunda, öğretim birimlerinin benimsenmesi sırasında ihtiyaç duyulan mesleki terimlerin ve eklerin açıklamalarının yer aldığı küçük bir sözlük oluşturulur. Ders kitabına ek olarak, tüm pratik çalışmaların yer aldığı bir CD oluşturuldu. CD'deki klasörler kitaptaki ders konularını ifade etmektedir.

Alıştırmalar, görevler, pratik çalışmalar ve sorular, dokuz yıllık ilköğretimde 7. sınıftaki öğrencilerin yaşına göre ve bu konuda önceki okul yıllarında edindikleri ön bilgiler doğrultusunda hazırlandı.

Yazar

İÇİNDEKİLER

1. Elektronik tablolama programına giriş 8
1.1 Elektronik Tablolama Programı - Microsoft OfficeExcel 2016 9
1.1.1 Programı başlatmak. Çalışma alanı öğeleri 9
1.1.2 Excel'de belgelerle çalışma 12
1.1.3 Çalışma sayfasının temel öğeleriyle çalışma 13
1.2 Tablodaki verilerle çalışma 16
1.2.1 Veri türleri 17
1.2.2 Sayısal verilerin formatı 18
1.3 Tabloyu düzenleme 21
1.3.1 Girilen verileri değiştirme ve düzenleme 21
1.3.2 Sütun ve satırlar ile işlemler 22
1.3.3 Sütun ve satırların boyutlarını değiştirme 23
1.4 Tabloyu biçimlendirme 25
2 Mantıksal rekabetçi ödevleri çözerek bilgi kavramlarına giriş 50
2.1 Mantıksal rekabetçi ödevleri analiz etme ve çözme 53
2.2 Görevin bilgisayar bilimindeki kavramlarla ilişkisi - 57
2.2.1 Veri yapıları 57
2.2.2 İkili sayılar 59
TEKRAR EDELİM! UYGULAYALIM! 64
3. Görsel bir ortamda programlamaya giriş 67
3.1 Grafik programlama. Scratch programına giriş 69
3.1.1 Scratch Başlatma 69
3.1.2 Scratch'in temel araçlarına giriş 70
3.1.3 Scratch'te basit programlar oluşturmak 72
3.2 Etkileşimli olay programları 77
3.3 Daha karmaşık sorunlara programların geliştirilmesi 83
TEKRAR EDELİM! UYGULAYALIM! 89
4 Standart bir yapısal programlama dili aracılığıyla programlamaya giriş 91
4.1 Bir program yapma süreci 93
4.2 Entegre bir programlama ortamının temel unsurlarına giriş 95
4.2.1 Code :: Blocks'un Yüklenmesi 95
4.2.2 Code::Blocks'un arayüzü 98
4.2.3 Yeni bir kaynak kod dosyası oluşturun 102
4.3 Program kodlarının görünümü 104
4.4 Hazır örnek programların yürütülmesi 107
4.5 C++ programlama dilinin temel öğeleri 111

4.6 Beyanlar 114
4.6.1 Ekran görüntüsü bildirimi 114
4.6.2 Değer atama beyanı 116
4.7. Programların geliştirilmesi 118
4.8 Aritmetik işlemler ve ifadeler 120
4.9. Sabitler ve değişkenler 122
4.9.1 Sabitler 122
4.10 Programa veri girmek için ifadeler (teknikler) 125
4.11 Karşılaştırmalı ifadeler 128
4.12 Bir koşul karşılanana kadar bir döngüde tekrarlama yapısı 135
TEKRARLAYALIM! UYGULAYALIM! 140
5. Web günlüklerine (bloglar) giriş 143
5.1 Blog kavramı ve uygulaması 145
5.2 Blog oluşturmak 147
5.2.1 Blog oluşturma adımları 147
5.2.2 Blog içeriğini düzenleme 149
5.2.3 Bir bloga resim ve video ekleme 150
5.2.4 Bloga bağlantı ekleme 151
5.2.5 Blog temasını / düzenini değiştirme 152
5.3. Bloglardaki içeriği yorumlama 154
TEKRAR EDELİM! UYGULAYALIM! 156
ANAHTAR KELİMELER SÖZLÜĞÜ 157
KULLANILAN LİTERATÜR 160

Elektronik Tablolama programı

Elektronik tablolama programına giriş
Elektronik Tablolama Programı - Microsoft Office Excel 2016
Tablodaki verilerle çalışma
Tabloyu düzenleme
Tabloyu biçimlendirme
Tablodaki verilerin otomatik olarak doldurulması
Elektronik tablolama programındaki formüller ve işlevler
Verileri sıralama
Grafik oluşturma
TEKRARLAYALIM! UYGULAYALIM!

1. Elektronik tablolama programına giriş

Ne öğreneceğiz?

Formülleri ve işlevleri kullanarak farklı veri türleri ve hesaplama dizileri içeren tablolar oluşturup, biçimlendirme araçlarıyla düzenleyip, grafiksel olarak temsil etmek.

Günlük yaşamdaki insanlar genellikle bir elektronik tabloya yerleştirilmiş çeşitli verilerle karşı karşıya kalır. Örneğin: Okullardaki öğretmenler, bir konudaki öğrenci başarısını izlemek ve kaydetmek, öğrenci başına ortalama başarıyı ve genel olarak sınıfın tamamını hesaplayabilmek, üç aylık başarı karşılaştırmaları yapabilmek veya bir ders programı gerçekleştirmek için elektronik tablolar vb. oluşturur. Okul kütüphanesi personeli, okulun kullanabileceği kitaplardan oluşan bir elektronik tablo yapabilir. Öğrenciler bu programı çeşitli okul ve ders dışı amaçlar için kullanabilecekler, örneğin: Sınıftaki öğrenciler için bir veri sayfası oluşturmak, bir telefon rehberi oluşturmak, okulun düzenliliği için bir program oluşturmak, ödevlerin düzenliliğini gözden geçirmek, bir projeyi oluştururken verilerilerin toplanıp işlenmesi, toplanan verilerin grafik gösterimi vb. Ebeveynler bu programı aile gelirini, aylık harcamaları takip etmek için kullanabilirler.

Bu nedenle, “Elektronik Tablolama Programı “ konusunun amacı, **MS Office Excel 2016** programında, elektronik tablolarla çalışmak, belirli işlevlere ve oluşturulan formüllere göre hesaplamalar yapmak ve verilerin grafik olarak gösterilmesidir.

Bu konuyu okuyan öğrenciler şunları yapabilecek:

- Bir elektronik tablolama belgesi oluşturmak;
- Farklı formatlarda veri girme;
- Tablo düzenleme / biçimlendirme;
- Formüller ve işlevler kullanarak hesaplamalar yapmak;
- Bir tablodaki verileri sıralamak;
- Bir tablodaki verileri filtreleme;
- Veri sunumu için farklı tiplerde grafikler oluşturmak.

Kavramların ve malzemenin teorik olarak benimsenmesine ek olarak, pratik çalışmaya özel bir vurgu yapacağız. Bunu yapmak için, Microsoft Office 2016 ofis paketini bilgisayarınıza yüklemeniz gerekir.

Şekil 1: Microsoft Excel 2016 simgesi

Anahtar kelimeler

Çalışma kitabı, çalışma sayfası, tablo, sütun, satır, hücre, sıralı / sıralı olmayan hücreler, etkin hücre, formül, işlev, grafik.

1.1 Elektronik Tablolama Programı - Microsoft Office Excel 2016

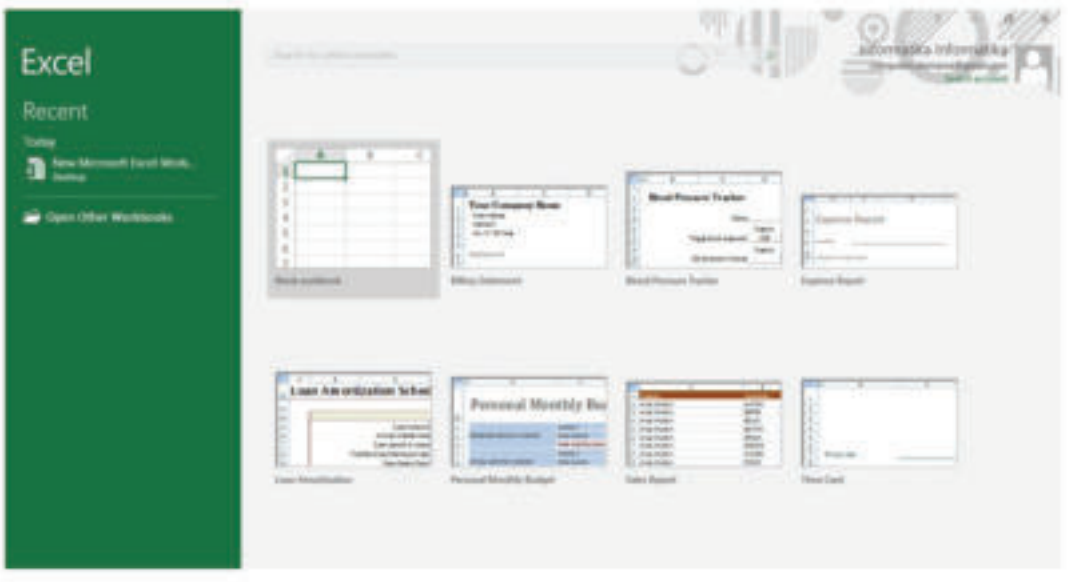
Ne öğreneceğiz?

Geçen okul yılında "Kelime işlem programı" konusunu okurken tablo terimini öğrendik. Tablo, sütunlar adı verilen dikey bölümler ve satırlar adı verilen yatay bölümler halinde organize edilmiş bir veri kümesidir. Her sütun ve satır kesişir ve kesişme noktası hücre olarak adlandırılır.

Microsoft Office Excel 2016 uygulaması, Microsoft Office 2016 ofis paketinin bir parçasıdır ve verilerin tablo şeklinde sunumunu, tabloları düzenlemeyi, elektronik tabloları gerçekleştirmeyi ve verilerin grafiksel gösterimini, gözden geçirme ve profesyonel raporlar oluşturmayı vb. Sağlar.

1.1.1 Programı başlatmak. Çalışma alanı öğeleri

Bir elektronik tablolama belgesi oluşturmak için, Microsoft Office Excel 2016 programını başlat menüsünden programı temsil eden uygun simgeyi tıkladıktan sonra aşağıdaki pencere açılır:



Şekil 1: Excel 2016'da yeni bir belge başlatmak

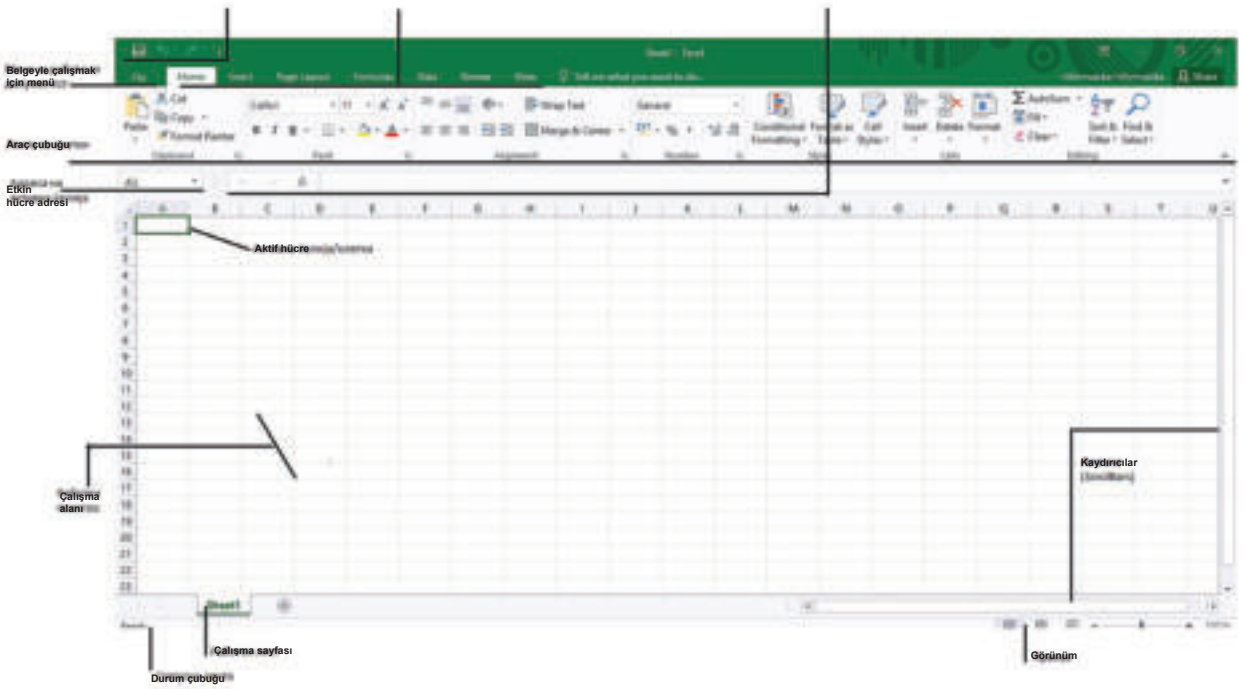
Bu pencere iki bölüme ayrılmıştır. Pencerenin sol tarafında, kullanıcının yakın zamanda açtığı ve bağlantı özelliğine sahip olduğu belgelerin bir listesi bulunur, yani belgenin adına tıklandığında bu belge hemen etkinleştirilir. Pencerenin sağ tarafında, kullanıcı bir Blank Workbook - açabilir, yani yeni veya boş bir çalışma kitabı açabilir ya da bir belgenin hazır bir şablonunu (Template) seçip, o şablon üzerinde çalışabilir.

Hatırlayalım!

Dosya nedir? Dosyaların başka adı nedir? Özellikleri nelerdir?

MS Excel 2016'da oluşturulan belgelere çalışma kitapları denir ve .xlsx uzantısını alır. Her çalışma kitabı çalışma sayfalarından (WorkSheet) oluşur. Her çalışma sayfası tablolardan oluşur. Tablonun dikey kısımları sütun olarak adlandırılır ve harflerle (A, B, C... ..Z, AA, AB, AC...) gösterilir ve yatay kısımlar satır olarak adlandırılır ve sayılarla (1,2,3...) gösterilir. Hücre, verilerin girildiği yerdir. Her hücrenin, sütunun harfinden ve içinde bulunduğu satır numarasından oluşan bir adı vardır. Örneğin: A1, B5, D16 vb. Her çalışma sayfası 256 sütun ve 65.536 satırdan oluşur.

Blank Workbook seçilmesi, Microsoft Office Excel 2016 çalışma penceresini etkinleştirir. Aşağıdaki resim, programın arayüzünü gösterir.



Şekil 2: MS Excel 2016 arayüzü

1. **Quick Access Toolbar - Hızlı Erişim Araç Çubuğu** - en sık kullanılan komutlara hızlı erişim sağlayan çubuktur. Kullanıcının ihtiyaçlarına göre değiştirilebilir.

2. **Menu Bar - Menü Çubuğu** - görevlerine göre komutların bir gruplandırılmasını, aslında uygun kart için bir komut çubuğunu oluşturan menü çubuğudur.

3. **Araç Çubuğu** - hangi menünün seçildiğine bağlı olarak görünür veya değişir.

4. **Etkin hücre adresi (Name Box)** - seçilen hücrenin adını görüntülemek için bir alandır, yani etkin hücreyi görüntüler.

5. **Formula Bar - Formül Çubuğu** - formülleri ve işlevleri girmek, verileri veya etkin hücrenin içeriğini görüntülemek için kullanılan bir çubuktur.

6. **Etkin hücre** - kullanıcının verileri girdiği hücredir.

7. **Sheet Tabs - Sayfa Sekmeleri** - çalışma sayfaları şeridi.

8. **Scroll Bar - Kaydırma Çubuğu** - çalışma penceresinde yatay ve dikey hareket etmek için kullanılan şeritlerdir.

9. **Status Bar** - Durum Çubuğu - pencerenin altında bulunur ve çoğu kullanıcıya geçerli sayfa hakkında bilgi veren birden çok seçeneği görüntüleyecek şekilde ayarlanabilir.

10. **Menü Dosyası** - Excel'de belgeyle çalışmak için temel komutları içerir.

1.1.2 Excel'de belgelerle çalışma

Excel'de belgelerle çalışma prosedürü, Microsoft Office paketinin diğer programlarıyla tamamen aynıdır, yani tümü File - Dosya menüsünden verilir. Belgelerle ilgili temel komutları ve işlemleri hatırlayalım:

Etkinlik / işlev	Etkinlik / işlevin anlamı
New	Yeni belge oluşturmak
Open	Var olan belgeyi açmak
Save	Belgeyi kaydetmek
Save As	Belgenin bir kopyasını farklı adla oluşturmak ya da farklı bir yerde oluşturmak
Print	Belgeyi yazdırmak
Share	Belgeyi başka kullanıcılarla paylaşmak ya da e-mail olarak gndermek
Export	Belge türünü değiştirmek ya da belgeyi .pdf veya .xps e dönüştürmek
Close	Belgeyi kapatmak

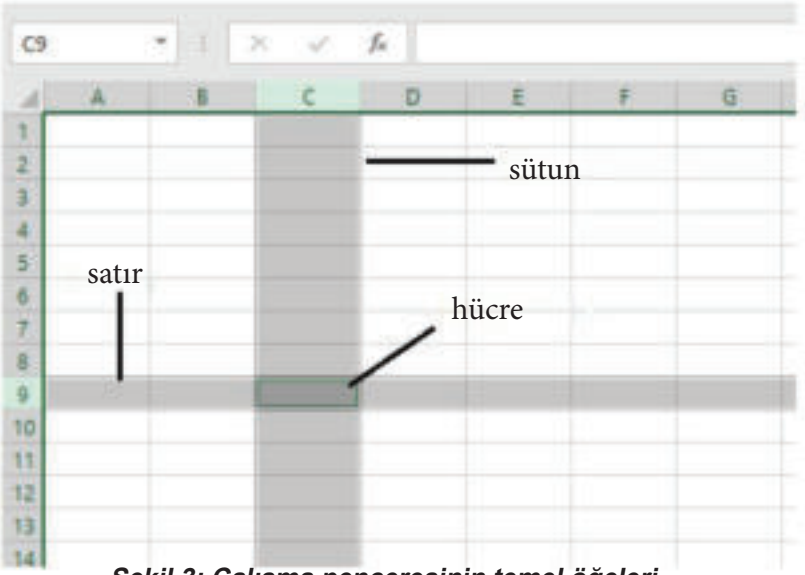
Tablo 1: Excel 2016'daki çalışma belgesiyle etkinlikler/ işlemler

Not!

Bu ders kitabındaki sütunların ve satırların kesişimi için hücre kelimesini kullanacağız.

1.1.3 Çalışma sayfasının temel öğeleriyle çalışma

Aşağıdaki pencerede, çalışma sayfasının temel öğelerini gösterir: tablo, satır, sütun ve hücre:



Şekil 3: Çalışma penceresinin temel öğeleri

Veri girmeye başlamak için önce bir hücre seçip içine verileri giriyoruz. Seçili ve veri girmeye hazır olan hücreye aktif hücre denir. Örneğin:

Ödev

Masaüstünüzde bir klasör oluşturun. Adınız ve soyadınız ile adlandırın. Bu klasörde yapacağınız pratik ödevleri kaydedeceksiniz.

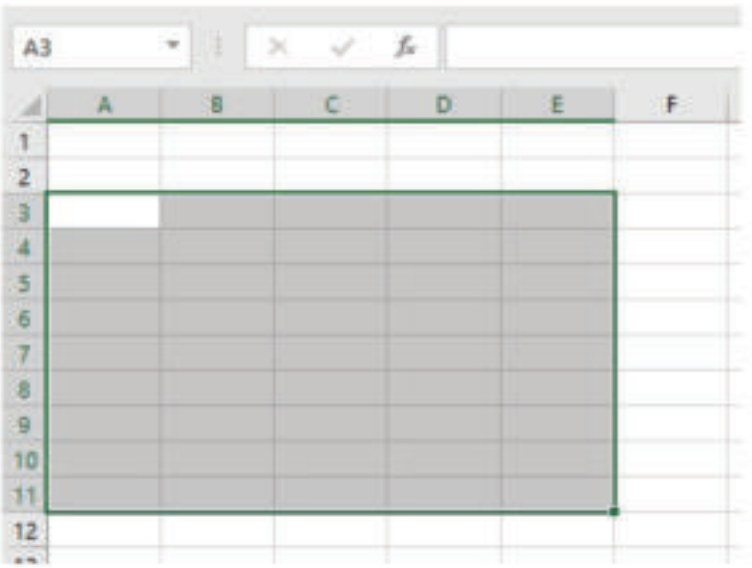
Yeni bir belge açın.
A1 hücresine tıklayın ve “egzersiz 1“ verisini yazın.
A2 hücresini tıklayın ve adınızı yazın.
B2 hücresini tıklayın ve sınıfınızı yazın.
Masaüstünde oluşturulan klasörde belgeyi “Ödev1.xlsx” adıyla kaydedin.

Çalışma sayfasındaki öğelerle çalışırken her zaman onları seçmeli veya işaretlemeliyiz. Seçme veya işaretleme öğeler üzerinde yazı tipi seçimi, yazı tipi boyutu, yazı tipi rengi, çerçeveler ayrıca kopyalama, kırpma, silme gibi işlemleri gerçekleştirebiliriz.

Bir hücre tıklanarak vurgulanır. İhtiyaca bağlı olarak, bazen aynı anda birden fazla hücre seçebiliriz. **Bitişik hücreleri** seçmek için ilk hücreye tıklıyoruz ondan sonra klavyede Shift tuşunu basılı tutarak son hücreye tıklıyoruz veya fareyle ilk hücreye tıklayıp ve

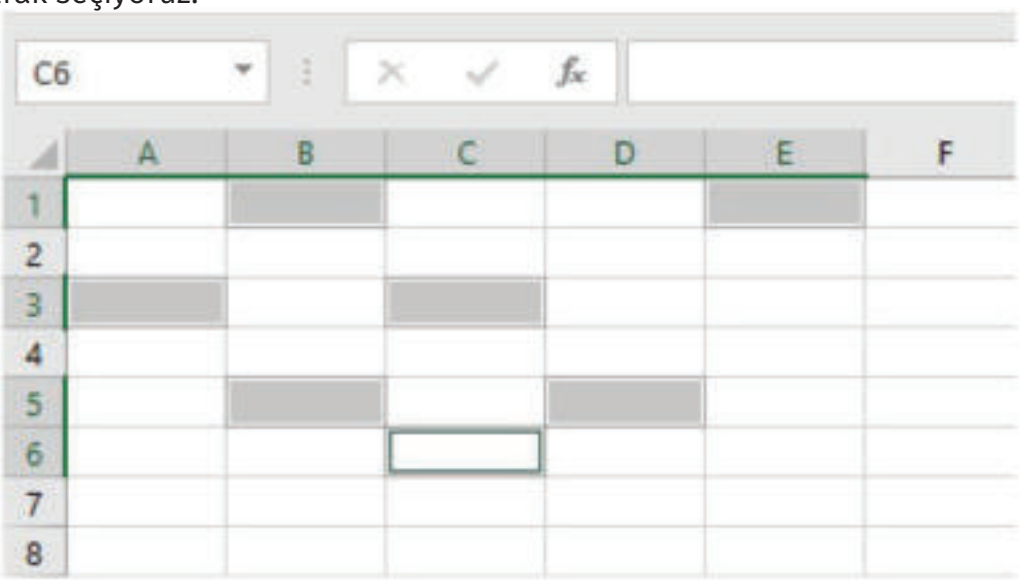
son hücreye kadar sürükleyip gereken bitişik hücreleri seçebiliriz. Bu şekilde hücre aralığını belirliyoruz.

Hücre aralığı, ardışık veya bitişik hücrelerin toplamıdır ve aşağıdaki gibi gösterilir, örneğin: A3: E11.



Şekil 4: Sıralı / bitişik hücreleri seçme

Ardışık olmayan veya **bitişik olmayan** hücreleri de seçme imkanımız bulunmaktadır. Bunları hücreye tıklayarak, klavyede Ctrl tuşunu basılı tutarak gereken diğer hücrelere tıklıyarak seçiyoruz.



Şekil 5: Bitişik olmayan hücreleri seçme

Çalışma belgesinde ilerlemek için, çalışma penceresinin sağ tarafında ve altında bulunan kaydırma çubuklarının (Scroll Bars) yanı sıra klavyedeki tuşları kullanacağız:

Klavye tuşu	Hareket
Enter ya da ↓	Bir hücre aşağı
↑	Bir hücre yukarı
←	Bir hücre sola
→	Bir hücre sağa
Ctrl + ←	Satırdaki ilk hücreye gider
Ctrl + →	Satırdaki son hücreye
Home	Satırın başlangıcı
End	Satırın sonu
Ctrl + Home	Tablonun başlangıcı
Ctrl + End	Tablonun sonu

Tablo 2: Çalışma penceresinde gezinme

Egzersiz 1

Ödev1 belgesinde, Sayfa1 çalışma sayfasındaki B4:F8 hücrelerini seçin ve onları maviye boyamak için Home menüden Fill Color aracını kullanın. Çalışma sayfasını "bitişik" olarak adlandırın. Belgede yapılan değişiklikleri kaydedin.

Aynı belgeye yeni bir çalışma sayfası ekleyin. Çalışma sayfasını bitişik olmayan olarak adlandırın. Hücreleri seçin: A2, C2, E2, G2, B3, D3, F3, A4, C4, E4, G4, B5, D5, F5, A6, C6, E6, G6, B7, D7, F7. Belgede yapılan değişiklikleri kaydedin.

Ezberle!

Microsoft Office Excel 2016, bir elektronik tablodaki verileri temsil etmek, hesaplamalar yapmak ve verileri grafik olarak görüntülemek için kullanılır. Excel 2016'da oluşturulan belgelere Çalışma Kitapları (Workbook) denir ve .xlsx uzantısına sahiptir. Çalışma kitapları veya belgeler çalışma sayfaları içerir (Sayfa1, Sayfa2...). Çalışma sayfaları elektronik tablolar içerir. Tablo, sütunlar adı verilen dikey bölümler ve satırlar adı ve-

rilen yatay bölümler halinde düzenlenmiş bir veri kümesidir. Her sütun ve satır kesişir ve kesişme noktası hücre olarak adlandırılır. Çalışan bir belgeyle ilgili en yaygın etkinlikler veya işlemler şunlardır: Yeni bir belge açmak, mevcut bir belgeyi açmak, bir belgeyi kaydetmek ve bir belgeyi yazdırmak.

Sorular

1. Neden Excel 2016 kullanmalı?
2. Elektronik tablolama programında oluşturulan belgelerin adı nedir?
3. Elektronik tablolama programı arayüzü öğeleri nelerdir?
4. Aktif belgedeki yatay bölümler elektronik tablolama programında nasıl ifade edilmiştir?
5. Bir elektronik tablolama programında bir sütun ve bir satırın kesişiminin adı nedir?

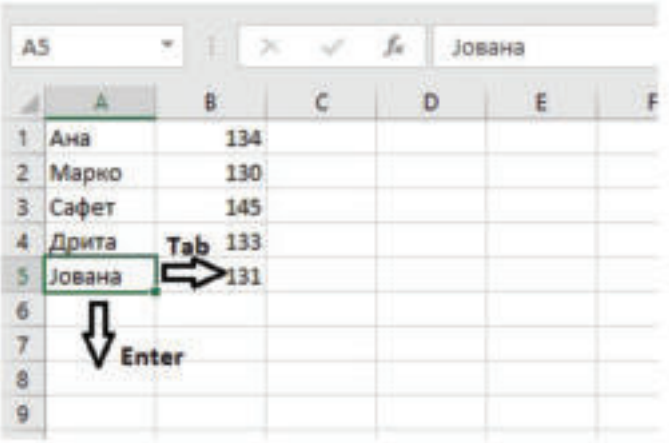
1.2 Tablodaki verilerle çalışma

Not!

Alfabetik veriler ve alfa sayısal veriler hücrenin solunda, sayısal veriler ise hücrenin sağında sıralanır. Elbette biçimlendirme araçlarıyla bunları istediğimiz gibi sıralayabiliriz.

Bir elektronik tablolama programında bir tablo oluşturmak için önce verileri girmemiz gerekir. Tablodaki veriler aktif hücreye girilir ve daha sonra önceki ünitede görüntülediğimiz çalışma penceresinden hareket etmek için klavye üzerindeki tuşları kullanarak girmeye devam ederiz.
Örneğin, öğrencilerin ve boylarının bir listesini oluşturmak için aşağıdaki adımları gerçekleştiriyoruz:

1. A1 hücresine tıklayın ve bir öğrenci adı girin;
2. A2 hücresine gitmek için klavyeden Enter tuşuna basın;
3. Bir adı yeniden girin ve on (10) ad girene kadar prosedürü tekrarlayın;
4. Sonadı girdiğinizde, sizi bir hücre sağa götürecek olan klavyedeki Tab tuşuna tıklayın. Orada bir sayı ile öğrencinin boyunu girin;
5. Çalışma belgesinde gezinmek için düğmelerin yardımıyla listedeki diğer öğrenciler için veri girmeye devam edin;
6. Belgeyi masaüstü klasöründe "**Ödev2.xlsx"** adaıyla kaydedin.



Şekil 1: Bir tabloya veri girme

Pratik görevden, aktif hücreye veri girdiğimizde, verilerin **Formül Çubuğundada** göründüğünü fark edebiliriz. Veri içeren herhangi bir hücreye tıkladığınızda, hücrenin içeriği Formül Çubuğu'nda görüntülenirken hücre adresi **Ad Kutusu'nda** görünür.

1.2.1 Veri türleri

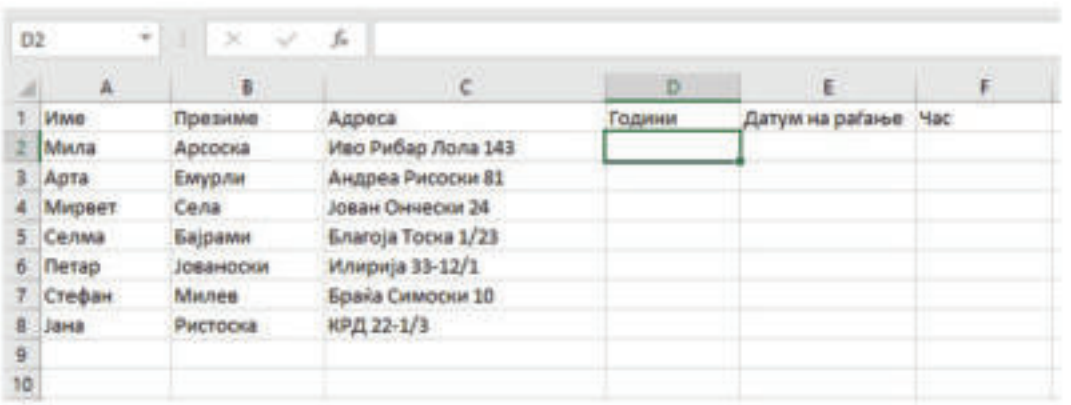
Tabloya girdiğimiz veriler **alfabetik**, **sayısal** ve **alfa sayısal** olabilir. Alfabetik veriler yalnızca harfleri içerir, yani metin, sayısal veriler sayı dizileridir, alfa-sayısal veriler ise harf / metin ve sayıların birleşimidir.

Alfabetik ve alfa sayısal veriler

Metin verilerini (alfabetik) ve bir metin ve sayı kombinasyonunu (alfanümerik) girmek için aşağıdaki ödevi gerçekleştireceğiz. "Ödev2.xlsx" belgesine, öğrenciler için temel veriler içerikli bir tablo oluşturacağımız yeni bir çalışma sayfası ekleyelim. Çalışma sayfasını "Veriler" olarak yeniden adlandırın.

1. A1, B1, C1, D1, E1, F1 ve G1 hücrelerinde başlık hücreleri oluşturacağız: Ad, soyad, adres, yıllar, doğum tarihi ve saat.

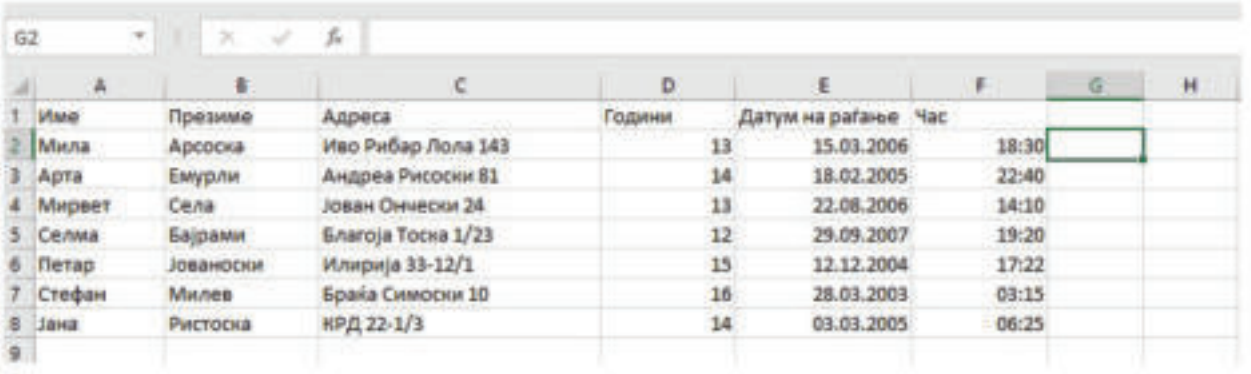
2. Çalışma belgesinde hareket tuşlarını kullanarak, verileri sütunlara girin: ad, soyad, adres.



Şekil 2: Alfabetik ve alfa sayısal verileri girme

Sayısal veriler

Sayısal veriler hakkında konuştuğumuzda, tam sayılar, pozitif ve negatif sayılar, tarih, saat, para birimi gibi farklı sayısal veri türleri olduğunu görebiliriz. “Veri” tablosunu dolduralım. Tüm verileri girdikten sonra, belgedeki değişiklikleri kaydedin.



	A	B	C	D	E	F	G	H
1	Име	Презиме	Адреса	Години	Датум на раѓање	Час		
2	Мила	Арсоска	Иво Рибар Лола 143	13	15.03.2006	18:30		
3	Арта	Емурли	Андреа Рисоски 81	14	18.02.2005	22:40		
4	Мирвет	Села	Јован Ончески 24	13	22.08.2006	14:10		
5	Селма	Бајрами	Благоја Тоска 1/23	12	29.09.2007	19:20		
6	Петар	Јованоски	Илирија 33-12/1	15	12.12.2004	17:22		
7	Стефан	Милев	Браќа Симоски 10	16	28.03.2003	03:15		
8	Јана	Ристоска	КРД 22-1/3	14	03.03.2005	06:25		
9								

Şekil 3: Sayısal verileri girme

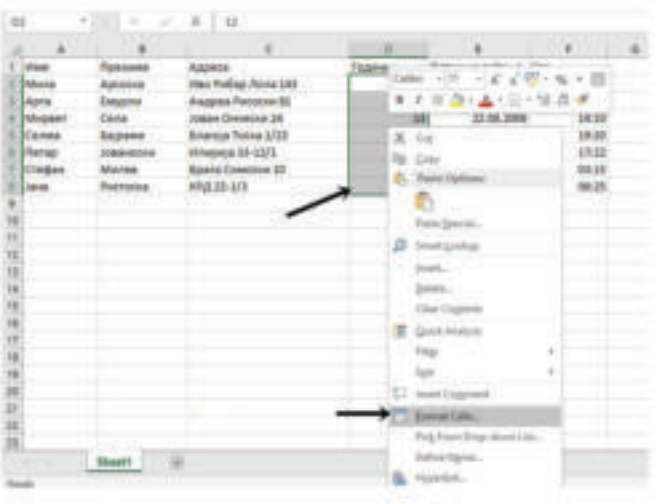
Bu nedenle, bir tarih girerken günü bir nokta (.) ile aydan ayırdığımızı fark edebiliriz. Ayrıca ayı yıldan bir noktayla ayırdık. Bir tarih girerken, eğik çizgi (/) veya yalnızca eğik çizgi (-) kullanabiliriz. Excel daha sonra bu verileri tarih olarak tanır. Bir saat girerken, iki nokta işareti (:) ile saatleri dakikalardan ve saniyelerden ayırıyoruz.

1.2.2 Sayısal verilerin formatı

Tablolarda verileri girerken farklı bir veri formatı seçebiliriz. Veri formatının seçimi şu şekilde gerçekleştirilir:

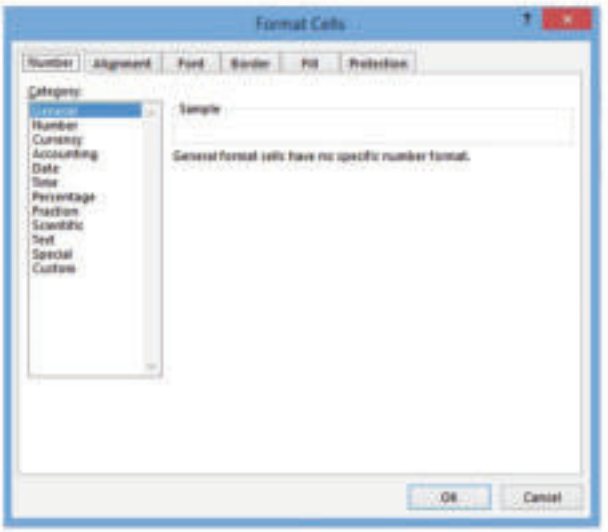
1. formatını değiştirmek istediğimiz verileri seçiyoruz;
2. Seçili hücrelere sağ tıklayın ve **Format Cells** - **Hücreleri Biçimlendir** seçeneğini seçin.

Aşağıdaki pencere açılır:



Şekil 4: Format Cells - Hücreleri Biçimlendir seçeneği

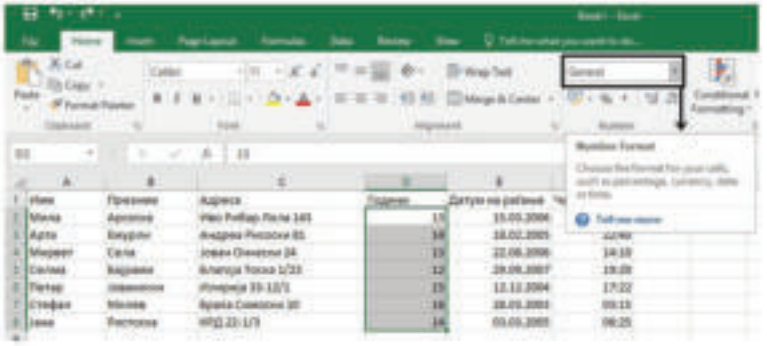
Hücreleri Biçimlendir seçeneğini seçtikten sonra, **Number** - **Sayı** sekmesini seçiyoruz ve ardından kategorileri seçtiğimiz bir pencere açılır:



Şekil 5: Hücreleri Biçimlendir penceresi

1. **General** – genel biçim.
2. **Number** – tam sayılar ve ondalık sayılar, pozitif ve negatif sayılar.
3. **Currency** – para birimi biçimi: denar, euro, dolar, lira, pound, vb.
4. **Accounting** – değerin solundaki para birimi işaretlerini ve sıfırda (0) bir kısa çizgi (-) görüntüler.
5. **Date** – tarih biçimleri.
6. **Time** – zaman biçimleri.
7. **Percentage** – yüzde biçimini görüntüler.
8. **Fraction** – kesir biçimleri.
9. **Scientific** – bir sayının üstel biçimde görüntülenmesi.
10. **Text** – metin dizeleri.
11. **Special** – özel biçimler.
12. **Custom** – ek veri formatlarının seçimi.

Farklı bir sayısal veri biçimi seçmek için gösterilen prosedürün dışında, aşağıdaki görüntüde gösterildiği gibi Home – Ana menü ve Number Format - Sayı Biçimi seçeneğini de kullanabiliriz:



Şekil 6: Ana menü üzerinden veri formatını seçme

Tarih biçimini gg.mmm.yy ve saat formatını seçelim, saat ve dakikaların ayrıca sabah / öğleden sonra seçeneğini de belli edelim. Belgeye yapılan değişiklikleri kaydedin.

Ezberle!

Elektronik tablolama programında bir elektronik tablo oluştururken, veriler aktif hücreye girilir. Farklı veri türleri vardır: alfabetik, sayısal ve alfa numerik. Sayısal verilerin farklı biçimleri olabilir: tamsayı, ondalık sayı, pozitif ve negatif sayı, tarih, saat, para birimi vb. V eri formatlarını fare ile sağ tıklarken Format Cells - Hücreleri Biçimlendir seçeneğinden veya Home-Ana menüden Number Format - Sayı Biçimi seçeneğinden seçiyoruz.

Sorular

1. Veriler hangi hücreye girilir?
2. Hücreye veri girdikten sonra hangi düğme veya düğmeler tıklanır?
3. Farklı sayı biçimlerini yaz!

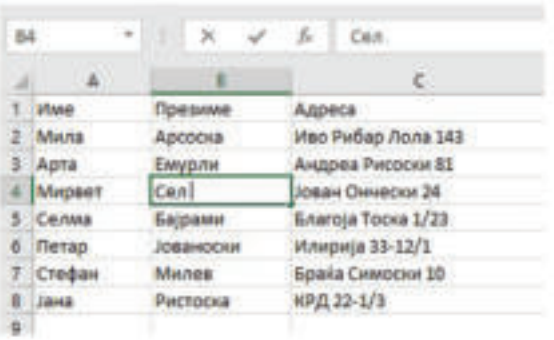
Üstel biçim, bir sayının bir kısmını E + n ile değiştirmek anlamına gelir; burada E (üs), 10 un (üs) derecedeki değeri önceki sayısıyla çarpılır. Örneğin, 2 ondalık bilimsel biçim 12345678901, 1.23E + 10 olarak görüntüler; bu, 10 un 10 derece değerini 1.23 ile çarpımıdır.

1.3 Tabloyu düzenleme

İlk bakışta elektronik tablolama programında bir tablo oluşturduğumuzda, teknik bir hata yapıp yapmadığımızı, yanlış veri girip girmediğimizi, tablodaki bir satırın veya tüm sütunun eksik olup olmadığını kontrol ettiğimiz sürece her şey yolunda görünüyor. Tabi ki tablonun tamamını silip yeniden oluşturmaya başlamayacağız ancak yapılan hataları düzelteceğiz. Değişiklik yapmak için "**Örnek2.xlsx**" i açacağız.

1.3.1 Girilen verileri değiştirme ve düzenleme

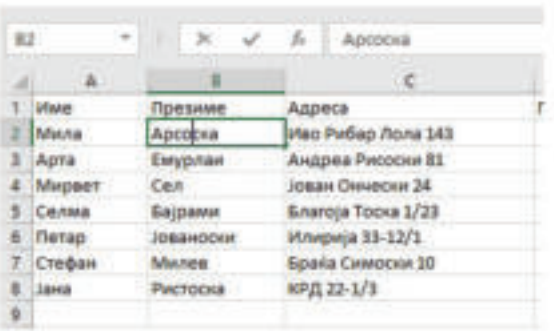
B4 hücresindeki verileri değiştirmek istiyorsak, yani mevcut olanın yerine yeni verileri girmek istiyorsak, üzerine tıklar ve yazmaya başlarız. O anda durum çubuğuna **Ready** yazılacaktır, bu da bizim yeni veri girme modundayız demektir. Örneğin: Sela yerine Selim yazacağız.



	A	B	C
1	Име	Презиме	Адреса
2	Мила	Арсоска	Иво Рибар Лола 143
3	Арта	Емурли	Андреа Рисоски 81
4	Мирвет	Сел	Јован Ончески 24
5	Селма	Бајрами	Благоја Тоска 1/23
6	Петар	Јованоски	Илирија 33-12/1
7	Стефан	Милев	Браќа Симоски 10
8	Јана	Ристоска	КРД 22-1/3
9			

Şekil 1: Bir hücredeki verileri değiştirme

B2 hücresine bir karakter, yani soyadına bir harf eklememiz gerekirse:
Fare ile B2 hücresine çift tıklıyoruz, imleci bir harf girmemiz gereken yere konumlandırıyoruz Durum çubuğuna **Edit** - **Düzenle** yazılacaktır, bu, veri değiştirme modunda olduğumuz anlamına gelir.
Klavyeden gereken harfi yazıyoruz.



	A	B	C
1	Име	Презиме	Адреса
2	Мила	Арсоска	Иво Рибар Лола 143
3	Арта	Емурлаи	Андреа Рисоски 81
4	Мирвет	Сел	Јован Ончески 24
5	Селма	Бајрами	Благоја Тоска 1/23
6	Петар	Јованоски	Илирија 33-12/1
7	Стефан	Милев	Браќа Симоски 10
8	Јана	Ристоска	КРД 22-1/3
9			

Şekil 2: "c" harfini girerken bir karakter / harf ekliyoruz.

Not!

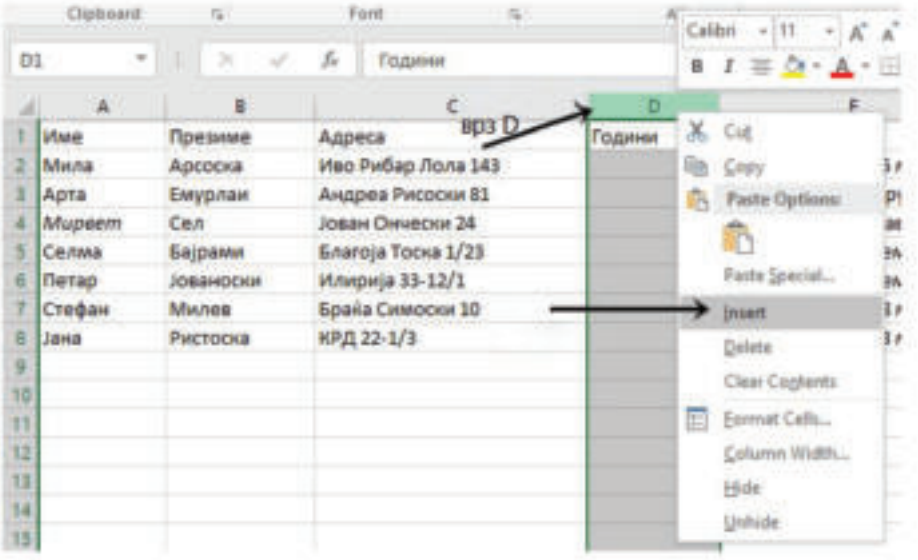
Birden fazla sütun veya satırı hızlı bir şekilde eklemek için, birden çok satır / sütun seçin, üzerlerine sağ tıklayın ve Insert - Ekle'yi seçin.

Hücrelerdeki veriyi veya verileri silmek istediğimizde, verileri seçip ardından klavyedeki **Delete** tuşuna tıklayarak silebiliriz.

1.3.2 Sütun ve satırlar ile işлемler

Oluşturulan tabloda, kullanıcının ihtiyacına göре sütun ve satır ekleyip silebiliriz. **Yeni bir sütun ekleme** prosedürü aşağıdaki gibidir:

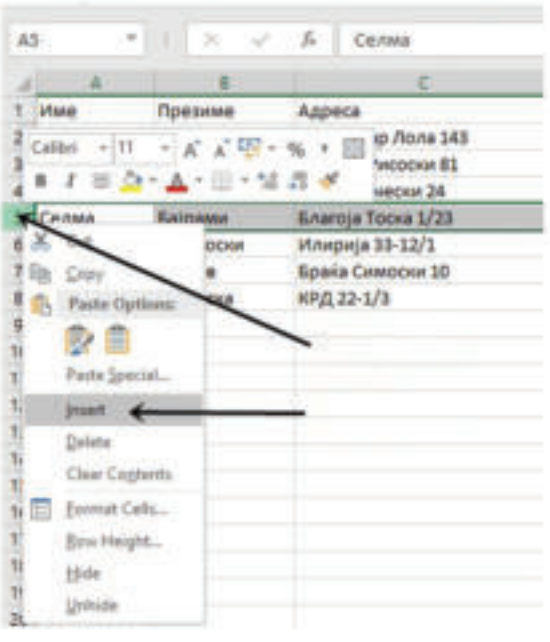
1. önüne yeni bir sütun eklemek istediğimiz sütunu işaretleyin;
2. işaretli sütunun harfine sağ tıklayın;
3. **Insert - Ekle** seçeneğini seçin.



Şekil 3: Bir tabloya sütun ekleme

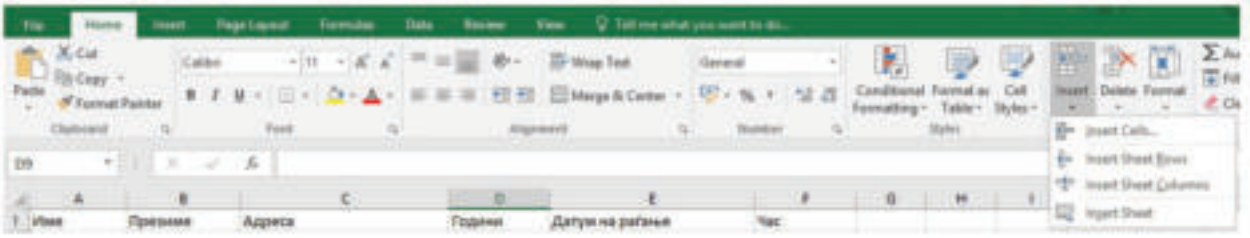
Tabloya bir satır eklemek için işlem aynıdır:

1. önüне yeni bir satır eklemek istediğimiz sütunu işaretleyin;
2. işaretli satırın numarasına sağ tıklayın;
3. **Insert - Ekle** seçeneğini seçin.



Şekil 4: Tabloya satır ekleme

Bir tabloya sütun ve satır eklemek, **Home** menü aracılığıyla ve bir sütun ve satır eklemek için uygun araçları seçerek de yapılabilir:



Şekil 5: Menü üzerinden sütun ve satır ekleme

Deneyin!

Bir tabloya sütun ve satır ekleme işlemini öğrendiğimizden, Insert - Ekle seçeneği yerine Delete - Sil'i seçmemiz gerektiğini biliyorsak bir sütun veya satırı silmeye çalışalım.

1.3.3 Sütun ve satırların boyutlarını değiştirme

Hücrelere veri girerken, genellikle **sütunları ve satırları yeniden boyutlandırmamız gerekir.** Sütunun ve satırların boyutlarını, istediğimiz şekilde değiştirme işlemi aşağıdaki şekilde gösterilmektedir:

Şekil 6: Bir sütun ve bir satırın boyutunu değiştirme

Resimden ilk önce işaretçiyi sütunun harfleri arasına ya da satırın numaraları arasına yerleştirdiğimizde, imlecin değiştiğini fark ediyoruz, imleç değiştiğinde fareyi tıklayıp sürüklüyoruz. Satır boyutunu değiştirirken, imleç 90 derece değişir işlem ise aynıdır.

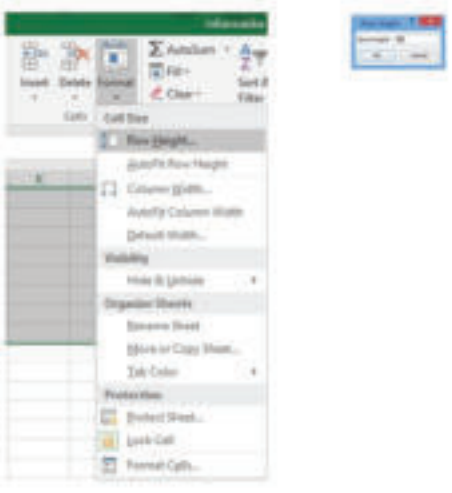
Not!

Sütun/satır işaretinden sonra satıra ya da sütuna çift tıklayarak sütunların ya da satırların boyutlarını daha hızlı değiştirebilirsiniz.

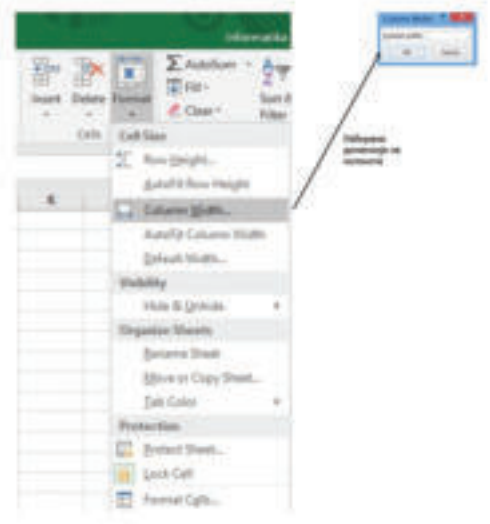
Ödev

"Ödev2.xlsx" belgesini açın ve doğum tarihi sütunu ve bu sütunun başlığının boyutlarını değiştirin. Ardından, tüm sütunlar ve satırlar için aynı boyutları elde etmek için Home menüdeki Format - Biçim seçeneği aracılığıyla sütun ve satırları yeniden boyutlandırma işlemini gerçekleştirin.

Kullanıcı belirli bir değer için sütunları ve satırları artırmak veya azaltmak istediğinde, Home menüsünden Format - Biçim aracını seçeriz:



Şekil 7: Home-Ana menüden Format-Biçim seçeneği ile satır boyutlarını değiştirme



Şekil 8: Home-Ana menüden Format-Biçim seçeneği ile sütun boyutunu değiştirme

Ezberle!

Bir belgeyi düzenlemek düzeltme işlemidir. Bunu yapmak için, hücreye tıklayarak ve hemen yazmaya başlayarak veya çift tıklayıp imleci düzeltme yapmak istediğimiz metin üzerine getirerek verileri değiştirebiliriz. Home menü aracılığıyla bir sütun ve bir satır ekleyebilir, bir sütun ve bir satırı silebilir ve boyutlarını değiştirebiliriz.

Sorular

1. Bir hücredeki verileri değiştirme işlemi nedir?
2. Sütunların ve satırların boyutları nasıl değiştirilir: İşlemleri yazınız!

1.4 Tabloyu biçimlendirme

Hatırlayalım

Biçimlendirmenin ne olduğunu hatırla! Biçimlendirme için en yaygın olarak hangi araçlar kullanılır? Biçimlendirirken en çok nelere dikkat etmen gerektiğini açıklayabilir misin?

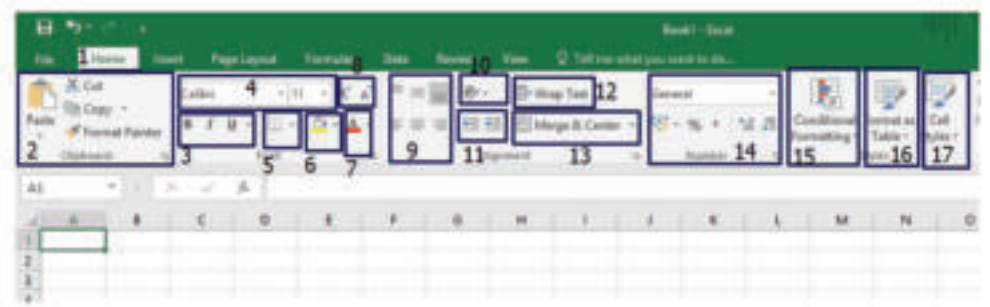
Hücrelerin, sütunların ve satırların görünümü, tablo düzeninin biçimlendirilerek sonlandırılır.

Veri biçimlendirme, aslında elektronik tablonun görünümünü geliştiren efektlerin eklenmesidir, böylece içindeki verilerin daha güzel görünmesini sağlar.

Hücre biçimlendirmesi:

- yazı tipi nitelliklerini değiştirme;
- hücrelerde veri hizalaması;
- metnin farklı açıdan görüntülenmesi;
- hücreleri birleştirme ve bölme;
- bir veya daha fazla hücreye çerçeve ve çizgi eklemek;
- hücreye renk ve efektler ekleme

Herhangi bir işlem yapılmadan önce hücreler seçilmelidir. Biçimlendirme araçlarının çoğu **Home**-Ana menüdedir:



Şekil 1: Home-Ana menüde biçimlendirme araçları

1. Home-Ana menü.
2. Kopyalama, kırpma, yapıştırma ve biçimlendirme araçları.
3. Stiller: kalın, eğik ve altı çizili.
4. Yazı tipi seçimi ve yazı tipi boyutu.
5. Kenarların seçimi.
6. Hücre rengi seçimi.
7. Yazı tipi rengi seçimi.
8. Yazı büyüklüğünü artırma / azaltma araçları.
9. Yazıları hizalama.
10. Metnin yönü.
11. Metnin ayrıntılı özellikleri.

12. Hücreler birleştirme ve bir merkez düzenleme.
13. Veri türü seçimi.
14. Koşullu biçimlendirme.
15. Tablo biçimlendirme.
16. Hücre stilleri.

Home-Ana menü harcında tablonun biçimlendirilmesi, tablodan hücreler seçildikten sonra **Format Cells** - Hücreler Biçimlendir seçeneğiyle sağlanabilir:

1. Veri formatı seçimi
2. Veri hizalama
3. Yazı tipi seçimi
4. Kenar seçimi ve kenar seçenekleri
5. Hücreleri renkle doldurulması
6. Çalışma sayfası koruması

Şekil 2: Format Cells - Hücreleri Biçimlendir menüsü aracılığıyla verileri biçimlendirme

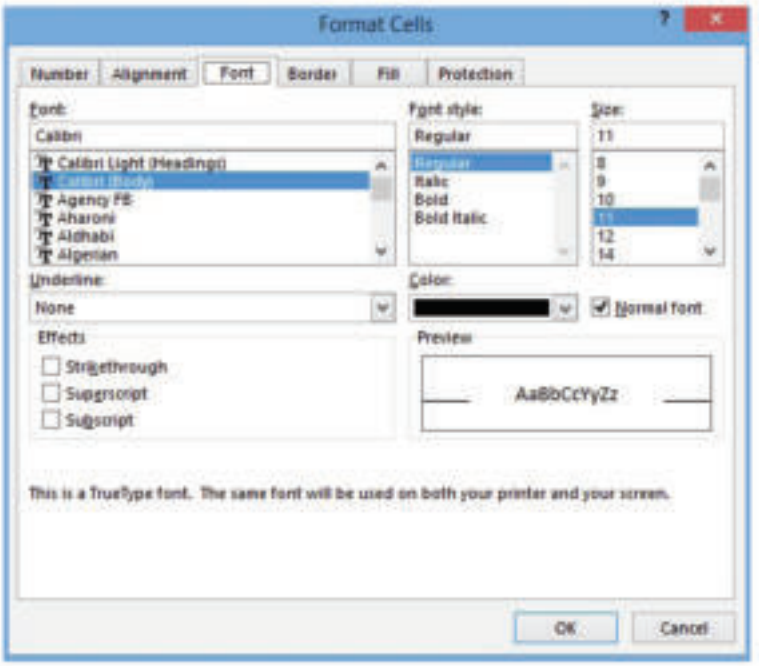
Alşgment-Hizalama sekmesi aracılığıyla, kullanıcı hücelerin içeriğini sıralayabilir, yani hücrelerdeki verileri sıralayabilir, böylece Metin Hizalamasında verilerin yatay ve dikey olarak sıralanmasının yanı sıra hücredeki metnin yönü ayarlanabilir.
Aşağıdaki seçenekler seçilerek ek düzenlemeler yapılır:

Wrap Text - Metni Kaydır - birden çok satırda düzenlenecek metni görüntüler.

Shrink To Fit- hücre boyutlarını değiştirmeden metni hücreye sığacak şekilde küçültür.

Merge Cells - Hücreleri Birleştir - seçili hücreleri birleştirir.

Font - Yazı Tipi sekmesini seçerek yazı tipi biçimini, stillerini, boyutunu ve yazı tipiyle ilgili değişiklik efektleri seçin:

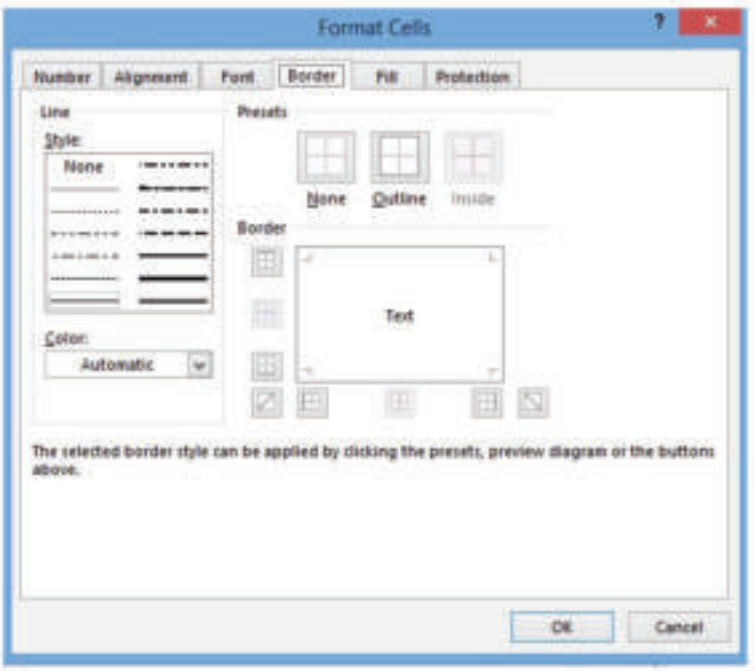


Şekil 3: Yazı tipi araçlarının seçimi

Not!

Home - Ana menüden yazı tipi seçeneklerini, yazı tipi stilini ve yazı tipi boyutunu da seçebilirsiniz.

Border - Kenarlık seçeneğinde, tablo hücrelerinin kenarlarının stilini, kalınlığını ve rengini seçilir:

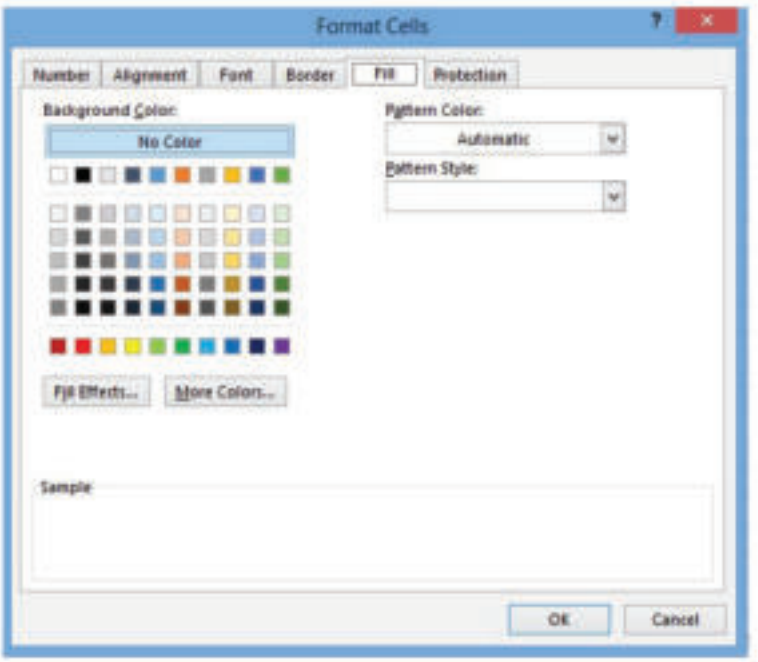


Şekil 3: Yazı tipi araçlarının seçimi

Not!

Home - Ana menü tablo kenarlarını düzenlemek için hızlı ve kolay erişim olarak araçlar içerir.

Fill - Doldur seçeneği, hücreleri istediğimiz renkle doldurmak için kullanılır:



Şekil 5: Hücre zemini renk seçimi

Ödev:

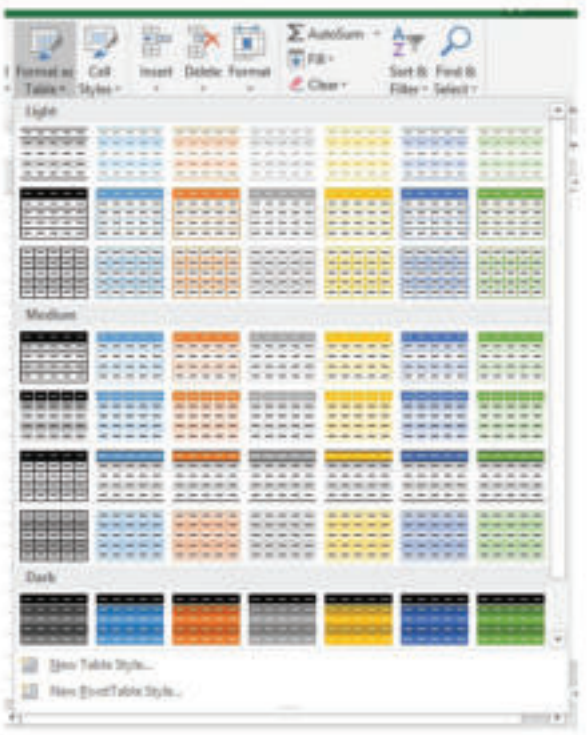
Yeni bir çalışma sayfası açın ve ilk çalışma sayfasında (Sheet1) bir ders planı oluşturun:

- A1, B1, C1, D1, E1 hücrelerinde hafta günlerinin adlarını girin;
- başlık hücrelerinin altındaki hücrelere, ders programına göre dersleri girin;
- Pazartesi sütunundan önce yeni bir sütun ekleyin ve ders saatlerinin sırasını yazın;
- başlık hücrelerinin üstüne yeni bir satır ekleyin;
- ilk altı hücreyi birleştirin ve "Haftalık Ders Programı" başlığını yazın;
- Gün adlarının yazıldığı başlık hücrelerine, Calibri yazı tipi, boyutu on dört (14), kalın ve italik stiller, harflerin rengi koyu mavi ve hücrelerin arka plan rengi açık mavi yapınız;
- tüm tabloya kenarlıklar ekleyin;
- sütunların / satırların boyutlarını gerektiği gibi ayarlayın;
- belgeyi masaüstü klasörüne kaydedin.

Hazır hücre biçimlendirmesi, şekilde gösterildiği gibi **Home**-Ana menüden **Cells Styles** Hücre Stilleri aracı seçilerek de yapılabilir. Kullanıcının karşısına sunulan seçeneklerden seçtiği stili uygulamak için hücrelerin önceden seçilmesi gerekir:

Şekil 6: Hücre stillerinin seçimi

Hazır hücre formatlarına ek olarak, kullanıcı aşağıdaki görüntüde gösterildiği gibi, Home-Ana menüden Format As Table - Tablo Olarak Biçimlendir aracını seçerek hazır bir tablo formatı seçebilir:



Şekil 6: Hücre stillerinin seçimi

Not!

Tablonun standart biçimlendirmesine ek olarak, özel biçimlendirme, yani koşullu biçimlendirme vardır. Buna Home - Ana menü ve Conditional Formating - Koşullu Biçimlendirme seçeneğinden erişiyoruz. Bu biçimlendirme yöntemini oluşturulan tablolardan birinde kullanmayı deneyin.

Ödev

Yeni bir çalışma sayfası ekleyin ve derslerin başlangıcı ve bitişi ve dersler arasındaki molaların süresi için verileri gireceğiniz bir tablo oluşturun. **Home - Ana** menüden Cells Styles - Hücre Stilleri uygulayarak veya **Home** menüsünden **Format Table - Tabloyu Biçimlendir** seçeneğiyle hazır bir tablo formatı seçerek tabloyu formatlayın.

Ezberle!

Bir tabloyu biçimlendirmek, tabloya, hücredeki verilerin şeklini, biçimini, yönünü ve hizalamasını değiştirmeyi içeren efektler eklemek anlamına gelir. Bir tabloyu biçimlendirirken, Ana menüdeki veya **Format Cells** - **Hücreleri Biçimlendir** komutu aracılığıyla araçlar en sık kullanılır. Hücrelerin ve tabloların hazır formları da vardır ve bunları **Home - Ana** menüden ve **Cells Styles - Hücre Stilleri** veya **Format As Table** - **Tablo Olarak Biçimlendir** komutlarından seçiyoruz.

Sorular

1. Biçimlendirme nedir?
2. Bir elektronik tabloyu biçimlendirme ve düzenleme arasındaki fark nedir?
3. Tablo formatlama için en yaygın olarak hangi araçlar kullanılır?
4. Hücrelerdeki metnin hizalanması ve yönü nasıl düzenlenir?
5. Format Cells - Hücreleri Biçimlendir komutu ile hangi seçenekleri seçebiliriz? Onlara başka bir yoldan erişebilir miyiz? Açıkla!
6. Bir metni bölme işlemi nedir?
7. Merge Cells - Hücreleri Birleştir komutu neden yararlıdır?
8. Home - Ana menüden Cells Styles - Hücre Stilleri seçeneğini niçin kullanılır?
9. Hazır tablo formatını kullanıyor musunuz? Seçim işlemi nedir?

1.5 Tablodaki verilerin otomatik olarak doldurulması

Hatırlayalım

Bilgisayar verişlemenin ne olduğunu hatırlıyor musunuz? Bilgisayar verişlemenin avantajları nelerdir? "Otomatik" işleme terimini açıklayabilir misiniz?

Bir elektronik tablodaki verileri otomatik olarak doldurmak, çok kullanışlı bir **Excel** aracıdır. Bu araç, verileri manuel olarak girmek yerine hücereleri doldurmak için Autofill - Otomatik Doldurma işlevini kullanabiliriz. Otomatik doldurma, bir elektronik tablolama uygulamasının ana işlevlerinden biridir, bu nedenle gerekli veriler elektronik tablolamaya manuelden çok daha kısa sürede girilir.

Aslında, **Microsoft Excelin Otomatik Doldurma** özelliği, elektronik tabloları daha verimli bir şekilde oluşturmamıza olanak tanıyarak, hücereleri yalnızca basit sayısal değerlerle değil, diğer veri veya formüllerle de veri kümeleriyle hızlı bir şekilde doldurmamıza olanak tanır. Ayrıca kullanıcının her seferinde aynı değerlerle yeni bir giriş girerek zaman kaybetmemesini sağlayacak bireysel listeler oluşturmak da mümkündür.

Otomatik hücre doldurmanın işlevi ve uygulanabilirliği hakkında bilgi sahibi olmadan önce, farklı işaretçi biçimlerinin anlamlarını anlayalım.

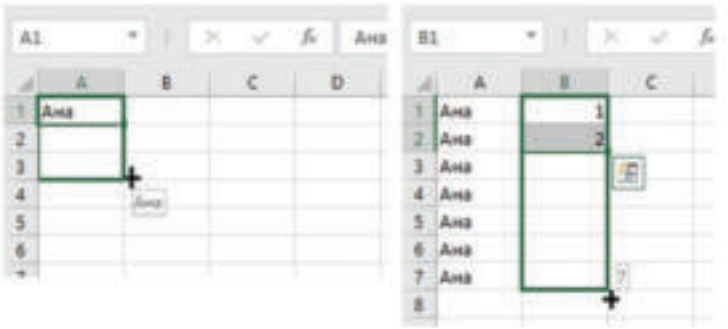
Işaretçi	Anlam
+	Otomatik hücre doldurma
	Hücre içeriğini kopyalama
	Hücre içeriğini hesaplama
	Bir sütunun boyutunu değiştirme
	Bir satırın boyutunu değiştirme
	Sütunu işaretle
	Bir satırı işaretle
	Bir hücreyi işaretle

Tablo 1: İşaretçi şekilleri

Hücreleri otomatik olarak doldurabilmek için şu "+" işaretçiyi kullanacağız. Hücerelerin otomatik olarak doldurulması işlemi aşağıdaki gibidir:

1. Aktif hücre olarak işaretleyeceğimiz ve verileri girileceğimiz simgeye tıklıyoruz;

2. Verileri girdikten sonra klavyeden Enter tuşuna basıyoruz;
3. Fare ile verinin bulunduğu hücrenin sağ alt köşesine geliyoruz;
4. Otomatik doldurma için bir işaretçi elde ederiz;
5. Otomatik doldurmayı gerçekleştirmek istediğimiz hücreye tıklayıp sürüklüyoruz.



Şekil 1: Otomatik hücre doldurma

Not!

Listeler elektronik tablolama programında oluşturulabilir, örneğin: haftanın günleri, yılın ayları vb. Listeler oluşturma işlemi File - Dosya, Options -Seçenekler, Advanced - Gelişmiş menüsü ve Özel Listeyi Düzenle'yi seçeriz. Hazır bir liste oluşturmaya çalışın!

Ödev

1. Excel 2016'da yeni bir çalışma belgesi açın;
2. çalışma sayfası 1'de (Sayfa 1), sütun A'da, bir dizi oluşturmak için ilk (2) sayısından yirmi ikinci (22) ye kadar çift sayıları otomatik hücre doldurma kullanarak yaz;
3. aynı çalışma sayfasındaki B sütununda, üç (3) sayısından 23 yirmi üçe (23) kadar tek sayılar dizisini oluştur;
4. C sütununda Ana adının bir dizisini, D sütununda Ana Maria adının bir dizisini;
5. sıfırdan (0) başlayarak beş (5) değerine dönüşecek bir dizi oluşturmak için E sütununda;
6. F sütununa 100, 99, 98 ..... 89 sırasını doldurun.

Ezberle!

Otomatik hücre doldurma, verileri girmemize, diziler oluşturmamıza ve hücreleri daha kolay ve daha hızlı bir şekilde doldurmamıza izin verdiği için elektronik tablolama programının birçok avantajından biridir. Otomatik hücre doldurma gerçekleştirmek için + işaretçiyi kullanıyoruz. Otomatik Doldurma yalnızca metin içeren verilerde değil, aynı zamanda sayılar, çok sayıda verinin çeşitli biçimleri, formüller ve işlevler üzerinde de yapılabilir. Elektronik tablolama programında, hücreleri otomatik olarak doldurmak için kullanılacak özel listeler oluşturulabilir.

1.6 Elektronik tablolama programındaki formüller ve işlevler

Tanımda da belirttiğimiz gibi Excel programı, bize bir elektronik tablo ve verilere genel bir bakış sunmanın yanı sıra, hesaplamalar yapmamıza da olanak tanır. Bu nedenle hesap tablosu programında formüllerin ve fonksiyonların uygulanmasındaki temel amaç hızlı ve doğru hesaplamalar yapmaktır.

1.6.1 Formüller

Elektronik tablolama programlarındaki formüller, verilerden ve aralarında çeşitli matematiksel operatörlerden oluşturulur. Excel 2016, aşağıdaki tabloda gösterilen standart formül karakterlerini kullanır:

Simge	Matematik işlemi
+	Toplama
-	Çıkarma
*	Çarpma
/	Bölme
^	Derece/Üssü

Tablo 1: Hesaplamaları yapmak için matematik operatörleri

Not!

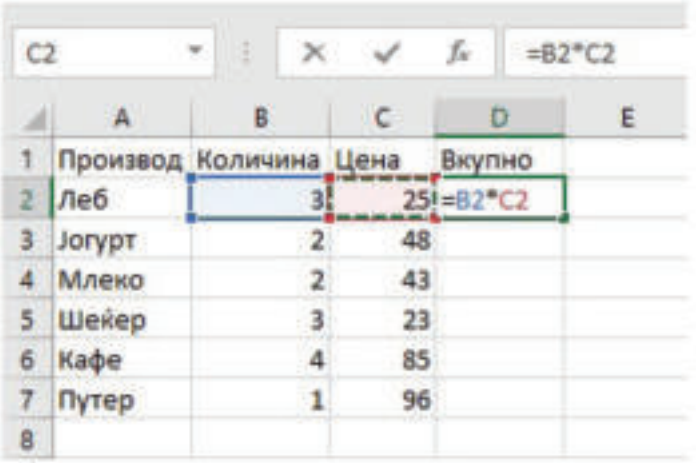
Doğru (True) veya yanlış (False) değer döndüren <,>, =, <=,> =, <> gibi mantık operatörlerde kullanılabilir. Formüller oluştururken parantezler de kullanılabilir.

Formüller ve işlevler tablo hücresine girilir ve her zaman "=" işaretiyle başlar.

Örneğin:

Formüller kullanarak süpermarkette ne kadar ödeyeceğimizi hesaplamak için verileri girdikten sonra şu işlemleri yapıyoruz:

1. D2 hücresine tıklayıp "=" işaretini yazıyoruz;
2. Miktarın bulunduğu hücrenin adresini giriyoruz;
3. çarpım işaretini ve ürünün fiyatının yazıldığı hücrenin adresini ekliyoruz;
4. klavyeden Enter tuşuna basıyoruz.



Şekil 1: Bir formül kullanarak hesaplama

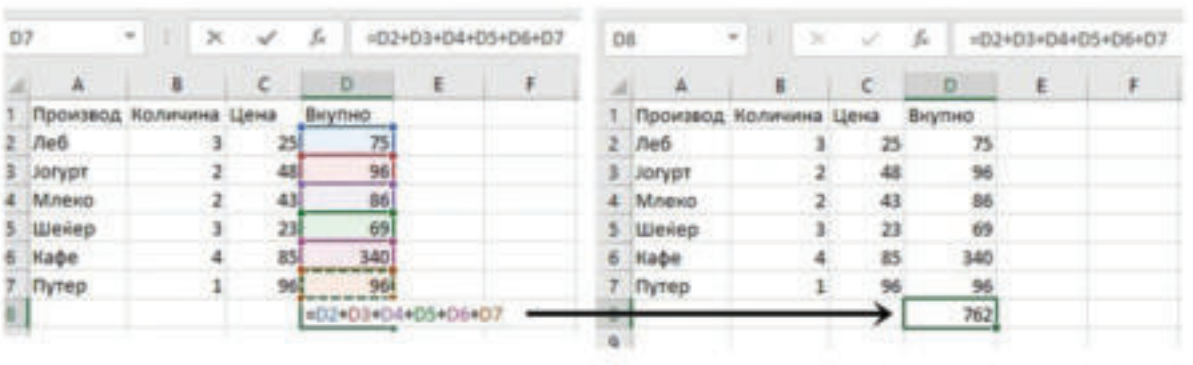
Not!

Hücre adreslerini girmenin dışında, hücre adresini tıklayarak formüle ekleyebiliriz.

Ödevden, etkin hücreye bir formül girerken, etkin hücrede klavyedeki **Enter** tuşuna basmanın sonucu görüntülediğini ve formül çubuğunda (**Formula Bar**) sonucu almak için kullandığımız formülü göründüğünü fark ediyoruz. Diğer ürünlerin hesaplanması için hücrelerin otomatik doldurma seçeneğinden yararlanabiliriz.

Ödenecek toplam tutarı hesaplayalım!

1. D8'e tıklayın ve "=" işaretini girin.
2. D2: D7'nin hücre adreslerini aralarında toplama işareti ile giriyoruz.
3. Klavyede Enter'a tıklayın ve toplam sonucu alın.



Şekil 2: Bir formül kullanarak toplam miktarın hesaplanması

Not!

Formüller oluştururken, verilerin değerini değiştirirken sonucu otomatik olarak değiştirmek için hücrelerdeki değerlerle değil adresleriyle çalışacağız.

Ödev

Verilen tabloyu doldurunuz:

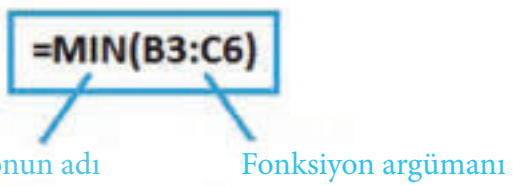
D2

	A	B	C	D
1	X	Y	z=x*2-2*y	s=(x+y)/4
2	0	3		
3	1	6		
4	2	9		
5	3	12		
6	4	15		
7	5	18		
8	6	21		
9	7	24		
10	8	27		
11	9	30		
12	10	33		

1.6.2 İşlevler

Fonksiyonlar, bir tablodaki verilerle belirli hesaplamaları yapmak için kullanılabilen hazır formüllerdir. İşlevlerin kendi adları vardır ve kategoriler halinde gruplandırılmıştır: matematiksel, istatistiksel, mantıksal vb. İşlevler, hücre adresleri, hücre sıralaması veya biçimi olmayan hücrelerle çalışır.

Eşittir işareti (=) işlevden önce yazılır, ardından işlev adı ve son olarak işlev bağımsız değişkenleri gelir. Fonksiyonun argümanları aslında hesaplamaya katılacak hücrelerin etiketleridir.



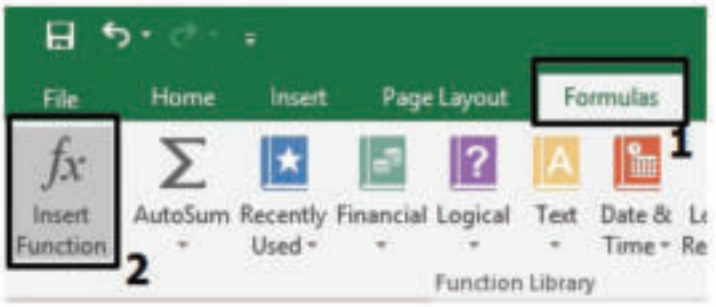
Şekil 3: Fonksiyon öğeleri

En sık kullanılan işlevler şunlardır:

SUM - bir hücre grubundaki çok sayıda değerin toplamını hesaplar.
MIN - bir grup hücreden en küçük sayısal değeri verir.
MAX - bir grup hücreden en büyük sayısal değeri verir.
AVERAGE - bir hücre grubundaki sayısal değerlerin ortalamasını hesaplar.

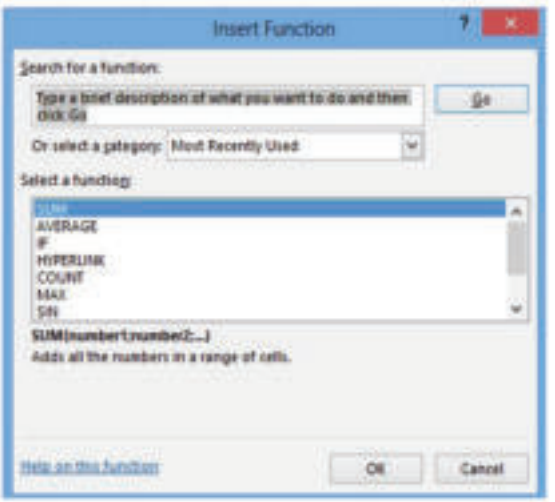
COUNT - seçilen hücrelerdeki sayısal girişlerin sayısını verir.
COUNTIF- belirli bir koşul karşılanırsa seçilen hücrelerdeki girişlerin sayısını verir.

Bir fonksiyon girme prosedürü, hesaplamanın yapılacağı hücreyi seçerek başlar ve ardından **Formulas -Formüller** menüsü ve **Insert Function - Fonksiyon Ekle** komutu ile fonksiyonu girmeye devam ederiz.



Şekil 4: Fonksiyonları girme işlemi

Ondan sonra, gelen görüntüde bir kategori seçip ve gereken işlevi seçiyoruz.



Şekil 5: Fonksiyonu seçme

Hesaplamayı seçilen fonksiyonun yardımıyla gerçekleştirmek için bir sonraki pencereden argümanlar ekliyoruz:

Şekil 6: Fonksiyon argümanlarının seçimi

Pencereden **OK** - Tamam butonuna tıkladığınızda hesaplamanın sonucunu alacağız.

Ancak bazen formül veya işlevin sonucunun görüntülenmesi gereken hücrede hata semboller görüntülenir: #####, # DIV / 0!, #NAME?, #REF!, # VALUE!. Bunlara formül veya işlevde hata mesajları da denir (**Error message**).

Hata uyarıları	Anlamı
#####	Neticelerin gösterilmesi için sütun dar dır
#DIV/0!	Sıfır ile bölme olmaz
#NAME?	Formülde yanlış yazılan hücre adresi var
#REF!	Formülde silinmiş olan hücre adresi var
#VALUE!	Formülde bir hücrede yazı var

Tablo 2: Formül veya işlevdeki hata mesajları

Ödev

Aşağıdaki şekilde gösterildiği gibi elektronik tablolama programında bir elektronik tablo oluşturun:

	A	B	C	D	E	F	G
1		Aylara göre okunan kitap sayısı					
2	Öğrencinin adı ve soyadı	Ocak	Şubat	Mart	Nisan	Mayıs	Toplam okunan kitap
3	Taha Berk	4	3	2	1	0	
4	Ahmet Fatih	5	2	2	1	1	
5	Belinay Ali	4	3	1	2	1	
6	xxxx	3	1	2	1	1	
7	xxx	6	1	3	2	0	
8							
9	Aylara göre toplam okunan kitap						
10	En çok okunan kitap						
11	En az okunan kitap						
12	Okunan kitabın ortalama değeri						
13	Toplam veriler						
14	HİÇ kitap okumayan ğrenci sayısı						

SUM, MIN, MAX, AVERAGE, COUNT ve COUNTIF işlevlerini kullanarak gerekli hesaplamaları gerçekleştirin.

Ezberle!

Elektronik tablolama programlarındaki formüller, verilerden ve bunlar arasındaki çeşitli matematiksel operatörlerden oluşturulan matematiksel ifadelerdir. Formüller hücrelerdeki verilerle değil, adresleriyle ilgilenir. Fonksiyonlar, bir tablodaki verilerle belirli hesaplamaları yapmak için kullanılabilen hazır formüllerdir. Fonksiyonların kendi adları vardır ve kategorilere göre gruplandırılmıştır: matematiksel, mantıksal, istatistiksel vb.

Sorular

1. Formüller nedir ve işlevler nedir?
2. Formüller ve işlevler arasındaki farkı açıklayın!
3. En sık kullanılan işlevler hangileridir?
4. Ünitede öğretilenlerin yanı sıra diğer Excel işlevlerini keşfedin, numaralandırın ve listeleyin!
5. Ardışık hücrelerden çeşitli değerlerin toplamını hesaplayan işlevin sözdizimini yazın!

1.7 Verileri sıralama

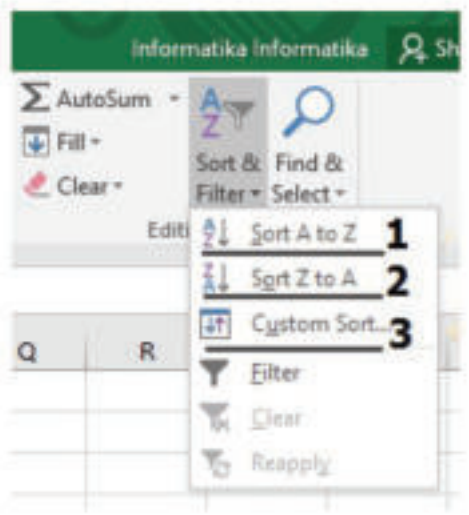
Çalışma sayfası çok fazla çerçeve içeriyorsa, bilgileri hızlı bir şekilde bulmak zor olabilir. Filtreler, çalışma sayfasındaki verileri daraltmak için kullanılabilir ve yalnızca kullanıcının ihtiyaç duyduğu bilgilerin görülmesine izin verir. Elektronik tablodaki veriler, kullanıcının ihtiyaçlarına göre düzenlenebilir ve raporlar oluşturulabilir. Bu, aşağıdaki işlemlerle yapılır:

- verileri belirli bir sırayla sıralamak;
- verileri belirli kriterlere göre filtrelemek;
- toplu ve alt küme verileri oluşturma.

Bu öğretim biriminde, diğerlerinden ayrı olarak erişimi kolaylaştıran özel bir araç olarak verilerin sıralanmasına odaklanacağız.
Sıralama, tablodaki satırların sırasını değiştirmek anlamına gelir, böylece içlerindeki veriler belirli bir sırayla düzenlenir. Excel'de aşağıdaki sıralama yöntemleri mümkündür:

- azalan veya artan düzeyde sıralama;
- alfabetik, sayısal, kronolojik veya kullanıcı tanımlı sırayla sıralama;
- tanımlı koşullu biçimlendirmeye göre sıralayın.

Sıralama tek bir kritere, yani bir sütuna göre yapılacaksa, o zaman kullanıcının sıralamak istediği sütunun bir alanına ve ardından Home - Ana Menüden tıklamak yeterlidir.



Alfabetik sıraya göre sıralamak

Ters alfabetik sırada sıralama

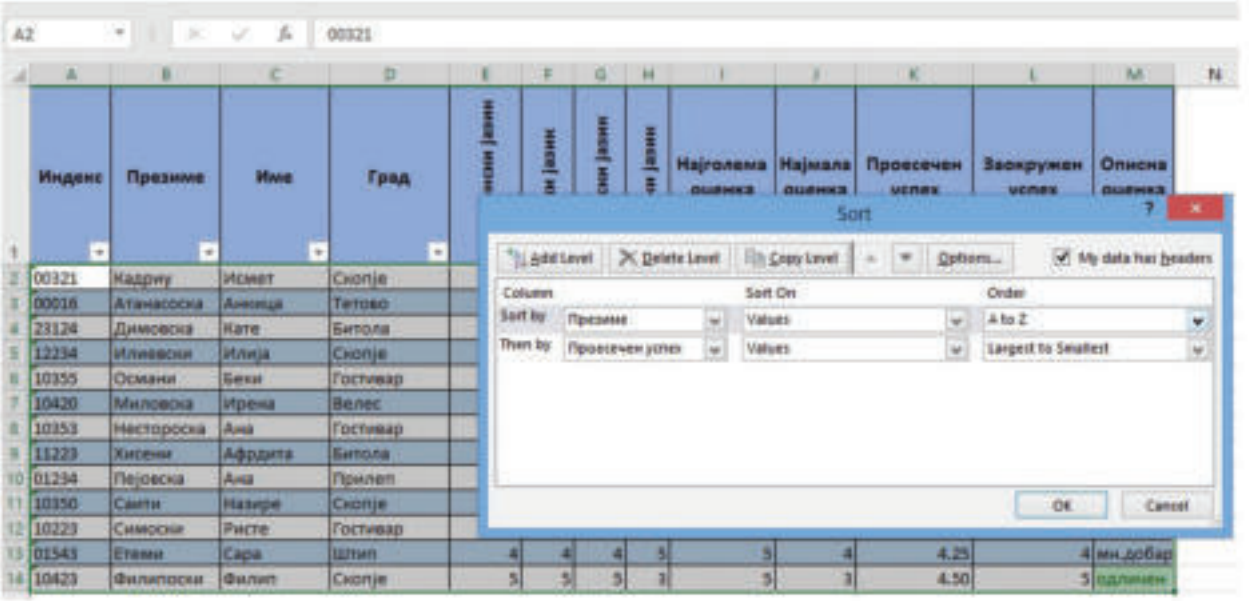
Belirli bir duruma göre sıralama

Şekil 1: Sıralama

Kullanıcı **Custom Sort - Özel Sıralama** seçeneğini seçerse, alanda
Column (Sort by) sıralanacak kriterleri seçer, **Sort on** alanında sıralanacak veri türü seçilir ve 3. alanda sıralama düzeni seçilir. Örneğin, aşağıdaki tablo aşağıdaki kriterlere göre sıralanır:

1. Öğrencinin soyadına göre alfabetik sırayla;

2. İkinci kriter olarak ortalama başarı, en yüksek ortalamadan en küçüğüne doğru.



Şekil 2: Verileri Özel Sıralama ile sıralama

Ödev

Elektronik tablolama programında aşağıdaki "Kütüphane verileri" tablosunu oluşturun:

	A	B	C	D	E	F
1	**Kütphane verileri**					
2	*Sıra no*	*Kitap*	*Yazar adı*	*Yazarın soyadı*	*Sınıf*	*Verildi*
3	1	Мајка	Јован	Јовановиќ - Змај	II	84
4	2	Зоки Поки	Оливера	Николовска	II	68
5	3	Поезија	Рифат	Кукај	III	35
6	4	Шеќерна приказна	Славко	Јанески	III	44
7	5	Сказна за детето Вилен	Глигор	Прличев	IV	54
8	6	Училиште	Драган	Луќиќ	III	30
9	7	Орхан	Нецати	Зекирија	IV	57
10	8	Летни цвеќиња	Наим	Фрашери	VI	67
11	9	Силјан Штркот	Марко	Цепенков	VII	46
12	10	Принцезата Арѓиро	Исмаил	Кадаре	VII	44
13	11	Бегалка	Видое	Подгорец	VIII	70

Alfabetik sırayla "**Kitap**" kategorisine göre sırala ve ardından "**verildi**" kategorisinde en büyük değerden en küçüğüne kadar kralanma sayısı yazılsın.

	A	B	C	D	E	F
1	Kütphane verileri					
2	*Sıra no*	*Kitap*	*Yazar adı*	*Yazarın soyadı*	*Sınıf*	*Verildi*
3	1	Мајка	Јован	Јовановиќ - Змај	II	84
4	2	Зоки Поки	Оливера	Николовска	II	68
5	3	Поезија	Рифат	Кукај	III	35
6	4	Шеќерна приказна	Славко	Јанески	III	44
7	5	Сказна за детето Вилен	Глигор	Прличев	IV	54
8	6	Училиште	Драган	Луќиќ	III	30
9	7	Орхан	Нецати	Зекирија	IV	57
10	8	Летни цвеќиња	Наим	Фрашери	VI	67
11	9	Силјан Штркот	Марко	Цепенков	VII	46
12	10	Принцезата Арѓиро	Исмаил	Кадаре	VII	44
13	11	Бегалка	Видое	Подгорец	VIII	70

Ezberle!

Sıralama, verileri kurallara göre sıralama işlemidir. Sıralama alfabetik sırada, ters alfabetik sırada veya **Custom Sort - Özel Sıralama** seçeneğiyle verilen tanımlı koşullara göre olabilir.

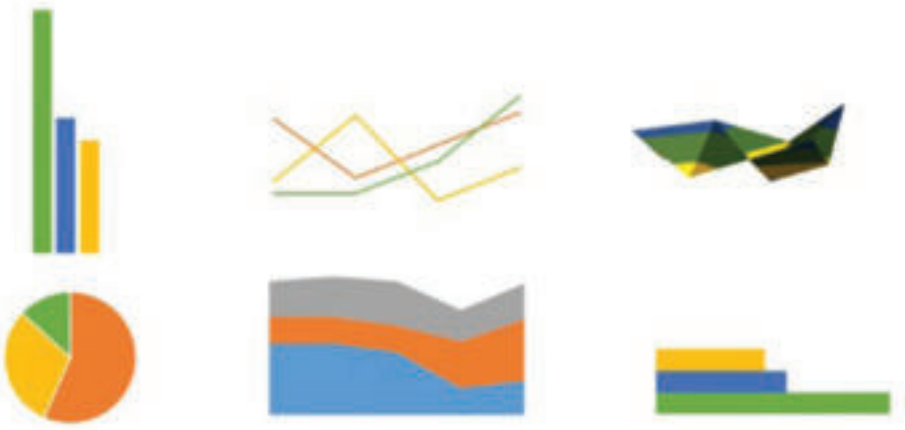
Sorular

1. Sıralama nedir?
2. Verileri alfabetik sıraya göre sıralama prosedürü nedir?
3. Verileri belirli bir koşul altında nasıl sıralayabiliriz?

1.8 Grafik oluşturma

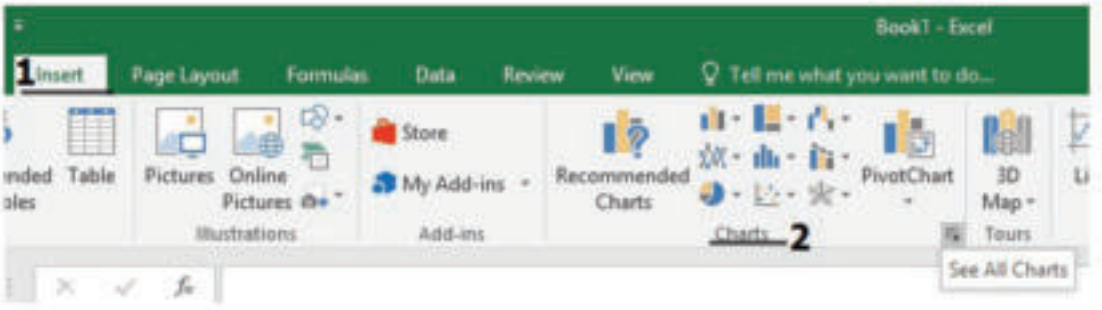
Bir tablo oluşturulduktan sonra, önceki öğretim ünitelerinde listelediğimiz prosedürlere göre biçimlendirip oluşturulduktan sonra, genellikle verileri resimli bir şekilde, yani bir grafik kullanarak sunmak gerekir. Verilerin resimli gösterimi, kullanıcının verilerin değerlerini, karşılaştırmalarını ve analizlerini hızlı ve net bir şekilde görmesini sağlar.

Buna göre **grafikler, tablodaki verilerin grafiksel bir temsilidir.** Aslında grafik, verileri bir görüntüye dönüştürür. Grafikler şunlar olabilir: çubuk, çizgi, pasta, simit, vb. Hepsi farklı amaçlar için kullanılır. Örneğin: kategorilere göre karşılaştırma, yüzde değerlerin karşılaştırması, vb. Resimde gösterildiği gibi.



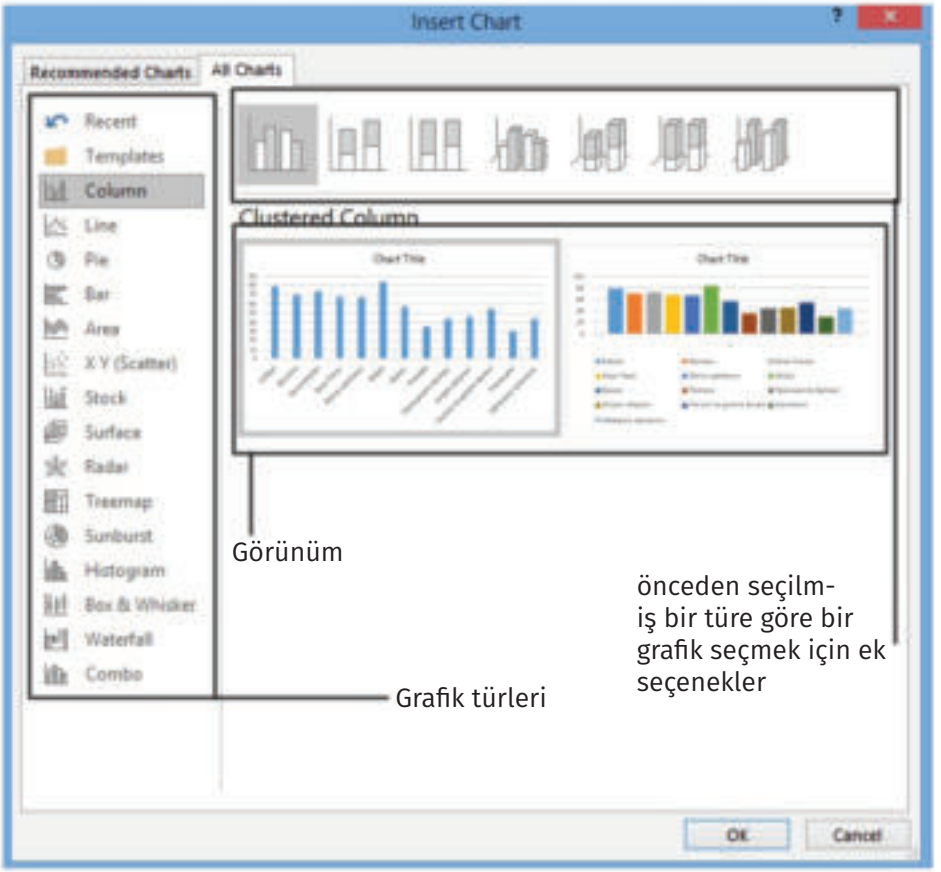
Şekil 1: Grafik türleri

Bir elektronik tablolama programında veriler çalışma sayfasında düzgün bir şekilde yazılır ve düzenlenirse, bir grafik oluşturma prosedürü başlatır. Ardından, bir grafikle görüntülenmesi gereken veri hücreleri aralığını seçer. Grafik olarak görüntülenecek veriler seçtikten sonra, **Insert - Ekle** menüsü ve **Chart** bölümü aracılığıyla Excel'de grafikler oluşturma komutunu uygulayabiliriz:



Şekil 2: Grafik ekleme işlemi

Resimden, bir grafik eklerken kategoriyi, yani grafiğin türünü seçebiliriz veya aşağıdaki pencereyi görüntüleyen **Chart - Grafik** bölümünün sol tarafında bulunan **Quick Launcher - Hızlı Başlatıcı** aracılığıyla hemen seçebileceğimizi fark ediyoruz:



Şekil 3: Grafik türlerini seçme prosedürü

Tamam düğmesine tıklandığında, grafik veri kaynağının, yani kaynak tablonun bulunduğu çalışma sayfasında görünür. Ardından, grafiğin aynı çalışma sayfasında kalması veya yeni bir çalışma sayfasına taşınması için gözden geçirilip geçirilmeyeceğine karar veririz. Grafik seçiliyken, grafiğin düzenini biçimlendirmenize ve düzenlemenize olanak tanıyan **Design - Tasarım** ve **Format - Biçim** menüleri görüntülenir. Örneğin, **Tasarım** menüsü aracılığıyla aşağıdaki olasılıkları uygulayabiliriz:

- grafik öğeleri eklemek;
- grafik öğelerinin düzeni;
- renk seçimi;
- grafik stili;
- veri seçimi;
- grafik türünü değiştirmek;
- grafiği yeni yada aynı çalışma sayfasına aktarmak.

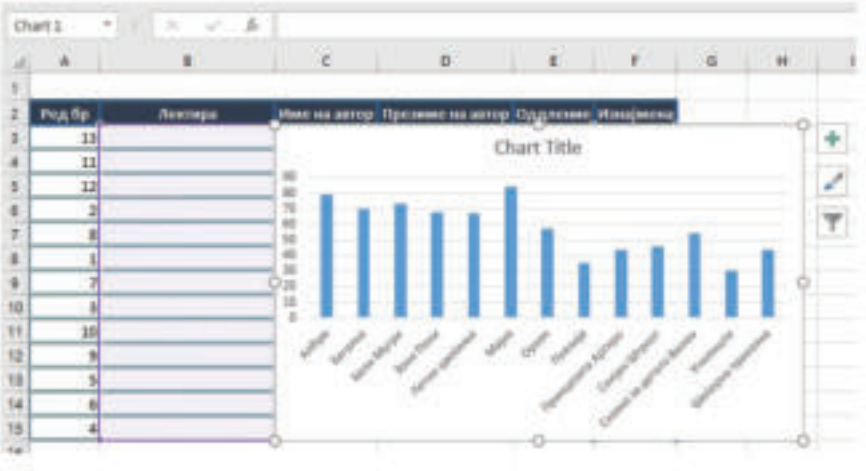
Biçim menüsü aracılığıyla grafiğin bölümlerine, çizgi rengine, eleman rengine, harf stillerine, eleman boyutlarına ve daha fazlasına şekiller ekleyebiliriz.

Not!

Formüller oluştururken, verilerin değerini değiştirirken sonucu otomatik olarak değiştirmek için hücrelerdeki değerlerle değil adresleriyle çalışacağız.

Ödev

Masaüstü klasöründe depolanan "Kitaplık Verileri" tablosunu açalım. Aşağıdaki görüntüde gösterildiği gibi Sütunlar grafiğini kullanarak hangi kitabın en çok okunduğunu gösteren bir grafik oluşturun:



Grafik yeni bir çalışma sayfasında olmalı ve **Design - Tasarım** ve **Format - Biçim** menüsündeki araçlar kullanılarak biçimlendirilmelidir.

Ezberle!

Grafikler, tablo verilerinin grafik temsilleridir. Verilerin değerleri, karşılaştırmaları ve analizleri hakkında hızlı ve net görünüm sağlarlar. Grafikler şunlar olabilir: çubuk, çizgi, pasta, simit vb.

Sorular

1. Grafik nedir?
2. Bir grafik oluşturma prosedürünü açıklayın?
3. 2016 Excel programı ne tür grafikler sağlar?
4. Grafiği düzenlemek ve formatlamak için Design - Tasarım ve Format - Biçim menüsünden hangi araçlar kullanılabilir?

TEKRARLAYALIM! UYGULAYALIM!

Ödev 1

MS Excel 2016'da bir çalışma sayfası açın. İlk çalışma sayfasına "ev" adını verin ve içinde farklı seçim yöntemleri kullanarak hücreleri renklendirerek bir ev elde edin. Belgeyi soyadınız ve sınıf adıyla kaydedin!

Ödev 2

Tablolama programında çalışma sayfası 1'de (Sayfa 1), sınıftaki öğrenciler için bir telefon rehberi oluşturun. Tablo şu verileri içermelidir: seri numarası, ad, soyad ve telefon numarası. Arkadaşınızın telefonuna ek olarak doğum gününü de ekleyin.

Ödev 3

Kütüphanenizdeki kitapların bir listesini yapın. Aşağıdaki bilgileri girin: Yazar, kitap adı ve yayın yılı.

Ödev 4

Verileri gireceğiniz bir tablo oluşturun: ay, gelir, giderler. Verileri girdikten sonra yeni bir sütun ekleyin: fark (gelir ve giderler arasındaki fark). Başlık hücrelerinin üstüne yeni bir satır ekleyin. Yeni satırın hücrelerini Fill - Dolgu rengiyle boyayın. Sütun boyutları yirmi beş (25), satır boyutlarına yirmi (20) yapın. Belgeyi kaydedin!

Ödev 5

Elektronik tablolama çalışma belgesinde aşağıdaki tabloyu oluşturun:

Ana dili

	A	B	C	D	E
1	Ders başarısı				
2	Ders	Notlandırılan	Notlandırılmayan	Olumlu notlar	Negatif notlar
3	Matematik	12	1	10	2
4	Bilişim	11	2	11	0
5	İngilizce	10	3	10	0
6	Ana dili				

Biçimlendirme araçlarını kullanarak tabloyu biçimlendirin: yazı tipi, boyut, stiller, sıralama, yönlendirme ve hücre stillerini veya hazır tablo formatları. Belgeyi kaydedin!

Ödev 6

Elektronik tablolama programında aşağıdaki tabloyu oluşturup, verilen geometrik şekillerin çevresini ve alanını hesaplamak için formüller yazın.

	A	B	C	D	E
1	Geometrik şekiller				
2		Ayrıt 1	Ayrıt 2	Çevre	Alan
3	Dikdörtgen	6	4		
4	Küp	3	3		
5	Kare	5	5		
6					

Ödev 7

Elektronik tablolama programında, ülkedeki ortalama sıcaklıkları aylara göre gösteren bir tablo oluşturun. Hangi ayın en sıcak ve hangisinin yılın en soğuk olduğunu belirlemek için işlevleri kullanın!

	A	B
1	Ocak	5.6
2	Şubat	6.2
3	Mart	8.8
4	Nisan	12
5	Mayıs	17
6	Haziran	21
7	Temmuz	23
8	Ağustos	23
9	Eylül	18
10	Ekim	15
11	Kasım	10
12	Aralık	6.9
13		

Ödev 8

Arkadaşlarınızın verileri ile bir tablo oluşturun: adı, soyadı, yaşı ve bunları ters alfabetik sırayla sıralayın.

Ödev 9

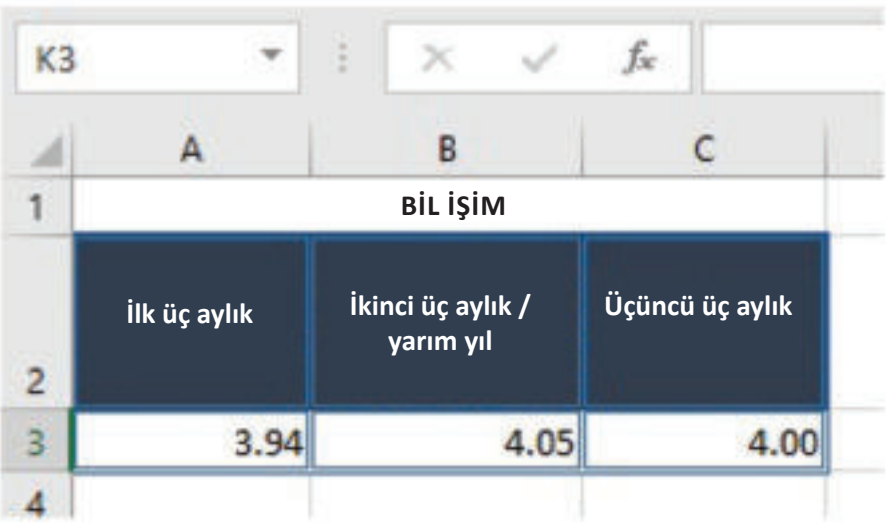
Elektronik tablolama programında, arkadaşlarınız için bir veri tablosu oluşturun ve biçimlendirin: ad, soyad, başarı. Bunları soyadlarının alfabetik sırasına göre ve ardından başarısına göre en büyükten en küçüğe doğru sıralayın.

Ödev 10

Elektronik tablolama programında VI - IX sınıflarından alınan notların başarısını karşılaştıracak bir grafik oluşturun. Bunu yaparken, yeni bir çalışma sayfasında oluşturulmuş bir pasta grafiği uygulayın.

Ödev 11

Şekilde gösterildiği gibi, her üç ayın sonundaki ortalama başarıyı gösteren bir grafik oluşturun. Ardından, Columns-Sütunlar grafiğini kullanarak verileri grafik olarak görüntüleyin.



Ödev 12

Şehrinizde, metrekare başına su maliyetini araştırın ve ne kadar ödemeniz gerektiğini hesaplayın!

Proje çalışması

Elektronik tablolama programında, çalışma sayfası 1'de aşağıdaki tabloyu oluşturun:

Sıra no	Soyad	Ad	Şehir	Makedonca	İngilizce	Almanca	Arnavutça
1	Кадриу	Исмет	Тетово	4	5	5	3
2	Атанасоска	Анкица	Скопје	5	5	5	5
3	Димовска	Кате	Прилеп	4	4	2	3
4	Илиевски	Илија	Битола	3	3	2	2
5	Османи	Бакиja	Скопје	3	3	5	4
6	Миловска	Ирена	Штип	4	4	4	5
7	Нестороска	Ана	Гостивар	5	4	4	2
8	Хисени	Афродита	Велес	2	2	3	3
9	Пејовска	Ана	Битола	2	3	4	5
10	Сапти	Назире	Скопје	5	3	4	2
11	Симоски	Ристе	Гостивар	5	2	3	4
12	Етеми	Сара	Скопје	5	5	5	3
13	Филипоски	Филип	Гостивар	3	3	5	5
14	Спироски	Ристе	Валандово	3	4	4	4

- Diller içeren hücre başlıkları metnin dikey yönüne sahip olmalı ve Wrap Text - Metni Kaydır seçeneği uygulanmalıdır.
- "Arnavutça" sütununun yanına "En yüksek not" sütununu ekleyin, en yüksek notu hesaplamak için bir fonksiyon yazın.
- "En yüksek not" sütunu yanına "En düşük not" sütununu ekle ve bir fonksiyon yardımıyla en düşük notu hesapla.
- Her öğrencinin ortalama başarısını hesaplamak için tablonun sonunda sütun aç.

- A15 ile J15 arasındaki hücreleri birleştirin ve sınıf ortalamasını yazın.
- Ortalama notu "iyi" olan öğrenciler kırmızı renklendirilmeli ve "en iyi notlu" öğrenciler koşullu biçimlendirme kullanılarak yeşile boyanmalıdır.
- K15 hücresindeki sınıfın not ortalamasını hesaplayın.
- Dersler sütunlarının altında, derslerde sırasıyla kaç tane beşli, dörtlü, üçlü ve kaç tane iki olduğu hesaplansın.
- Tabloyu şu şekilde biçimlendirin: yazı tipi, yazı tipi rengi, yazı tipi stilleri, hücre arka plan rengi, çerçeve türü ve görünürlüğün görüntülenmesi sağlanan çerçeve rengi. Hazır elektronik tablo formatları / tasarımları da kullanılabilir.

- Verileri A ile başlayarak soyadına göre sırala ve ikinci koşul başarısına göre en iyiden başlayarak sıralayın.
- Öğrenci başarısını gösteren ayrı bir çalışma sayfasında bir grafik oluşturun. Grafiği istediğiniz gibi biçimlendirin.
- Belgeyi kaydedin!

Mantıksal rekabetçi ödevleri çözerek bilgi kavramlarına giriş

Mantıksal rekabetçi ödevleri çözerek bilgi kavramlarına giriş
Mantıksal rekabetçi problemleri analiz etme ve çözme
Ödevin bilgisayar bilimindeki kavramlarla ilişkisi - bilgi kavramları
Veri yapıları
İkili sayılar
Kriptografi

TEKRARLAYALIM! UYGULAYALIM!

2 Mantıksal rekabetçi ödevleri çözerek bilgi kavramlarına giriş

Ne öğreneceğiz?

Formülleri ve işlevleri kullanarak, biçimlendirme araçlarıyla düzenlenmiş ve grafiklerle grafik olarak temsil edilen farklı veri türleri ve hesaplama dizileri içeren tablolar oluşturmak.

Bilgisayar bilimi okurken, diğerlerinin yanı sıra, en heyecan verici ve ilginç şeylerden biri, yeni bir düşünme ve problem çözme ödevlerini ve olayları çözme yöntemini öğrenmektir, buna **bilgisayar düşüncesi** denir. Bu düşünce tarzı 21. yüzyıl için çok önemlidir ve günlük yaşamda, eğitim sırasında ve mesleki gelişimde birçok durumda başarıya ulaşmanın temeli olarak görülmektedir.

Bir problemi veya problem durumunu çözmek için bilgisayarları kullanmadan önce, problemin özünü anlamak gerekir. Bu nedenle, bilgisayarla düşünmenin, yaratıcılık, açıklama ve kanıtlama yeteneği, takım çalışması ve işbirliği gibi bir dizi çoklu beceri olduğu söylenebilir.

Bilgisayar düşüncesinin ana unsurları:

- mantıksal düşünme;
- algoritmik düşünme;
- etkili çözme,
- bilimsel düşünme;
- yenilikçi düşünme.

Bu konuda, bilgisayar düşüncesinin özel ve temel bir becerisi olan **mantıksal düşünmeye** özel bir vurgu yapacağız. Mantıksal düşünme genellikle bilgisayar kavramlarıyla ilişkilendirilir, ancak son zamanlarda bilgisayar fikirlerini diğer disiplinlere entegre etmekle eşanlamlı hale geldi.

Bir şeyin mantıklı olduğunu söylediğimizde, aslında o şeyin anlamlı olduğunu düşünürüz. Bireyin gerçeklere ve kanıtlara dayalı bir şekilde düşünme yeteneği, mantıksal düşünme yeteneği olarak bilinir. Dolayısıyla **mantıksal düşünme, belirli bir sonuca varmak için objektif muhakemenin kullanıldığı bir süreç olarak tanımlanabilir.**

Deneyin!

Mantık terimini keşfedin! Tanımla ve dene mantıksal düşünme ile eleştirel düşünme arasındaki farkları araştırıyor mu? Aralarındaki benzerlikler ve farklar nedir?

Mantıksal düşünme yardımıyla çözülen problem problemleri veya durumları bir yapıya, gerçekler arası bağlantılara ve anlam taşıyan, yani mantıksal bir neden-sonuç ilişkisine sahiptir. Bu düşünme biçimi, aşağıdakiler gibi derinlemesine **analizi** içerir: mevcut tüm seçenekleri ölçme, gerçekleri ve rakamları kullanma, artıları ve eksileri temel alan önemli kararlar verme, bu da böyle bir süreçte duyguların hesaba katmadığı anlamına gelir. Mantıksal düşünmenin en yaygın örneği **SUDOKU**'dur. Birkaç hücrede bulunan değerler hakkında bilgi verilir. Bulmacayı çözmek için kurallara uyarak diğer tüm hücrelerdeki değerleri kilitlemeniz gerekir. En ufak bir hata yapılırsa çözümden uzaklaşırız.

Bu süreç görünmez olsa da, akıl yürütme becerilerimiz sayesinde her gün üstesinden geldiğimiz çeşitli olay ve durumlarla karşılaşıyoruz. **Örneğin**: Süpermarkette, ihtiyacımız olan her şeyi daha düşük bir fiyata alıp alamayacağımızı görmek için fiyatları hesaplarken veya tüm sorumluluklarımızı bir güne sığdırmaya çalışırken, bizim düşünen makinemiz doğru çözümü bulmak için sürekli düşünüyor. Buna göre mantıksal düşünmede saygı duymamız gereken en önemli şeyler şunlardır:

- **Olaylara sadece kendi bakış açınızdan bakmayın.** Özneldir ve bizi hedefe, yani nihai çözüme götürmez. Örneğin: Mert ve Berat bir restoranda öğle yemeği yiyorlar. Berat öğle yemeğinin tadını çıkarırken Mert'in tabağında hoş olmayan ve kötü bir koku var. Mert yemeğin kokusunu beğenmediğinden, yemeğin sağlıksız ve kötü hazırlanmış olduğu sonucuna vardı. Bu bir sonuca varmanın mantıklı bir yolu değil, çünkü Mert"in yiyeceklerinin sağlıksız ve kötü hazırlanmış olduğuna dair hiçbir kanıtı yok. Mantıklı bir sonuca varmak için, kişinin kendi öznel görüşlerini dışlamalı ve yemeği hazırlamak için hangi bileşenlerin kullanıldığı, hazırlama prosedürünü bilmeli, varsayımlara güvenmemeli ve kanıtlanmış bilgilere odaklanmalıdır.

- **Başlamadan önce düşünün** - Bir strateji oluşturun. Strateji, düşünme sürecinde büyük rol oynar. Araştırmanıza gerçekleri yorumlamanıza yardımcı olacak sorular sorarak başlayın. Büyük resme geçmeden önce ayrıntıları aktif bir şekilde arayın, ayrı parçalar ve grup olarak nasıl çalıştıklarını öğrenin.

Kelimelerin anlamlarına dikkat edin. Küçük dil varyasyonları, ödevin veya problem durumunun anlamında büyük bir fark yaratır. İfadeler arasındaki farkı belirlemek, mantıksal düşünmedeki tutarsızlıkları ve belirsizlikleri kesinlikle ortadan kaldıracaktır. Örneğin: "gerekli" kelimesi "yeterli" den farklıdır. Gerekli, yeterli olanın aksine koşul veya prosedürün karşılanması gerektiği anlamına gelir; bu, bir çözüme ulaşmak için asgari çaba sarf etmek anlamına gelir.

Son olarak, mantıksal düşünme, problemleri çözmek için bir dizi beceri ve tekniklerden oluşur aynı zamanda farklı amaçlar için yazılım / program oluşturma süreci olarak programlamadan önceki bir adımdır.

Anahtar terimler!

bilgisayarlı düşünme, mantıksal düşünme, analiz, programlama, veri yapısı, ikili sayılar, kodlama, kriptografi, özgür yazılım.

Hatırlamak!

Bilgisayar düşüncesi, mantıksal düşüncenin yardımıyla nesnel bir çözüme ulaşan, sorunlu ödevleri ve durumları çözmenin yeni bir yoludur. Mantıksal düşünme, farklı teknikler ve yöntemler kullanılarak farklı yollarla elde edilen gerçeklere ve kanıtlara dayanır. Bu düşünce tarzı asla varsayımlara, arzulara, duygulara ve duygusal durumlara dayanmaz.

2.1 Mantıksal rekabetçi ödevleri analiz etme ve çözme

Problemin veya mantıksal ödevin **analizi**, nihai çözüme giden yolu başlatmak için objektif olarak yapmamız gereken ilk ve temel adımdır. **Analiz** yardımıyla ödev veya problem durumunu ayrıştırır, bütünün gerçeklerini veya unsurlarını keşfeder, sonuca ulaşmak için bağlantıları keşfederiz.
Ödev **analizi**, aşağıdaki gibi özetlenebilecek üç adımı içerir:

Şekil 1: Bir problemi analiz etme adımları

Problem analizi dikkatlice organize edilirse ve uygun adımları takip ederse, tam olarak içerdiği detaylı, dikkatli ve entegre analiz nedeniyle birçok roblemi çözmek ve nihai bir çözüme ulaşmak için uygulanabilir.

Deneyin!

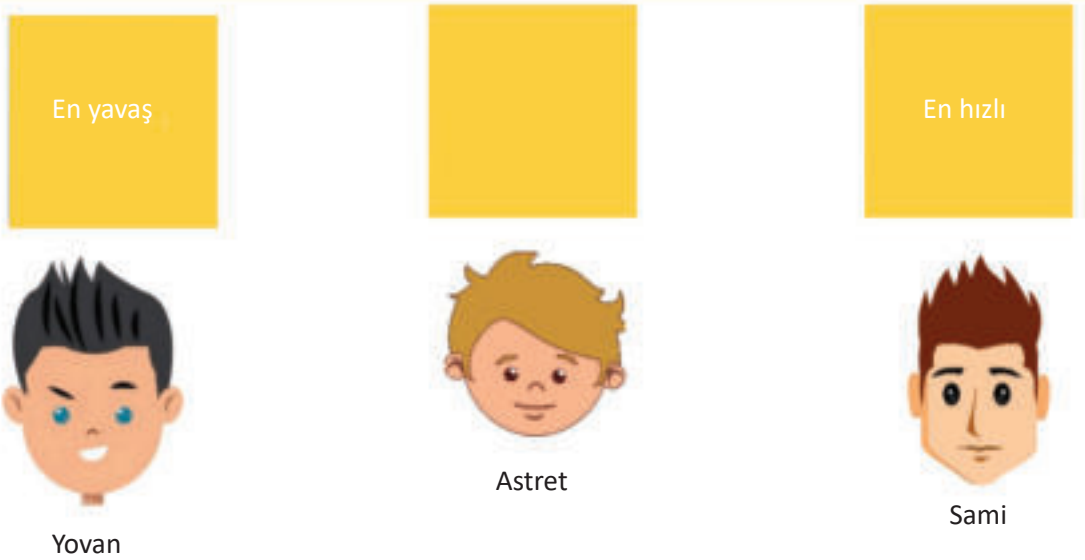
Analizin, problemi ayrı unsurlara veya parçalara ayırma süreci olduğunu biliyorsak, analizin tersi olan süreci araştırmaya çalışın!

Mantıksal düşünme hakkında edinilen bilgileri uygularken ana hatlarıyla belirttiğimiz bazı teknikleri uygulayarak birkaç problem ödevini analiz edelim.

Ödev

Yusuf, Özgür ve Sami akrandırlar ve aynı okula gidiyor. Üçü de ulusal ve uluslararası yarışmalarda çok sayıda ödül kazanan iyi sporcudur. Verilen ifadelerin tümü doğruysa, hangisi en yavaştır?

1. Yusuf en yavaş değildir
2. Özgür en hızlıdır
3. Sami en hızlı değildir



İfadelerden biri **Astret'in** en hızlısı olduğunu belirtiyor ve bu nedenle Astret karakterini "en hızlı" olarak işaretlenen kareyle ilişkilendiriyoruz. Artık boş bırakılan iki alan var. **Yovan'ın** "en yavaş olmadığı" ve bu yüzden onu ikinci kareye bağladığımız da belirtiliyor. Yalnızca "en yavaş" olarak işaretlenen ilk alan boş kalır ve biz onu kendimiz bağlarız. Bu nedenle en yavaş kim sorusuna doğru cevap **Sami'dir.**

Belirlenen ödevi çözerken **soyutlama** tekniğini kullandık çünkü ödevin en önemli özelliklerini çıkardık. Belirlenen özellikler bize problem ödevini incelemeye nereden başlayacağımızı, hangi yoldan gideceğimizi ve olası bir çözüme nasıl ulaşacağımızı öğretti. Aynı şey **bilişimde** sorunlu bir durumu çözerken de olur. Her zaman problem durumunun çözümü için önemli olan bilgilerle ilgilenir ve problemi çözmek için gereksiz bilgiler atılır veya ihmal edilir.

Ödev

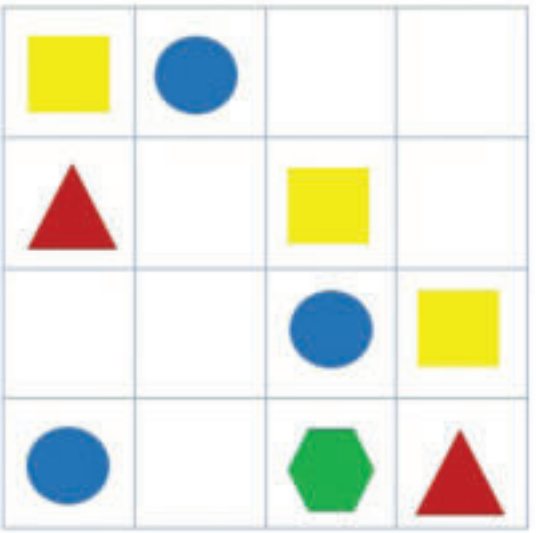
Önceki probleme benzer şekilde, bir sonraki problemi kendi başınıza doğru çözümü bulmaya çalışın!
Tüm ifadeler doğruysa, hazine nerededir?

Hazine 2'de değildir | **Hazine 1 ya da 3'te dir** | **Hazine burda değildir**

Sorular ve ödevler

Sudoku, mantıksal düşünmeyi ve sonucu elde etmek için ilgi çekici olmayan gerçekleri ortadan kaldırma sürecini içeren en popüler oyunlardan biridir.

Basit bir Sudoku ödevini çözelim!

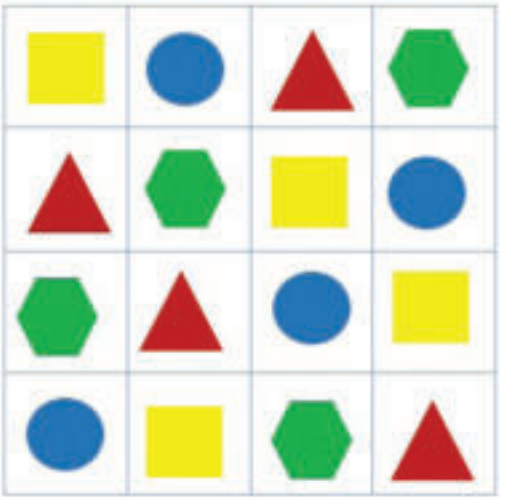


Tabloda bazı alanlarda geometrik şekiller verilmiştir ve bazı alanlar boştur. Boş alanlar aşağıdaki şekillerle doldurulmalıdır:

Her geometrik şekil, yatay ve dikey olarak her zaman aynı türden sadece bir şekil olacak şekilde düzenlenmelidir. Dairenin tabloya yalnızca bir kez eklenebileceğini ve bu nedenle uygun yerin belirlenmesi gerektiğini, yani yatay ve dikey yönde yalnızca bu tür bir figür olması gerektiğini not ediyoruz. Doğru yeri bulalım. Birinci, üçüncü ve dördüncü satırlarda (yatay olarak), zaten yuvarlak bir şekil vardır, bu nedenle ikinci satır bir seçenek olarak kalır.

Ama ikinci sıradaki hangi alanda? Hangi sütunda?

Birinci, ikinci ve üçüncü sütunlar halihazırda bir daire şekli içerir ve bu seçenek dışında tutulur. Bu nedenle, birinci, üçüncü ve dördüncü satırların yanı sıra birinci ikinci ve üçüncü sütunları bir olasılık olarak atlıyoruz. İkinci satıra ve dördüncü sütuna daire şeklini eklemenin tek yolu kalır ve bu doğru cevaptır. Daire ile örneğe göre tüm alanları doldurursak şu sonuca ulaşırız:



Deneyin!

Sayı alanlarını doldurma kurallarını uygulayarak Sudoku çözmeye çalışın! Pek çok elektronik örnek var, basılı medyada bulunabilir.

Mantıksal rekabet halindeki problemleri çözerken **eliminasyon** süreci, Sudoku örneğinde olduğu gibi, önemli olmayan ve bizi problem durumunun nihai çözümüne götürmeyen gerçeklerin dışlanmasına izin verir. Günlük yaşamda, eleme en çok, birden fazla olası yanıt sunan testleri çözmede kullanılır, böylece olası doğru yanıt seçeneklerinin sayısını azaltmak için yanlış yanıtları ortadan kaldırılır.

Bu nedenle, mantıksal ödevleri analiz ederken, nihai çözüme ulaşmak için mantıksal düşünme tekniklerini kullandık. Burada verilen kuralları soyutlama, ortadan kaldırma ve bunlara uyma tekniklerini kullandık. Bununla birlikte, problem durumları ve ödevleri çözerken, bilgi veya bilgisayar süreçlerinin bir parçası olarak, nihai çözüme ulaşmak için tek amaç ve en optimal şekilde ödevin gerekliliğinden ve doğasından kaynaklanan diğer teknikler kullanılır.

Ezberle!

Sorun durumlarının ve ödevlerin analizi objektif olmalı ve önyargılı olmamalıdır, yani öznellik unsurları dahil edilmemelidir. Analiz prosedürü üç adımı içerir: ödevi oluşturan unsurlara ayrıştırma, unsurlar arasındaki ilişkileri belirleme ve elde edilen sonuçlara göre yeniden yapılandırma. Bir ödevi veya problem durumunu analiz etmenin en yaygın teknikleri: soyutlama, eleme, kurallara uyma, benzerlikler bulma vb.

2.2 Görevin bilgisayar bilimindeki kavramlarla ilişkisi - bilgi kavramları

Mantıksal düşünme ve mantıksal rekabet halindeki ödevleri çözme, birçok bilimsel disiplinde kullanılabilen, ancak çoğunlukla bilgisayarlar ve **bilgi kavramlarıyla** ilişkilendirilen becerilerdir. Bu nedenle, mantıksal düşünmenin bir uygulama çözümleri oluşturma süreci olarak **programlamadan** bir önceki adım olması tesadüf değildir.

Buna göre **programlama, programlama dillerini kullanarak bir program yazma süreci olarak tanımlanmaktadır.** Programları oluşturan kişilere **geliştirici** denir.

Programın oluşturulmasının başlangıcından itibaren, geliştiriciler, her biri kendine özel bir şekilde çözüme, yani programın oluşturulmasına giden yolu açan çok sayıda adım gerçekleştirirler. Bu süreçte, geliştiricilere bilgi kavramları tanıtılır. Bu nedenle, bu öğretim içeriğinde bazılarını tanıyacağız.

2.2.1 Veri yapıları

Veri yapıları, verileri toplamanın, organize etmenin, işlemenin ve depolamanın özel bir biçimi veya yoludur. En yaygın kullanılan yapılar: satırlar, bağlantılı listeler, yığınlar, öncelikli satırlar, ikili ağaç vb.

Bu nedenle, veri yapıları şunları oluşturmak için kullanılabilir:

- Belirli bir çözüme ulaşmak için gereken adımların listesi.
- Bir hedefe ulaşmak için bir dizi talimat.
- Bir form veya şema vb. doldurma talimatları.

Ödev

Aşağıdaki resmi elde etmek için verilen etiketleri kullanarak bir dizi adım oluşturun:

Adım 1	2	3	4	5	6	7	8	9	10
Adım 11	12	13	14	15	16	17	18	19	20
Adım 21	22	23	24	25	26	27	28	29	30

Çözülen ödev sayesinde, nihai çözüm için bir dizi talimat oluşturmayı başardık, yani kareler ızgarasını görüntülenen resme göre renklendirdik.

Ödev

Hanoi Kulelerini duydunuz mu? İşte bir örnek! Bu örnek, yalnızca üç disk içeren bir ödevi gerçekleştirme adımlarını gösterecektir.

Amaç, sıraya dikkat ederek diskleri bir yerden başka bir yere taşımak için adımlar atmaktır.

Adım 1: 1 numaralı diski üçüncü sıraya aktarın.
Adım 2: 2 numaralı diski ikinci sıraya aktarın.
Adım 3: 1 numaralı diski, 2 numaralı diskin hemen üzerindeki ikinci sıraya aktarın.
Adım 4: 3 numaralı diski üçüncü sıraya aktarın.
Adım 5: 1 numaralı diski ilk sıraya aktarın.
Adım 6: 2 numaralı diski, 3 numaralı diskin üzerindeki üçüncü sıraya aktarın.
Adım 7: 1 numaralı diski, 2 numaralı diskin üzerindeki üçüncü sıraya taşıyın.

Belirtilen adımlara göre oyunun sağladığı hareket kurallarına dikkat ederek tüm diskleri bir yerden başka bir yere taşımayı başardık. Yedi adımda diskleri ilk satırdan son satıra taşıdık ve disklerin sırasını aynı bıraktık.

"**Hanoi Kuleleri**" ödevini yerine getirirken, yığının tepesindeki elemanların önce alınması için ana özelliği istifleme olan bir **veri yığını / yığın yapısı** ile tanıştık. Yani "**ilk giren, son çıkar**" ilkesine saygı duyulduğunu söylüyoruz. Elbette transfer sırasında, ödevin doğasından kaynaklanan kuralları takip ediyoruz.

Deneyin!

Bir biyoloji öğretmeninin yardımıyla, bir kelebeğin yaşam döngüsünü gösterebileceğiniz bir dizi oluşturmaya çalışın. Kelebeğin yaşam döngüsünün kaç dönemi vardır? Bu döngülerden herhangi biri atlatılabilir mi?

2.2.2 İkili sayılar

Hatırlayalım!

Bilgisayarların dili nedir? Veriler bu dili kullanarak nasıl sunulur?

İkili sayı sistemi, 0 ve 1 rakamları olmak üzere iki sayısal değerden oluşur. Yalnızca bir rakam 0 veya 1, bit olarak adlandırılır. Bir bit ile "doğru" veya "yanlış", "evet" veya "hayır", "meşgul" veya "serbest" ve diğer birçok durumu temsil edebiliriz.
İkili sayıların yardımıyla, bilgisayarlardaki veriler **sıfırlar ve birler dizisi** halinde temsil edilir. Bir kelimedeki harfleri ikili sayılara dönüştürmek **ikili koddur**. Herhangi bir doğal sayı bir dizi ikili sayı ile temsil edilebilir, örneğin:

Doğal sayılar	0	1	2	3	4	5	6	7	8	9	10
İkili sayılar	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010

Doğal bir sayıyı ikili gösterime dönüştürmek için, 2 tabanını derece ile kullanırız: n: 0, 1, 2, 3 vb. Örneğin, ikili sayıdaki 5 sayısı şöyle görünür: 0101 ve bu sonuç şundan gelir:

5	Tabanı 2 olan güçler	$2^3 = 8$	$2^2 = 4$	$2^1 = 2$	$2^0 = 1$
		$4 + 1 = 5$			
	Bit ler görünümü	0	1	0	1

Her alana doğal sayının ikili gösterimini yazmamız için aşağıdaki bulmacayı dolduralım:

Yatay:

1) 2
3) 5
4) 34
5) 20

Dikey

1) 17
2) 10
3) 10

Aşağıdaki rekabetçi ödevi mantıksal düşünme yardımıyla çözelim:

Ödev

7. sınıf öğrencileri, çoktan seçmeli sorular içeren bir coğrafya testine girdi. Öğretmen aşağıda gösterildiği gibi tabloya doğru cevapları 1 ve yanlışları 0 ile kaydetmiştir.

	Soru 1	Soru 2	Soru 3	Soru 4	Soru 5	Soru 6
Yaren	0	0	1	1	1	0
Selen	1	1	1	1	1	1
Taha	1	0	0	0	1	1
İrem	0	0	0	0	1	0
Ahmet	0	0	1	1	0	1
Ozan	1	0	1	0	1	1
Beren	0	0	1	1	1	1

Öğretmen, materyalin tam olarak benimsenmesi için ek yollar oluşturmak amacıyla tablodan hangi materyalin en zayıf ve hangisinin en çok benimsenmiş olduğu hakkında bir rapor oluşturmak istedi.

Tabloda listelenen verilere göre soruları yanıtlayın!

- Çoğu öğrenci hangi soruyu cevapladı?
- En az öğrenci hangi soruya cevap verdi?
- Hangi öğrenci en yüksek puana sahip?
- En zayıf öğrenci kim?
- Sonuçlara göre hangi öğrenciler ek derslere katılmalıdır?

2.2.3 Kriptografi

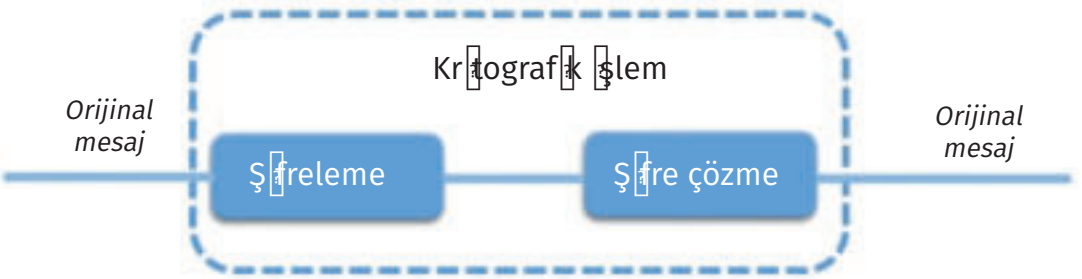
Kriptografi, yalnızca özel kişilere anlayabileceği bir şekilde mesaj gönderme ve alma yöntemlerinin araştırılmasıyla ilgilenen bilimsel bir disiplindir. Aslında, kriptografi kullanıcılara şunları garanti eden güvenlik kodların kullanımına izin verir:

- verilerin yetkisiz erişime karşı korunması;
- güvenli ve korunan iletişim;
- mesajların güvenirliği ve gizliliği;
- verinin kaynağını, yanıbilgiyi vb. kontrol etmek;

Kriptografi, kredi kartları, bilgisayar şifreleri, e-ticaret ve kullanıcı korumasının özellikle önemli olduğu diğer faaliyetlerle çalışırken kullanılır.
Kriptografinin, veri veya mesaj dönüşümlerini içeren bir süreç olduğu söylenebilir:

- şifreleme;
- şifre çözme.

Şifreleme, bir mesajı bir koda dönüştürme işlemidir ve **şifre çözme**, şifrelenmiş bir mesajı orijinal biçimine dönüştüren ters işlemdir. Bu işleme kriptografik işlem denir, aşağıdaki görüntüde gösterilmiştir:



Şekil 2: Kriptografik süreç

En ünlü kriptografi örneği telgraf iletişiminde kullanılan Mors kodudur. Aşağıdaki şekil, Mors kodu ile temsil edilen harf, sayı ve noktalama işaretini göstermektedir:

Ancak en yaygın kullanılan şifreleme modeli Sezar şifrelemesidir. Sezar şifreleme modelini kullanarak bir mesajını şifresinin nasıl çözüldüğünü görelim.

Ödev

Jana ve Azra VII. Sınıftan arkadaşlar. Bilişim dersinde bir "İnternet kullanımında güvenlik" proje görevi oluşturmaları gerekiyor ve bu amaçla İnternet'e ihtiyaçları vardır. Bilgisayara format atarken Internet'e erişim parolası silinmiştir. Jana'nın erkek kardeşi şifreyi hatırlıyor. Parolayı onlara vermek için parolanın bir şifre olduğu ve çözülmesi için aşağıdaki talimatların izlenmesi gerektiğini söylüyor.

1. Deşifreleme Sezar modeli ile yapılır;
2. Sezar'ın modeli gizli bir numara (anahtar) kullanır;
3. Şifreleme sırasında birkaç tamsayı anahtar olarak kullanılmıştır.

Parola	?	?	?	?	?	?	?	?	?	?	?
Anahtar	1	1	3	3	2	1	2	2	3	1	2
Şifre	S	M	T	M	O	Љ	Џ	P	Ж	J	Џ

Başlayalım! Alfabeyi bir tabloya yazalım ve sonra anahtarları kullanalım. Bizim durumda, şifre çözmek için anahtarlar verilmiştir, yani harfler 1, 2 ve 3 yer için sağa öteleyiniz.

Tablo 1: Anahtar 1 ile Sezar modeli

Harf	А	Б	В	Г	Д	Ѓ	Е	Ж	З	Ѕ	И	Ј	К	Л	Љ	М	Н	Њ	О	П	Р	С	Т	Ќ	У	Ф	Х	Ц	Ч	Џ	Ш
Anahtar 1	1	А	Б	В	Г	Д	Ѓ	Е	Ж	З	Ѕ	И	Ј	К	Л	Љ	М	Н	Њ	О	П	Р	С	Т	Ќ	У	Ф	Х	Ц	Ч	Џ
Şifre	Ш	А	Б	В	Г	Д	Ѓ	Е	Ж	З	Ѕ	И	Ј	К	Л	Љ	М	Н	Њ	О	П	Р	С	Т	Ќ	У	Ф	Х	Ц	Ч	Џ	

Tablo 2: Anahtar 2 ile Sezar modeli

Harf	А	Б	В	Г	Д	Ѓ	Е	Ж	З	Ѕ	И	Ј	К	Л	Љ	М	Н	Њ	О	П	Р	С	Т	Ќ	У	Ф	Х	Ц	Ч	Џ	Ш
Anahtar 2	1	2	А	Б	В	Г	Д	Ѓ	Е	Ж	З	Ѕ	И	Ј	К	Л	Љ	М	Н	Њ	О	П	Р	С	Т	Ќ	У	Ф	Х	Ц	Ч
Şifre	Џ	Ш	А	Б	В	Г	Д	Ѓ	Е	Ж	З	Ѕ	И	Ј	К	Л	Љ	М	Н	Њ	О	П	Р	С	Т	Ќ	У	Ф	Х	Ц	Ч	

Tablo 3: Anahtar 3 ile Sezar modeli

Harf	А	Б	В	Г	Д	Ѓ	Е	Ж	З	Ѕ	И	Ј	К	Л	Љ	М	Н	Њ	О	П	Р	С	Т	Ќ	У	Ф	Х	Ц	Ч	Џ	Ш
Anahtar 3	1	2	3	А	Б	В	Г	Д	Ѓ	Е	Ж	З	Ѕ	И	Ј	К	Л	Љ	М	Н	Њ	О	П	Р	С	Т	Ќ	У	Ф	Х	Ц
Şifre	Ч	Џ	Ш	А	Б	В	Г	Д	Ѓ	Е	Ж	З	Ѕ	И	Ј	К	Л	Љ	М	Н	Њ	О	П	Р	С	Т	Ќ	У	Ф	Х	Ц	

Ödevde bize kod ve anahtar veriliyor, hadi şifreyi bulalım!

Anahtar	1	1	3	3	2	1	2	2	3	1	2
Kod	S	M	T	M	O	Љ	Џ	P	Ж	J	Џ
Parola	И	Н	Ф	О	Р	М	А	Т	И	К	А

Ezberle!

Mantıksal düşünme ve mantıksal rekabetçi ödevleri çözme, programlamanın başlangıcıdır. Programlama, bir program yazma sürecidir. Program oluşturan kişilere geliştirici denir. Mantıksal ödevler ve çözümleri şu bilgi kavramlarıyla ilgilidir: Diziler, listeler, yığınlar / satırlar, öncelikli satırlar, ikili sayılar, kriptografi vb..

TEKRAR EDELİM! UYGULAYALIM!

Ödev 1

Okul girişinden sınıfınıza giden yolu gösteren bir dizi talimat oluşturun.

Ödev 2

12, 18 ve 36 doğal sayılarını ikili sayılara dönüştürün!

Ödev 3

11100111 ikili sayısını doğal sayıya çevirin!

Ödev 4

236 sayısını ikili sayıya çevirin!

Ödev 5

Caesar'ın modelini kullanarak şifreli bir mesaj oluşturun!

Ödev 6

Packman ve Tetris oyunlarının özelliklerini keşfedin. Hangi bilişim kavramlarıyla ilişkilendirebiliriz? Açıkla!

Ödev 7

Şadiye, Eslem'e, Sezar şifresini içeren bir mesaj göndermek ister. Şadiye'nin göndereceği mesaj nedir?

Ödev 8

Stephen, Amir'den bir mesaj aldı: Sezar koduyla kodlanmış "PGOP". Eğer Stefan anahtarın üç (3) numara olduğunu bilirse Amir, Stefan'a ne söylemek istedi?

Ödev 9

Jasmine, Tommy'ye bir mesaj gönderdi: " УНК ОТДSA". Ancak mesaj Elvin tarafından ele geçirildi. Elvin, Jasmine ve Tommy'nin yazıştığını ve her bir harfin altı (6) sıra hareket edecek şekilde karşılık geldiğini biliyor. Mesajı böyle deşifre etti. Elvin deşifre sırasında hangi mesajı aldı?

Ek ödevler:

Mantıksal düşünme, mantıksal rekabetçi ödevleri çözme pratiği yapmak için, www.talent.mk resmi web sitesinde yayınlanan, Dabar'ın ödev kılavuzlarını ve ödevlerin açıklamalarını kullanabilirsiniz.

Mantıksal rekabetçi ödevleri ve kullanabileceğiniz oyunları çözmek için internet adresleri şunlardır:

www.bebras.org

https://www.bbc.com/bitesize/articles/zqnc4wx

http://digit.mile.mk/

Gelişmiş görsel programlama

Görsel bir ortamda programlamaya giriş
Grafik programlama. Scratch programına giriş
Etkileşimli etkinlik programları
Daha karmaşık sorun durumlarına sahip programların geliştirilmesi
TEKRAR EDELİM! UYGULAYALIM!

3. Görsel bir ortamda programlamaya giriş

Hatırlayalım!

Programlar nelerdir? Programlama için tanımlar verin! Bir program nasıl geliştirilir? Hangi programlama dillerini biliyorsunuz?

Mantıksal düşünme ve mantıksal rekabetçi ödevleri çözme, **programlamaya** giriş niteliğindedir. Mantıksal düşünmenin programlamadan önce bir adım olduğunun söylenmesi tesadüf değildir.

Bu nedenle **programlama, bir problem ödevinin veya problem durumunun nihai çözümüne ulaşmak için bir düşünme yolu veya becerisi olarak tanımlanabilir.**

Şekil 1: Scratch logosu

Sonuç olarak, programlamanın amacı, bilgisayarın belirli işlemleri gerçekleştirmek veya istenen eylemi gerçekleştirmek için izlemesi gereken bir dizi talimat oluşturmaktır.

Program yazarken ve geliştirirken, programcı **programlama ortamını** oluşturan bir grup program kullanır. Bu programlar:

- **editör** - programın kaynak kodunun yazıldığı;
- **Derleyici - kompajler** - kaynak kodunu makine talimatına dönüştürür
- **hazır program kitaplığı** - program ortamında önceden yazılmış küçük ve sık kullanılan programların bir koleksiyonu veya toplamı;
- **hata ayıklayıcı - debager** - hata ayıklama için kullanılır.

En ilginç ve çekici olanı, **görsel bir ortamda programlamadır,** çünkü problem durumlarını çözerken, adımlar ve talimatlar oluştururken aşağıdaki gibi görsel nesneler kullanılır: programlamayı kolay ve eğlenceli hale getiren karakterler, arka planlar, sesler, hareketler, efektler vb.

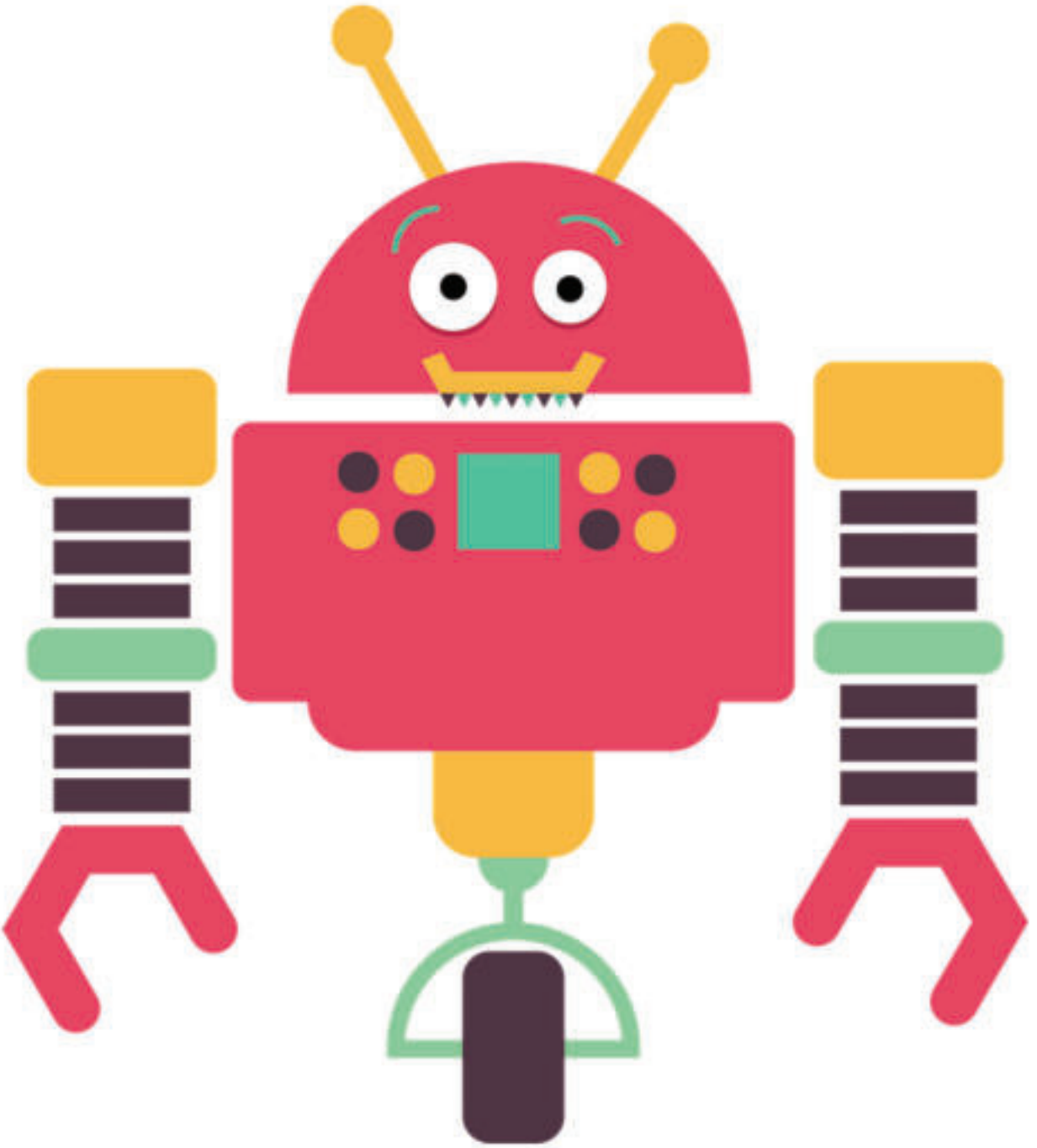
Programlamayı öğrenme sürecinde, görsel ortamda programlayı kullanan birçok uygulama bulunmaktadır. En yaygın kullanılan Scratch uygulamasıdır.

Scratch, ilginç hikayeler, animasyonlar, oyunlar, oluşturmak için kullanılan görsel bir programlama dilidir. Scratch programlama, figürler denilen nesnelerin yardımıyla projeler yapmaya dayanır. Komut seti bir yapbozu anımsatır, karmaşık programlama çözümleri oluşturmak, problemli ve algoritmik düşünme tarzına hakim olmak için çok çeşitli olanaklar sunar.

Bu konuda **Scratch programlama dilini,** Scratch arayüzünü ve ödevlere çözüm olarak proje etkinliklerinin nasıl oluşturulacağını öğreneceğiz.

Anahtar sözcükler!

grafiksel gösterim, koordinatlar, sabitler, değişkenler, ifadeler (komutlar), etkileşimli programlar, nesneler, olaylar (olaylar).



3.1 Grafik programlama. Scratch programına giriş

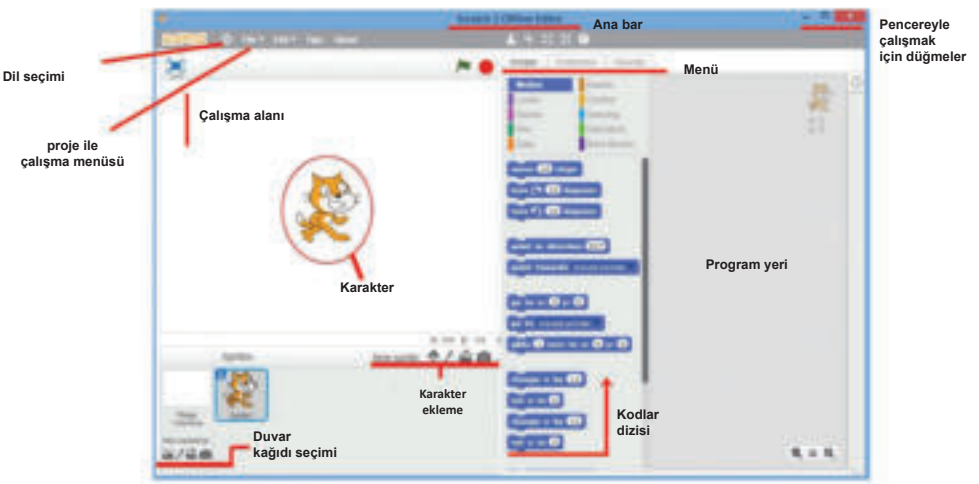
Görsel programlama, bir program oluşturmanın kavramını ve adımlarını öğrenmenin ilginç bir yoludur. Bu programlamanın ana özelliği, tüm araçların, şekillerin, komutların ve hatta kodların **grafiksel olarak görüntülenmesidir.** Bu arayüz, öğrencilerin en kolay ve en basit şekilde projeler oluşturmasına olanak tanır. Öğrenciler proje oluştururken yaratıcılıklarını geliştirir ve yaratıcı düşünür, sistematik olarak akıl yürütmeye ve birlikte çalışmaya başlar. Görsel bir programlama ortamında uygulanan en yaygın kullanılan programlama dili Scratch'tır.

Scratch, etkileşimli hikayeler, animasyonlar, oyunlar, müzik aletleri vb. oluşturmanıza olanak sağlayan bir programlama dilidir. Tüm araçlar animasyon efektleri, ses efektleri vb. grafiksel olarak uygun simgelerle temsil edildiğinden, öğrenciler programlama dilinin sözdizimini veya metin komutlarını öğrenmek zorunda kalmadan kod bloklarını sürükleyip birleştirebilir program oluşturabilir.

Scratch programlama görsel ortamı şu bağlantıdan çevrimiçi olarak kullanılabilir: https://scratch.mit.edu. Bu sayfa **Scratch**'in resmi sayfasıdır. Önce kullanıcının kullanıcı adı ve şifresiyle giriş yaptığı bir kullanıcı hesabı oluşturulur, ardından projelerin oluşturulmasına başlanır. Resmi siteden ücretsiz olarak indirilebilen çevrimdışı bir sürümü de vardır.

3.1.1 Scratch Başlatma

Scratch'in çevrimdışı sürümü, bilgisayarın **masaüstünde** bulunan simgesine çift tıklanarak başlar ve bu, iş için araçlara erişti̇ğimiz aşağıdaki pencereyi açar:



Şekil 1: Scratch ana ekranı

Uygulama arayüzünü ayrıntılı olarak anlatalım:

- başlık çubuğu programın adını gösterir;
- pencere çalışma düğmeleri projeyi küçültmenize, büyütmenize ve kapatmanıza olanak tanır;
- dil seçim aracı, kullanıcının menülerin ve komutların hangi dilde görüntüleneceğini seçmesine olanak tanır. Scratch, birçok dilde geliştirilmiştir;
- menü dosyası aracılığıyla kullanıcı proje ile ilgili aktiviteler gerçekleştirir: kaydetme, yeni bir proje açma, mevcut bir projeyi açma ve benzeri aktiviteler;
- menü çubuğundan projede kullanılacak nesnelerin siparişlerine ve düzenine erişiyoruz;
- projedeki karakter olan figürler masaüstüne yerleştirilir;
- şekil ekleme bölümü aracılığıyla, kullanıcı proje ödevinin bir parçası olacak karakterleri seçebilir;
- arka plan seçimi, proje ödevine uygun bir temanın uygulanmasına izin verir;
- Kodlar dizisi, kullanıcının seçebileceği komut veya deyim bloklarını içerir. Farklı işlevlere sahip oldukları için, yani farklı amaçlar için kullanıldıkları için farklı şekilde renklendirilirler. Kullanıcı, blokların-emirlerin hangi kategoriye ait olduğunu bilir;
- program konumu, kullanıcının bir etkinliği görüntülemek ve gerçekleştirmek için sürükleyerek seçtiği ve birleştirdiği bir dizi talimattır.

3.1.2 Scratch'in temel araçlarına giriş

Nesneler (Sprite)

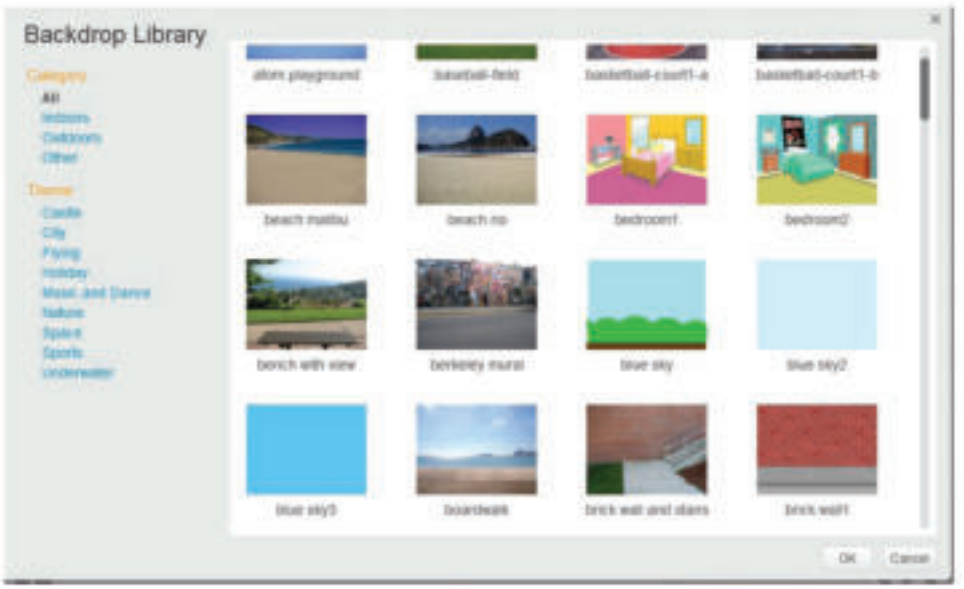
Kullanıcı bir program oluşturmaya başladığında, ekrandaki görsel ortamı başlattıktan sonra ilk olarak projede baskın rolü olan figürü veya karakteri fark eder, yani aktiviteleri gerçekleştirmek için programlar. Karakter başlangıçta Scratch logosudur, ancak resimde gösterildiği gibi Kitaplıktan Sprite Seç simgesi tıklanıp açılan galeriden yeni karakter seçilerek değiştirilebilir:

Resimden karakterlerin farklı kategorilere ayrıldığı görülmektedir. Kategoriye, ardından nesneye (**sprite**) ve son olarak da **Tamam** düğmesine tıklanarak bir nesne eklenir.

Galeri dışındaki nesneler bilgisayardan, kameradan veya kullanıcı tarafından oluşturulan bir karakterden eklenebilir.

Arka plan (Background)

Arka plan, karakterlerde olduğu gibi aynı şekilde ayarlanır. Duvar kağıdı değiştirme simgesine tıkladığınızda, resimde gösterildiği gibi bir duvar kağıdı teması seçim penceresi açılır:



Şekil 3: Duvar kağıdı galerisi

Duvar kağıdı temaları ayrıca bir bilgisayardan eklenebilir veya kendi başınıza oluşturabilirsiniz.

Emirler (İfadeler)

Programlamada ifadeler, bilgisayara ne yapacağını söyleyen komutlardır. Scratch'te, komut dosyaları bölümünde ifadeler bulunabilir:

Şekil 4: Scratch'teki ifade türleri (komutlar)

Resimde ifadelerin, yani emirlerin şu kategorilere ayrıldığını görüyoruz: hareket, görünüm, ses, kurşun kalem, olay, kontrol vb. Her olay veya sipariş kategorisi özel bir renkle işaretlenmiştir. Bu komutların kodları kategorilerin renkleri gibi aynı renktedir.

Bir program adımının oluşturulması gereken bloğu tıklayıp sürükleyerek ve blokları, yani adımları tek bir bütün halinde bağlayarak yapılır. Bu komutlar yerleştirildiğinde oluşturulan program başlatılır.

3.1.3 Scratch'te basit programlar oluşturmak

Scratch araçlarının yeteneklerini ve işlevselliğini anlamak için basit bir program oluşturacağız. Proje ödevinin teması dans etmektir. Başlayalım!

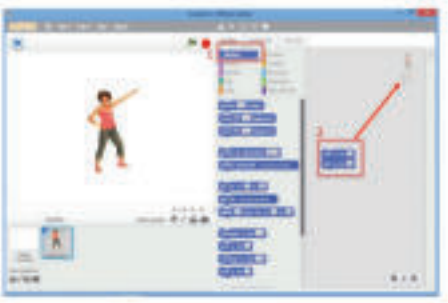
Önce Scratch programını başlatıyoruz ve bir karakter seçiyoruz. Bu proje etkinliği için Cassy Dance karakterini seçip masaüstünün ortasına yerleştiriyoruz. Masaüstünün ortasını nasıl buluruz?

Deneyin!

Bir matematik öğretmeninin yardımıyla koordinasyon sisteminin ne olduğunu tanımlamaya çalışın? Nasıl sunuluyor?

Scratch'teki masaüstü, x ve y eksenlerinden oluşan bir **koordinat sistemi** olarak temsil edilir. Koordinat sistemi, matematiksel bir ızgara veya **koordinat** adı verilen birçok noktadan oluşan bir model biçimindedir. Noktaların konumu aslında x ve y değerleri belirler. Masaüstünün ortası veya merkezi şu değere sahiptir: x = 0 ve y = 0. Merkezin dışındaki her noktanın özel bir x ve y değeri vardır. Aslında koordinatların değeri nesnenin yerini yani konumunu belirler.

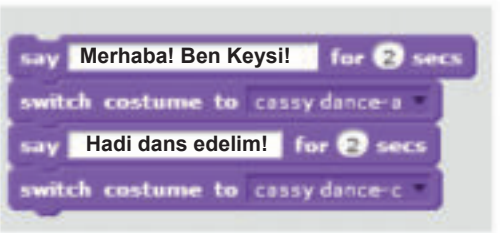
Cassy Dance karakteri projede bir nesnedir. Masaüstünün ortasına yerleştirmek için, **hareket - motion** menüsünden blokları seçiyoruz: **set x to 0 ve set y to 0** olarak ayarlayın, burada x ve y değerlerini 0 ile göstereceğiz:



Şekil 5: X ve y ye değerler verilmesi

Eksen değerlerini ayarlamak için bir sıraya sahip blokları seçerken, değerin belirlenmesi gereken yerin bir daire, yani giriş alanı şeklinde olduğunu fark edersiniz. Bu bize ihtiyaca göre değerin manuel olarak değiştirebileceğimizi söylüyor. Değerin ihtiyaca göre manuel olarak değiştirebiliriz. Bu nedenle, programlamada değerler için bu ayrılmış yerlere **değişken** denir.

Başlangıçta nesne, bizim durumumuzda Cassy Dance, giriş mesajları verebilir. Bunu, Scripts - Komut Dosyaları menüsü ve aşağıdaki komut bloklarını seçmek aracılığıyla yapıyoruz:



Şekil 6: Bir mesaj yazmak için komut seti

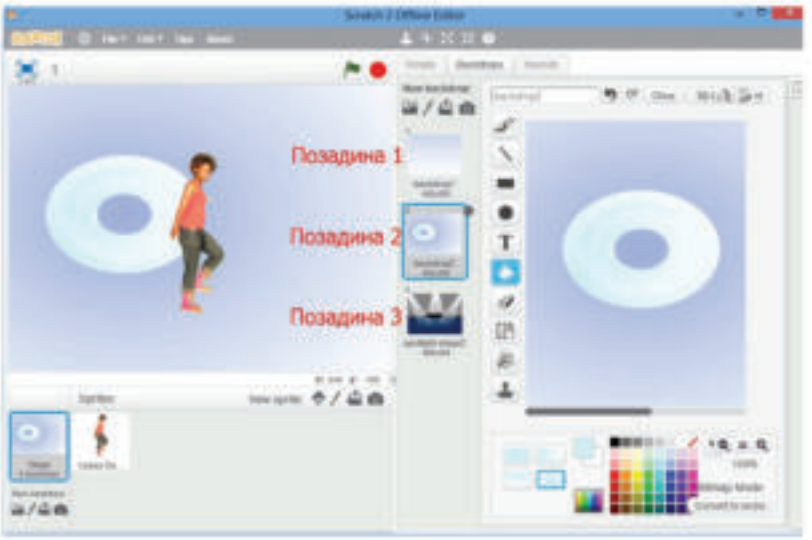
Koddan, bir mesaj için bir emir vermenin yanı sıra, nesnenin konumunda da değişiklikler yapıldı, böylece karakterin hareket ediyormuş gibi göründüğü fark edilebilir. Mor kodlar setinden önce, Events - Olaylar menüsünden komut setinin yürütülmesini başlatmak için, komutla aşağıdaki bloğu ekliyoruz:

Şekil 7: Komut dizisini başlatmak için bir başlangıç ekleme

Arka plan (Backdrop) ekleme prosedürü ile **Spotlight Stage 2**'yi seçiyoruz. Arka plan seçiliyken, menü çubuğunda **Backdrop - Arka Plan**'ı seçin ve aşağıdaki görünümü elde edin:

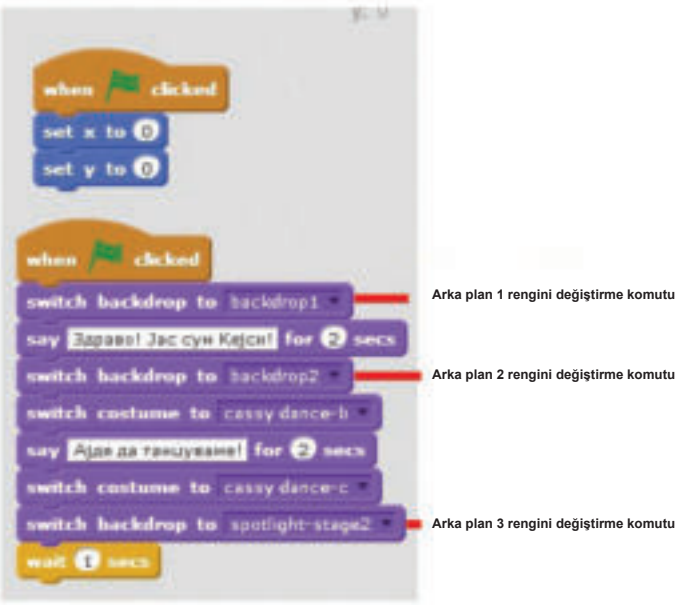
Şekil 8: Arka planda düzenleme

Sağ tıklayıp **çoğalt** seçeneğini kullanarak **arka plan 1'**in bir kopyasını oluşturun. Başka bir beyaz arka plan belirir. **Fill with color - Renkle doldur** aracını seçerek, pencerenin altındaki renk paletinden bir renk seçerek ve arka planı renklendirerek bu iki beyaz duvar kağıdını düzenleyebiliriz. Aynı şey **backdrop - arka plan 2** için de tekrarlıyoruz.



Şekil 9: Duvar kağıtları oluşturma ve düzenleme

Ardından, eklenen, oluşturulan ve düzenlenen duvar kağıtlarını proje aktivite koduna ekleyelim. **Look - Görünüm** menüsü aracılığıyla, arka planı değiştirmek için kod blokları ekliyoruz. Komutların sırası ektedir:

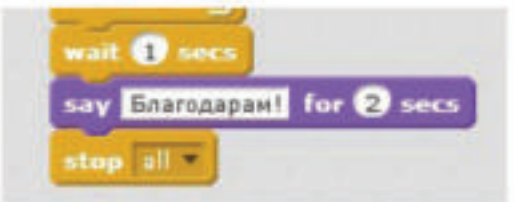


Şekil 10: Proje etkinliğinde arka planı dönüştürmek için komut seti

Nesneyi sahneye yerleştirmek için, hareket menüsünden koordinat değerleri ekleriz, komutlar x'i 15'e ve y'yi 100'e ayarlarız. Bu şekilde yerleştirilen obje dans gösterecek ve müzik eşliğinde olacak şekilde programlanacaktır. Blok komutları, yani eklenmesi gereken ifadeler aşağıdaki sıradadır:

Şekil 11: Tekrarlamak için döngüleri ekleme

Kodu tamamlamak için bir dizi talimat ekliyoruz. Örneğin, proje ödevimizi bir mesajla tamamlayacağız ve tüm süreçleri durduracağız:



Şekil 12: Programın tamamlanması

Bununla ilk projemizi tamamladık. Hadi bir bakalım!
Çalışma penceresinin sol tarafında, nesnelerin görüntülendiği bölümde, sağ üst köşede, projeyi başlatmak ve bitirmek için simgeler bulunur:

Şekil 12: Programın tamamlanması

Yeşil bayrağa tıklayarak, ne yaptığımızı görsel olarak gördüğümüz talimat setini uygulamaya başlarız, eğer ekranı durdurmak istersek, kırmızı durdur düğmesine tıklarız.

Ezberle!

Grafik programlama uygulamaları, grafik bir ekrana sahip, araçları, şekilleri, komutları vb. kullanarak bir programı yürütmek için komut dizilerinin oluşturulmasını sağlar. Scratch, etkileşimli hikayeler, animasyonlar, oyunlar oluşturmanıza izin veren bir programlama dilidir. Scratch komutları, yap-boz gibi bulmacaya benzeyen bloklar halinde görüntülenir. Sürükle ve bırak ile birleştirilirler. Scratch çalışma alanı, matematiksel bir ızgara veya noktalı bir desen olarak temsil edilebilir. Scartch projelerine eklenen nesneler, atanmış ifade bloklarıdır (komutlar). İfadeler (komutlar) bilgisayara ne yapması gerektiğini söyler. Bazı ifadelerde değerler için ayrılmış yerler var ve bunlara değişken adı verilir.

Sorular

1. Scartch'taki bloklar ne anlama geliyor? Neye benziyorlar?
2. Blok komutları nasıl kategorize edilmiştir? Birkaç kategori yazın!
3. Proje görünümünün üst kısmındaki yeşil bayrak ne anlama geliyor?
4. Proje görevindeki nesnenin koordinatları bize neyi gösterir?
5. Bir proje setinin arka planı nasıl ayarlanır?

3.2 Etkileşimli olay programları

Hatırlayalım!

"Etkileşimli programlar" teriminden ilk bahsettiğimiz zamanı hatırlıyor musunuz? Hangi konularda? Nasıl tanımladığımızı hatırlıyor musun?

"**Etkileşimli**" terimi, farklı konularda ve farklı alanlarda defalarca bahsedilmiştir. Örneğin, bilgisayar bilimleri dersini **etkileşimli** olarak nitelendirebiliriz çünkü etkinliklerin gerçekleştirilmesi için etkinlikler ve iletişim vardır. Sınıflarda genellikle ödevler ve problem durumlarını çözmek için tartışmalar yapılır, bu da hepimizin yüksek sesle düşündüğümüz, sonuçlar verdiğimiz, görüş verdiğimiz ve aynı zamanda birçok soruya cevap verdiğimiz anlamına gelir.

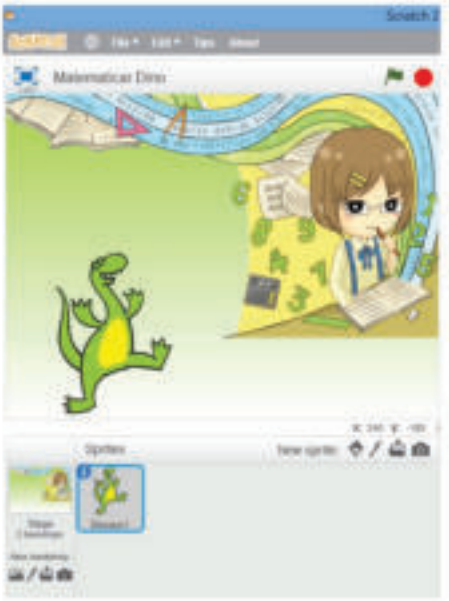
Aynı şey eğitim yazılımı kullandığımızda da oluyor. Program bize talimatlar verir, biz kullanıcılar talimatları izler ve buna göre uygun eylemleri gerçekleştiririz.

Bilgisayar oyunları söz konusu olduğunda, oyunun kurallarına uymak aslında kullanıcı ile oyunun kendisi arasındaki etkileşimi temsil eder. Her kullanıcı hedefe ulaşmak ister, yani kazanmak ister ve bu nedenle oyunun kurallarına göre hareket etmelidir.
Bu öğretim ünitesinde **etkileşimli etkinlikler** içeren bir proje oluşturacağız.

İnteraktif oyunumuzdaki ana karakter Dinoaur1'dir. Vereceğimiz görevlerin matematiksel hesaplamalarını yapacak. Başlangıç olarak, girilen iki sayının bölümünü hesaplayacak blok komut dizileri oluşturacağız.

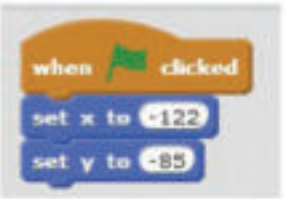
Programlamaya başlayalım!

İlk olarak Scratch programını başlatıyoruz. Araçlardan makas yardımıyla (**delete-sil)** nesneyi, yani **Cat** karakterini siliyoruz. Nesne ekleme aracı aracılığıyla, **spirit** bölümünde mevcut galeriden bir nesne ekliyoruz. Projemiz için Dinoaur1 nesnesini seçeceğiz. Ardından arka planları (**backdrop**) ekliyoruz. Programın sunduğu araçlar ve renk paleti yardımıyla kendimiz bir duvar kağıdı oluşturup ve düzenleyeceğiz. İkinci duvar kağıdı ise bilgisayarımızdan bir resim olacaktır. Ana ekran şöyle görünecektir:



Şekil 1: Projeye nesneler ve arka plan ekleme

Dinoaur 1'e tıklayın ve blok komutları dizisini oluşturmaya başlayın:



Şekil 2: Nesnenin yerini belirleme

Belirtilen blok komutları dizisi, komut dizisinin yürütülmesinin başlangıcını gösteren yeşil bayrağa tıklayarak, **x** ve **y** değerinin **-122** ve **-85** olacağı anlamına gelir, bu da **Dinoaur1'in** konumunu belirlediğimiz anlamına gelir.

Aşağıdaki blok kodları, projedeki ilk olaylardır:

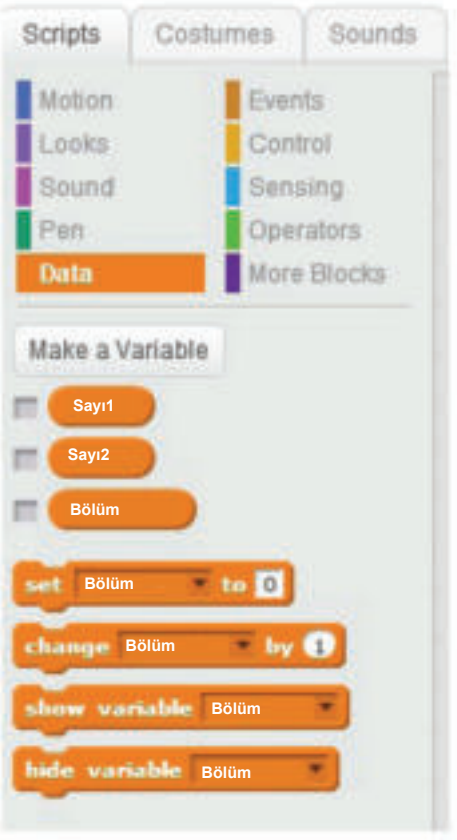
Şekil 3: Mesaj yayınlayan komutlar dizisi

Bu, yeşil bayrağa tıkladığımızda, blok komutlarının sırasının yürütüleceği anlamına gelir. Her şeyden önce, Dinoaur 1 bizi "M**erhaba matematikçiler"** olarak selamlıyor, hareket ediyor, yani masaüstünün ortasına "**kayıyor**". Dinoaur 1 masaüstünün ortasına geldiğinde düşünür gibi görüntülenir, rengini ve hareketli görüntüsünü değiştiriyor.

Sonra, bizi yeni bir duruma götürecek sıralı bloklarla devam ediyoruz. **Dinoaur1** talimatları verecek ve gerekli verileri klavyeden girdiğiz. Aslında, **Dinoaur1** bizim matematikçimizdir ve bize girilen iki sayının bölümünün hesaplanmasının sonucunu söyleyecektir. Bu amaçla, öncelikle klavyeden girilen değerlere ve hesaplamanın sonucuna sahip olacak üç **değişken** tanımlayacağız. Değişken oluşturma prosedürü aşağıdaki gibidir: **Scrits-Komut** Dosyaları menüsünden **Data-Veri** kategorisine erişiyoruz ve orada **Make variable - Değişken oluştur**'a tıklıyoruz. Bu değişken proje etkinliğinde diğer durumlarda kullanacağımız için seçeneğe tıklıyoruz ve değişkene isim veriyoruz.

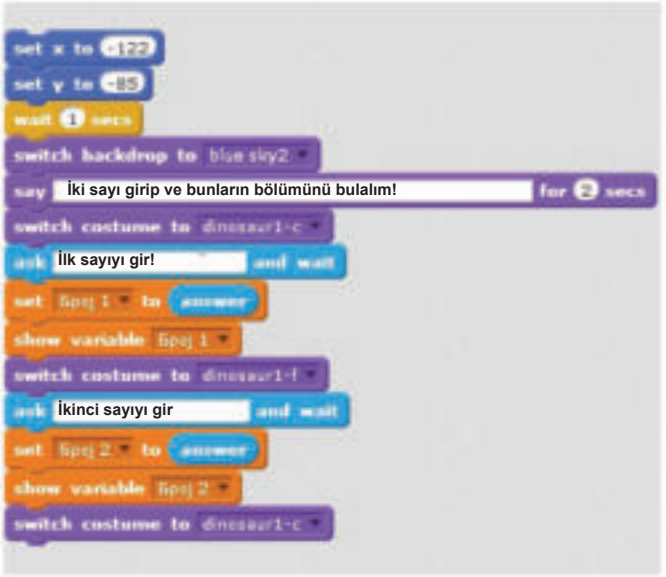
Not!

Değişkenin adını, istediğiniz gibi yapabilirsiniz, ancak değişken ismi hangi değere sahip olacağını ve rolün ne olacağını hatırlatması sizin için önemlidir. Örneğin, ödevde değişkenler Sayı1 ve Sayı2 adlarıyla tanımladık, ancak bunlar matematiksel adlara göre bölenler ve bölünenler olarak da tanımlanabilirler.



Şekil 4: Değişkenler oluşturma

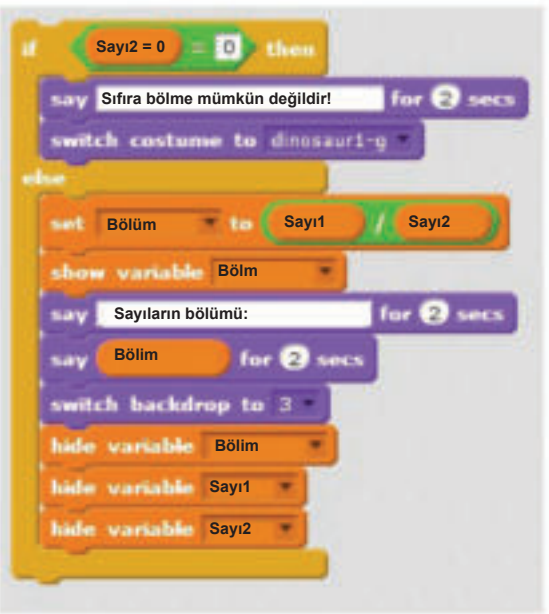
Bu projede **üç değişken** oluşturuyoruz. İlk değişken, birinci sayı (**sayı1**) için klavyeden girdiğimiz değeri, ikinci değişken ikinci sayı (sayı2) için girilen değeri ve üçüncü değişken (**bölüm**) ise çözümden kaynaklanan değeri gösterecektir. Oluşturulan değişkenleri buna göre uygulayalım!



Şekil 5: Veri kategorisindeki blok komutlarının uygulanması

Motion - Hareket menüsü aracılığıyla **Dionaur1**'i orijinal konumuna, yani başlangıç konumuna döndürmek için x ve y'ye değerler veriyoruz. **Looks-Görünümler** menüsü aracılığıyla, arka planı (backdrop) değiştirmek için bir blok komutu ve bir mesaj görüntülemek için bir blok komutu veriyoruz. **Dinoaur1** görünümünü değiştirir ve bize ilk sayıyı girmemizi söyler. Dinoaur1 görünümünü değiştirir ve bize ilk sayıyı girmemizi söyler. Blok komutu burada uygulanır, bu da bize klavyeden bir sayı girme ve girilen değerle alanı görüntüleme seçeneği sunar. Resimde gösterildiği gibi ikinci sayı için de aynısını tekrarlıyoruz.

Yapmamız gereken bir sonraki şey, ikinci sayının değerinin 0 olup olmadığını kontrol etmektir, çünkü matematik kuralı sıfıra bölmenin mümkün olmadığını söylüyor. Bu amaçla, **Kontrol** menüsü aracılığıyla blok komutunu uygularız: "koşul yerine getirilirse, aşağıdaki talimatları uygulayın" ve "koşul yerine getirilmezse, aşağıdaki talimatları uygulayın", yani:

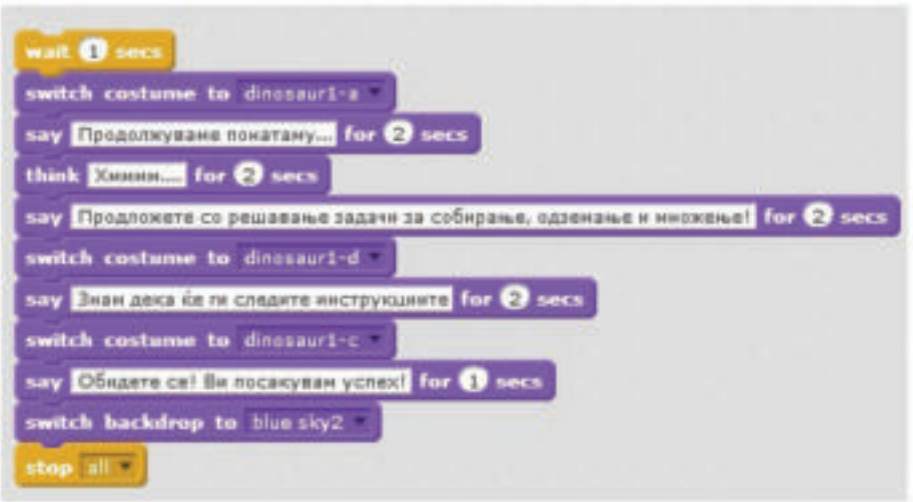


Şekil 6: If / else blok komutunun uygulanması

Ya da, ikinci sayının değeri sıfıra eşitse, şu mesajı görüntüleyin: “sıfıra bölme mümkün değildir” ve **Dinoaur1** görüntüsünü değiştirir. Koşul doğru değilse, o zaman:

- bölümün hesaplaması yapılsın;
- bölümün değeri görüntülensin;
- bölümü **Dinoaur1’**e mesajı olarak görüntülensin;
- yeni bir konuya, yani arka plana geçmek;
- yeni arka plana geçildiğinde, değişkenler görünmesin.

Bununla her iki sayının bölümünün hesaplamasını tamamladık. Proje istek üzerine sonlandırılabilir. Örneğin, bizim durumumuzda Dinoaur1 mesajlarla devam ediyor:



Şekil 7: Mesajları görüntülemek için blok komutları

Daha önce ekranda mesajları görüntülemek için prosedürü kullandık. Mesajlar, kullanıcının yaratıcılığına ve proje ödevinin çözümünü görsel olarak nasıl hayal ettiğine bağlıdır. “**stop all**” komutu ile, komut dizisinin yürütülmesinin sonunu işaretleriz.

Yeşil bayrağa tıklayarak blok komut dizilerini çalıştırmaya başlıyoruz ve böylece ne yaptığımızı kontrol ediyoruz. Bu proje etkileşimlidir, çünkü **Dinoaur1** talimatlar verir, bize nasıl hareket edeceğimizi söyler ve biz bu talimatları klavyeden harekete geçerek yürütürüz.

Ödev

Dinoaur1, girilen iki sayının bölümünü bulmamıza yardımcı oldu. İki sayıyı toplama, iki sayıyı çıkarma ve çarpım gibi diğer matematiksel işlemleri hesaplayacak blok deyimleri dizileri oluşturarak ödeve devam edin.

Düşün!

Üç sayı ile matematiksel işlemlerin hesaplamalarını yapacak bir dizi blok komutları oluşturmak mümkün müdür? İki sayı girerek hesaplamalar yapmak için oluşturduğumuz Dinoaur1 programında eksik olan nedir? Prosedürü açıklayın! Nerede değişiklik yapacağınızı gösterin!

Ezberle!

Kullanıcı, değişkenlerin değerlerini girebileceği ve sonuçları görüntüleyebilen programlara etkileşimli programlar denir. Değişkenler, klavyeden veri alabilir veya bir hesaplama sonucunda değer alabilir. Değişkenler, Scripts - Komut Dosyaları menüsü ve Data - Veri kategorisi aracılığıyla oluşturulur.

Sorular

1. Hangi programlara etkileşimli denir?
2. Değişkenler nelerdir? Değişken oluşturma prosedürü nedir?
3. Hangi komut değişkenin değerini girmenize izin verir?
4. Kullanıcı programa değişkenin değerini nasıl giriyor?
5. If ... Else komutunun amacı nedir?

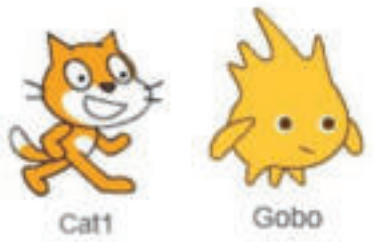
3.3 Daha karmaşık sorunlara programların geliştirilmesi

Scratch'te bir oyun oluşturmak, birden fazla **nesne** ve **olay** içerdiğinden biraz daha karmaşıktır. Her bir nesneye, hedefe, yani oyundaki son zafere ulaşmak için talimatlar verilir. Ayrıca meydana gelebilecek olaylar önceden tahmin edilmelidir. Elbette kendilerine uygun çözümlerin sunulması beklenmiyor. Örneğin, kazanırsanız, bir mesaj alırsanız veya kazanan bir ses duyarsanız ve kaybedersenız nesneyi başa döndürülsün.

Örneğin, iki nesnesi olan bir oyun oluşturacağız: **Cat** ve **Gobo**. Kedi farenin hareketine göre - sola ve sağa - hareket edecektir. Gökten daha fazla **Gobo** düşer ve **Cat** toplam hedeye sayısının çoğunu toplaması gerekir.

Scratch'te yeni bir projeye başlayalım.

Kütüphaneden - library Sprite Seç düğmesine tıklayarak yeni bir nesne / karakter ekliyoruz. Nesnelerin / karakterlerin bulunduğu galeriden Cat ve Gobo'yu seçiyoruz:



Şekil 1: Bir projedeki nesneler / karakterler

Küçültme aracını kullanarak nesnelerin / karakterlerin boyutlarını değiştiririz. Küçültme aracına tıklıyoruz ve ardından nesneyi, yani karaktere, nesneyi küçültmek istediğimiz kadar tıklıyoruz. Bizim durumumuzda iki nesnenin boyutlarını azaltacağız:

Şekil 2: Boyutları Küçültme aracıyla değiştirme

Library - Kitaplık aracından **choose backdrop** tan bir tema veya arka plan ekliyoruz. Oyunumuz için **Gingerbread** temasını kullanacağız:

Şekil 3: Proje teması

Daha sonra, nesnelerin canlandırılmasını ve bazı eylemleri gerçekleştirmesini sağlamak için her nesneye talimat kümeleri veya dizileri atanmalıdır. Örneğin, **Sprite1** nesnesi sola ve sağa hareket etmeli ve **Gobo'yu** toplamalıdır:

Şekil 4: Sprite 1 hareket talimatları seti

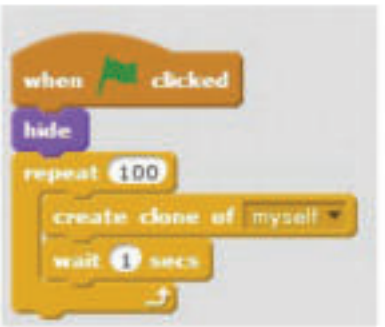
Not!

Listelenen blok komutlarından, farklı renklerde sunulan blok komutları olduğunu fark ediyoruz ve bu bize onların farklı bir komut kategorisine ait olduklarını gösteriyor.

Blok blok komutları açıklayalım!

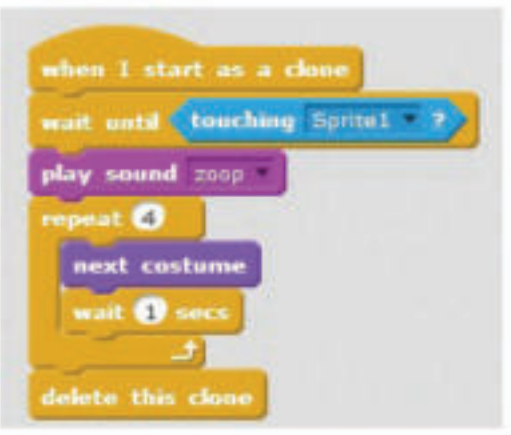
Yeşil bayrağa tıklamak, aşağıdaki talimatları yürütme sürecini başlatacaktır. Böylece x ve y koordinatlarına 0 ve -130 değerler veriyoruz. Hareket yönünü **90 sağa** doğru belirliyoruz. Nesne / karakter, farenin hareketine göre **x ekseni** boyunca hareket edecektir. Ayrıca, yeşil bayrağın tıklanmasıyla bir **ses dosyası** çalmaya başlayacaktır. Bizim durumumuzda, dans **slow mo** ses dosyası galeriden seçildi.

Ayrıca nesne / karakter **Gobo'ya** talimat vermemiz gerekiyor. Ödevimize göre, daha fazla **Gobo** örneği **veya klonu** gökten düşmeli. Sadece bir nesne - hediye tanımlamış olsak da, oyunda bu nesne sayılacaktır. Prosedürü görelim:



Şekil 5: Bir nesnenin klonlarını oluşturma

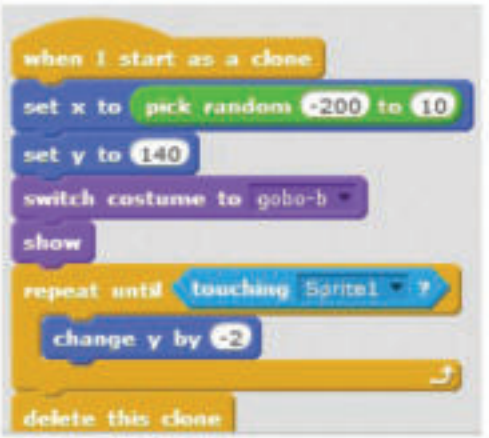
Bu komut dizisi veya emirler, yeşil bayrağa tıklamanın, bir klon, yani nesnenin / karakterin bir kopyasını oluşturmak için komut dizisinin yürütülmesinin başlangıcını işaret ettiği anlamına gelir. Bloku tekrarla komutu, 100 değerine ulaşılana kadar her saniye klonlar oluşturulmasına izin verir.
Yapılması gereken bir sonraki şey, nesneye söyleyecek komutlara sahip bloklar oluşturmaktır - nasıl davranılacağını ve hangi eylemlerin gerçekleştirileceğini klonlayın:



Şekil 6: Nesne için talimatlar - klon

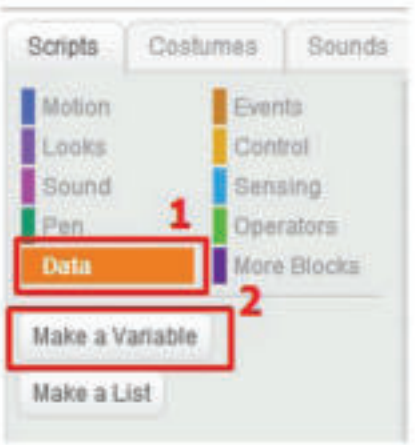
Şu anlama gelir: Klon nesnesi göründüğünde, **Spirit1** nesnesine kadar aşağı kaydırılacaktır. **Spirit1**'e dokunduğunda, **zoop** adlı bir müzik dosyasını çalmaya başlayacaktır. Döngüsel tekrar komutu, her sanye **Gobo** nesnesinin dört farklı şeklini değiştirecektir. Bundan hemen sonra nesne klonu silinecek, yani kaybolacaktır.

Sonraki talimat seti, klon nesnesinin **x eksenine** göre -200 ila 200 aralığında rastgele koordinatlarda görüneceğini ve **y eksenine** göre 140'tan başlayarak aşağı doğru hareket edeceğini belirtir. Sprit1 nesnesine dokunana kadar kendini değiştirecektir. Sonunda klon nesnesi kaybolacak.



Şekil 7: Nesnenin - klonun görünümü için gelişigüzel koordinatların ayarlanması

Oyunda puanları eksiktir. Sonraki komut dizisiyle saymayı etkinleştireceğiz. **Gobo**, **Spirit 1** nesnesine dokunduğunda puanlar değişecektir. İlk olarak, başlangıçta 0 değerine sahip olacak yeni bir değişken eklememiz ve ardından 1 arttırılması gerekir. Bir değişken eklemek, Scripts - Komut Dosyaları menüsünden **Data - Veri s**eçeneğine ve ardından **Make variable - Değişken oluştur** seçeneğine tıklanarak yapılır:

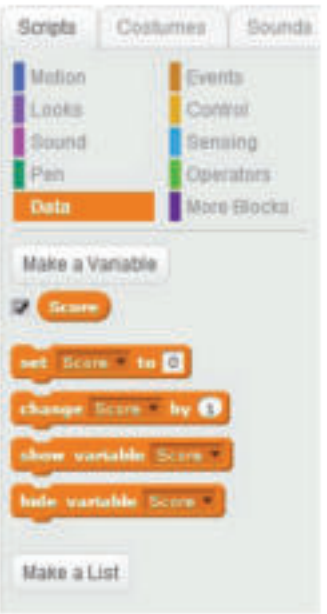


Şekil 8: Yeni bir değişken oluşturma

Kullanıcının değişken adlandırdığı ve oluşturulan değişkenin seçilen karakter için kullanılacağını belirten aşağıdaki pencere görünür:

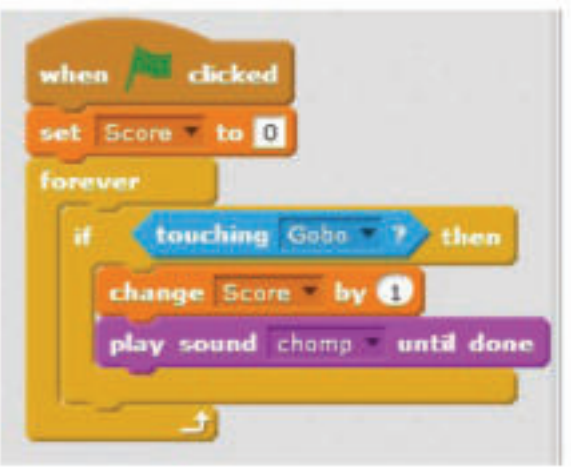
Şekil 9: Yeni bir değişken tanımlama

Tamam'a tıklamak, yeni değişkenle ilgili aşağıdaki blok komutlarını ekler:



Şekil 10: Yeni Puan – Score değişkeniyle ilgili blok komutlar

Değişkeni oluşturduktan sonra, puanları saymak için bir dizi blok komutu oluştururuz:



Şekil 11: Noktaları sayma

Oyunun nasıl çalıştığını görelim! Yeşil bayrağa tıklıyor ve oynamaya başlıyoruz.

Ödev

Scratch'te basit bir oyun oluşturun. Oluşturduğunuz oyunları arkadaşlarınızla paylaşın ve talimatları izleyerek oynayın.

Ezberle!

Birden çok nesne ve birden çok olay içeren projeler oluşturmak daha karmaşıktır. Her nesne ayrı ayrı programlanır, yani bir hareket, aktivite veya eylem oluşturmak için yürütülmesi gereken blok komutları dizileri verilir. Programlarda nesneler durum ve davranış tarzına göre tanımlanır. Gerçek dünyada nesneler şunlar olabilir: bir araba, bisiklet, top veya bir durum ve davranışla tanımlanan bir karakter. Olaylar, nesnenin durumuna veya davranışına bağlı olarak meydana gelir. Olaylar öngörülmeli ve onlar için uygun bir çözüm önerilmelidir.

Sorular

1. Nesne nedir?
2. Nesneler nasıl programlanır?
3. Nesneler ve olaylar arasındaki bağlantı nedir?

TEKRAR EDELİM! UYGULAYALIM!

Ödev 1

Derste pratik olarak üzerinde çalıştığımız ifade bloklarını (komutlar) uygulayarak Scratch'te bir proje ödevi oluşturmaya çalışın. Duvar kağıdı ekleyin, nesne ekleyin, nesneyi taşıyın, ses ekleyin vb.

Ödev 2

Scratch'te birden çok nesne ve olay içeren basit bir oyun oluşturun.

Ödev 3

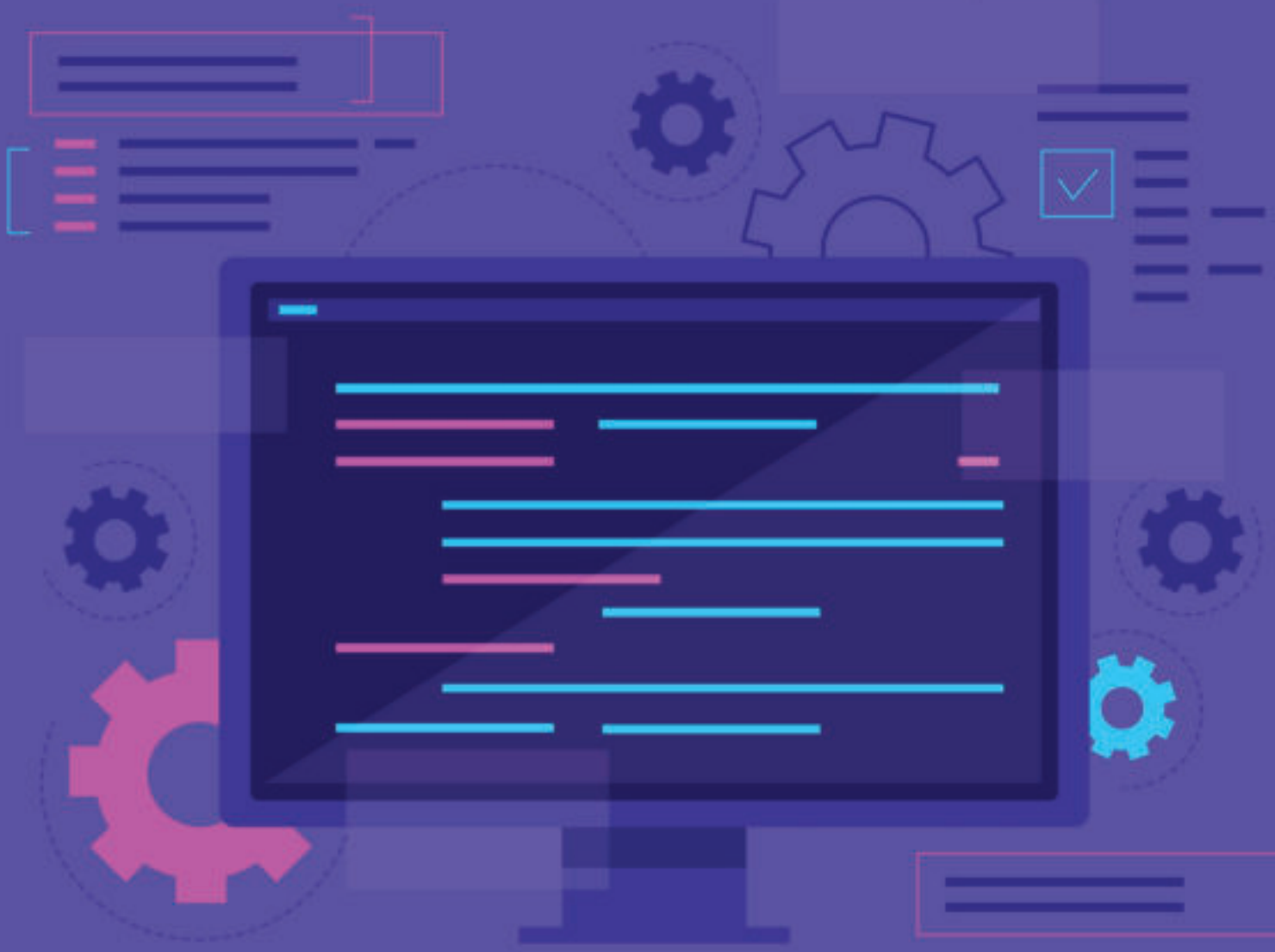
Bir kare, dikdörtgen ve dairenin alanını ve çevresini hesaplayacak programlar oluşturun.

Ödev 4

Scratch'te, klavyede bir sayı girerken pozitif mi yoksa negatif mi olduğunu kontrol edecek etkileşimli bir proje oluşturun.

Standart bir yapısal programlama dili aracılığıyla programlama

Standart bir yapısal programlama dili aracılığıyla programlamaya giriş
Bir program yapma süreci
Entegre bir programlama ortamının temel unsurlarına giriş
Hazır örnek program kodlarının görünümü
Hazır örnek programların yürütülmesi
C ++ programlama dilinin temel öğeleri
İfadeler
Programların geliştirilmesi
Aritmetik işlemler ve ifadeler
Sabitler ve değişkenler
Programa veri girmek için ifadeler (teknikler)
Karşılaştırmalı İfadeler
Bir koşul karşılanana kadar yapıyı bir döngüde tekrarlamak
TEKRAR EDELİM! UYGULAYALIM!



4 Standart bir yapısal programlama dili aracılığıyla programlamaya giriş

Ne öğreneceğiz?

Code :: Blocks entegre bir programlama ortamında çalışmak ve C++ programlama dilinde bir program oluşturmak.

Toplumumuzun her yerinde hızlı bir değişim ve hatta daha hızlı teknolojik gelişme çağında yaşıyoruz. Bilgisayar kullanılmadan yapılabilecek herhangi bir işi hayal etmek neredeyse imkansızdır. Bir bilgisayarın, üzerine kurulu programlar yardımıyla veri toplamak, saklamak ve işlemek için kullanılan elektronik bir cihaz olduğu gerçeği göz önüne alındığında, herhangi bir problem durumuna, farklı amaçlarla ve farklı alanlarda çözüm sunan en güçlü araç olmayacağı sonucuna vardık.

Günlük yaşamımızda aşağıdakiler gibi farklı programlar kullanırız:

- eğlenceli oyunlar oynarız;
- bir kelime işlemci programında (MS Word, Open Office Writer, vb.) makaleler yazıyoruz;
- elektronik tablolama programında (MS Excel, Open Office Calc, vb.) hesaplamalar yapıyoruz;
- multimedya sunumları oluşturmak ve düzenlemek için (MS Power Point, Open Office Impress, Prezi, vb.) programda sunumlar oluşturuyoruz;
- video dosyalarını uygun programlar (Winamp, VLC Player, Real Player, vb.) aracılığıyla izliyoruz;
- grafik düzenleyicilerde (MS Paint, InDesign, Corel Draw, Photoshop, vb.) görüntüler oluşturmak ve düzenlemeler;
- akraba ve arkadaşlarla iletişim programları (Viber, Messenger, WhatsApp vb.) aracılığıyla iletişim kuruyoruz.

Kullanıcının ihtiyaçlarına göre oluşturulmasını sipariş edebileceği özel amaçlar için başka programlar da vardır. Aslında programlar, kullanıcı ve bilgisayar arasındaki bağlantıdır. Bu nedenle, bilgisayar programları, bir bilgisayarın yürütmesi gereken bir dizi talimattır. Talimat dizilerini tasarlama, oluşturma ve bağlama sürecine programlama denir. Programlama dillerini kullanarak programlar oluşturan kişilere programcı denir.

Programlama dilleri, bir ödevi veya problem durumunu çözmek için farklı şekillerde bir araya gelen bir dizi kural, simge ve anahtar sözcüktür. Programlamanın öğrenilmesi, günlük durumlarda kullanışlı ve uygulanabilir olan mantıksal bir düşünme şekli geliştirir.

Anahtar terimler!

çevirmen, kaynak program, çalıştırılabilir program, editör, derleyici, hata ayıklayıcı, ifade, karşılaştırmalı ifade, döngü.

4.1 Bir program yapma süreci

Bir bilgisayar, telefon, tablet veya başka bir cihaz için olsun, bir program oluşturmak karmaşık bir süreçtir, çünkü oluşturulmasına yönelik adımlar bir programlama dilinde yazılmıştır. onu oluşturma adımları bir programlama dilinde yazılmıştır. Birçok programlama dili vardır ve her birinin kendine özgü bir dizi talimat ve yazım sözdizimi vardır. **Programlama dilleri** genellikle daha **yüksek seviyeli** ve daha **düşük seviyeli** programlama dillerine bölünmüştür.

Düşük seviyeli programlama dilleri makine yönelimlidir çünkü çalıştırıldıkları makineye bağlıdırlar, yani bu program bir işlemciye yazılırsa başka bir işlemcide çalışmayabilir. Bu nedenle, makine kodunda programlama yapmak zordur çünkü bilgisayarın yapısını iyi bilmeniz gerekir.

Düşük seviyeli programlama dillerinin aksine, **yüksek seviyeli programlama dilleri,** kullanılacak makineye bağlı olmadıkları için giderek daha fazla kullanılmaktadır, bu da farklı işlemcilere sahip bilgisayarlarda gerçekleştirilebilecekleri anlamına gelmektedir. Öte yandan, sembolleri, işaretleri ve aynı zamanda sözdizimleri doğal dile çok benzer ve anlaşılması kolaydır. En yaygın kullanılan yüksek programlama dilleri: **C, C ++, Fortran, Basic, Pascal, Java** ve diğerleri. Bir program oluştururken hangi seviye programlama dilini kullanırsak kullanalım, süreç aşağıdaki aşamalardan geçer:

Sıra No	Aşama	Anlam
1	Kaynak kodu yazma	Kaynak kodu yazmak, entegre bir programlama ortamında bir programlama dili ile yapılır. Bu program, bilgisayarın bellek cihazlarında .cpp uzantılı bir dosya olarak saklanan kaynak kodu (Source Code) olarak adlandırılır, örneğin: ödev.cpp
2	Kaynak kodun çevirisi	Kaynak kodu, derleyiciler adı verilen derleyiciler tarafından çevrilir. Kaynak kodu çevrilirken hatalar tespit edilebilir. Bu hatalara sözdizimi hataları denir ve genellikle programlama dilinden yanlış yazılmış sözcüklere, yanlış yazılmış veya unutulmuş noktalama işaretleri sebep olabilir. Çeviri yaparken, .obj uzantılı bir nesne kodu dosyası alırsınız, örneğin: odev.obj
3	Bağlantı kaynak kodu	Bağlayıcı (Linkers) adı verilen programların yardımıyla, nesne kodları çalıştırılabilir koda (Executable code) bağlanır. Çalıştırılabilir kod bilgisayar tarafından yürütülür. Böylece, .exe uzantısına sahip çalıştırılabilir bir dosya oluşturulur, örneğin: odev.obj
4	Programın test edilmesi	Programın test edilmesi, programın belirlenen koşulları karşılayıp karşılamadığını ve doğru çözüme ulaşıp ulaşmadığını kontrol etmeye hizmet eder, yani ürünün kalitesi ve olası hatalar hakkında bilgi veren bir araştırma türüdür. Bu aşama aynı zamanda programın geçerliliği ve doğrulanması sürecidir, yani dokümantasyonda belirtilen gereksinimleri karşılar, beklendiği gibi çalışır ve tüm özellikleri ile uygulanabilir.

Tablo 1: Program yazmanın aşamaları

Deneyin!

Çevrimdışı görsel program Scratch'i hatırlayın. Kodu nerede ve nasıl yazdığınızı, nasıl derlendiğini ve programları oluştururken hataların nasıl bulunduğunu karşılaştırın. Benzerlik ve farklılıklara dikkat edin.

Bu tematik birimdeki program oluşturma sürecinde, Code::Block entegre programlama ortamında C++ programlama dili ile programlar oluşturacağız.

Ezberle!

Bilgisayar programları, bir bilgisayarın yürüttüğü bir dizi talimattır. Bir program oluşturma sürecine programlama denir. Programlama dillerini kullanarak programlar oluşturan kişilere programcı denir. Programlama dilleri alt ve üst seviye olmak üzere ikiye ayrılır. Bir program oluşturmanın aşamaları şunlardır: kaynak kodu yazmak, kaynak kodu çevirmek, çalıştırılabilir koda bağlanmak, programı test etmek ve doğrulamak.

Sorular

1. Program oluşturma prosedürünün adı nedir?
2. Programları kim oluşturur? Nasıl?
3. Programlama dilleri nasıl bölünür?
4. Alt seviye programlama dillerini yazınız?
5. Üst seviye programlama dillerini yazınız?
6. Üst seviye ve Alt seviye programlama dilleri arasındaki fark nedir?
7. Bir programlama dilinin sözdiziminin ne olduğunu tanımlayın!
8. Kaynak program nedir? Kaynak program dosyası hangi uzantıya sahiptir?
9. Bilgisayarın çalıştırdığı programın adı nedir?
10. Bir program oluşturmanın aşamalarını listeleyin!

4.2 Entegre bir programlama ortamının temel unsurlarına giriş

Entegre Geliştirme Ortamı (IDE - Integrated Development Environment), yazılımı yazmak ve test etmek için gereken temel araçları içeren bir yazılım paketidir. Bu uygulama tipik olarak yazılım yazmak, değiştirmek, derlemek, dağıtmak ve hata ayıklamak için birçok işlev sunar. Bu araç setinin temel amacı, yazılım geliştirmeyi basitleştirmek ve kodlama hatalarını en aza indirmektir.

Bir program oluştururken, kullanıcılar yazılım kodunu oluşturmak ve test etmek için bir dizi araç kullanır. Geliştirme araçları genellikle kaynak kodu düzenleyicileri, kod kitaplıkları, derleyiciler ve test platformlarını içerir. Aslında, programcı **kaynak kodunu kod düzenleyicide yazar ve düzenler, derleyici kaynak kodunu bilgisayarın çalıştırabileceği okunabilir bir dile çevirir** ve hata ayıklayıcı herhangi bir sorunu veya hatayı çözmek için yazılımı test eder.

Deneyin!

Günlük okul sorumluluklarında bir problem durumu hayal edin. Bütünleşik bir programlama ortamı olarak düşünmeye çalışın: çözüm adımları oluşturun, bu adımları derleyin, böylece arkadaşlarınız ek olası unsurları anlayıp onlarla konuşun, bunun doğru çözüm olup olmadığını bulmaya çalışın.

C ++ programlama dili ile program oluştururken kullanacağımız bir entegre programlama ortamı **Code :: Blocks**'tur.

Code :: Blocks, birden çok programlama dilinin derlenmesini ve test edilmesini destekleyen entegre bir programlama ortamıdır. Bir dizi derleyiciyle çalışır. Açık kaynak C ++ programlama dilinde ücretsiz bir yazılım geliştirme araç setidir ve aşağıdaki araçları içerir: Metin düzenleyici, çeviri ve bağlantı yazılımı ve hata algılama yazılımı.

4.2.1 Code :: Blocks'un Yüklenmesi

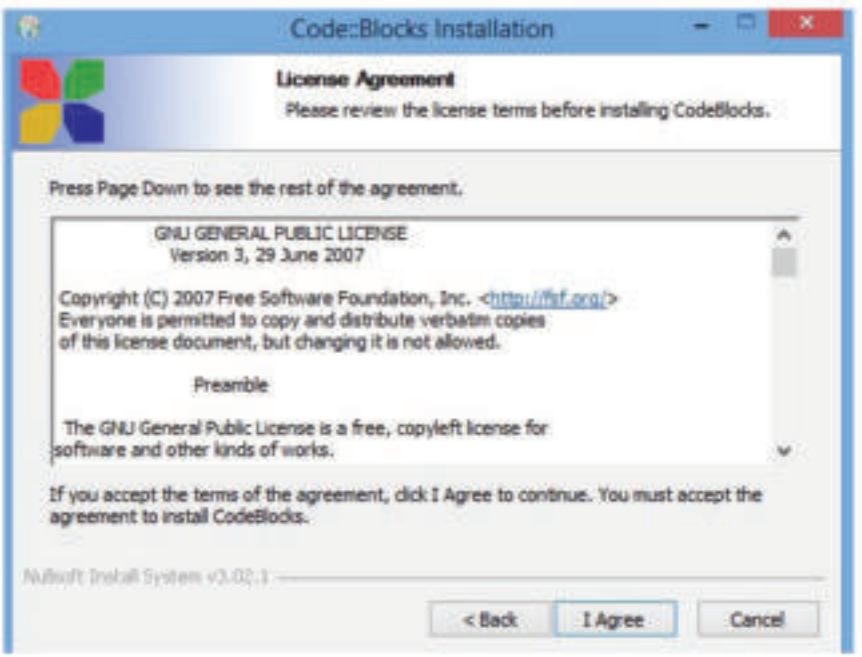
Code :: Blocks ücretsiz bir açık kaynaklı program olduğu için aşağıdaki sayfadan indirebiliriz:

https:////www.codeblocks.org/downloads/binaries.

İndirilen dosyayı çalıştırıyoruz yani yükleme işlemi dosyanın üzerine çift tıklayarak başlatmaktır. Aşağıdaki görüntüde gösterildiği gibi yükleme işlemine devam etme **sihirbazı** görüntülenir:

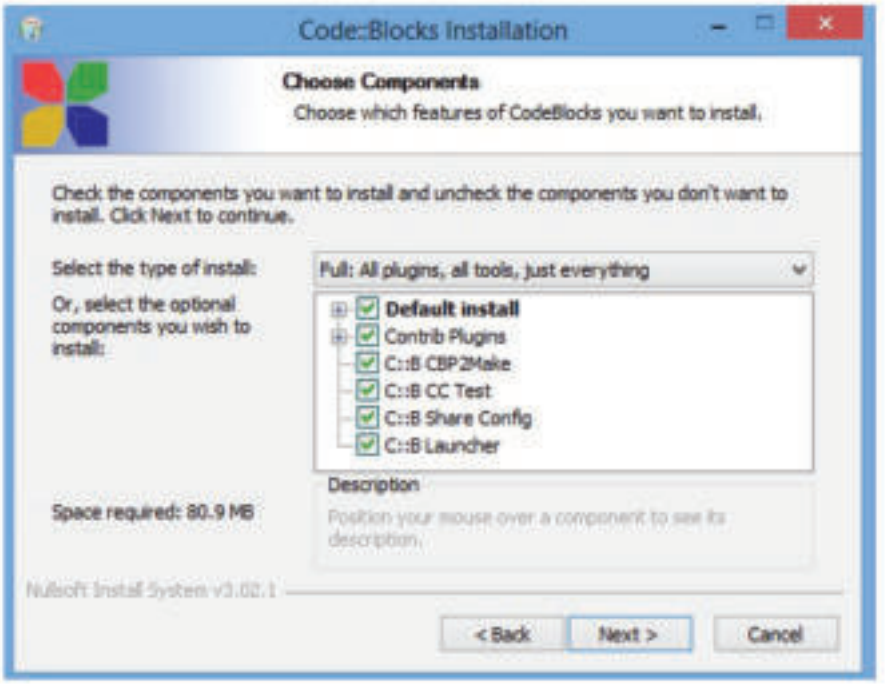
Şekil 1: Code :: Blocks yükleme işleminin başlangıcı

İleri - Next düğmesine tıklayarak, bu programı kullanma ilkelerini kabul edip etmediğimiz olan yükleme işleminin bir sonraki adımına yaklaşıyoruz:



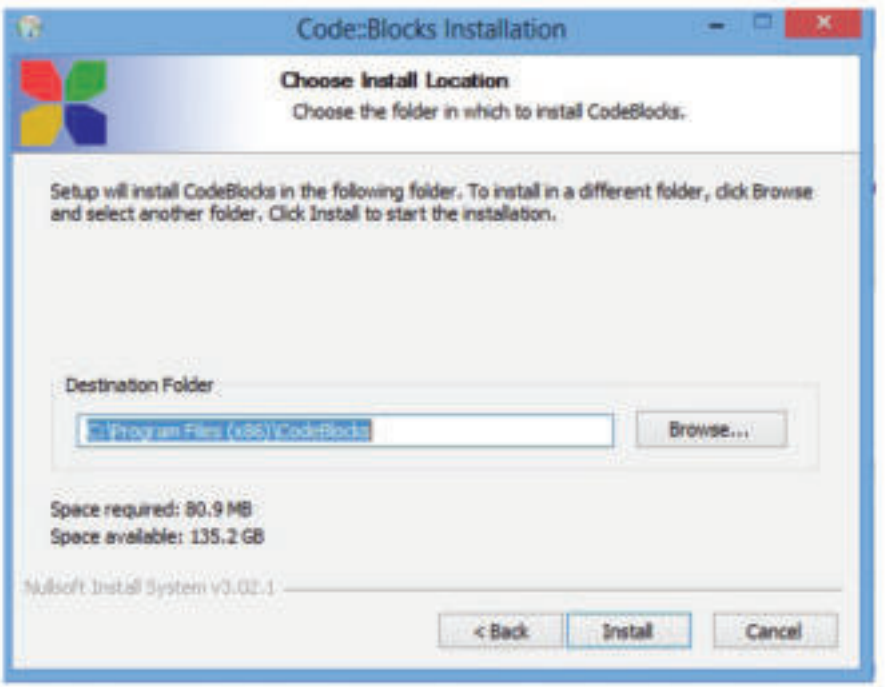
Şekil 2: Program kurulum kurallarının kabulü

İşleme devam etmek için **Kabul Ediyorum – I Agree** butonuna tıklayın ve bu programın bir paket olarak yüklemek istediğimiz bileşenlerin seçimi olan bir sonraki adıma geçin:



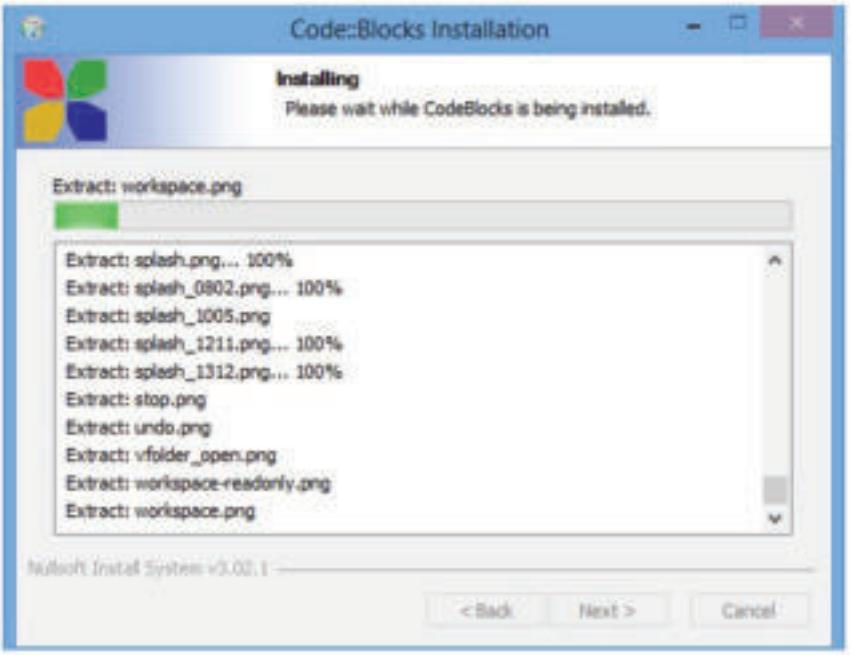
Şekil 3: Code::Blocks arayüzünde ek bileşenlerin seçimi

İleri - Next düğmesine tıklayarak, programın normal ve sorunsuz çalışması için gerekli dosyaları depolayacak klasörü seçtiğimiz aşağıdaki pencereyi açar:



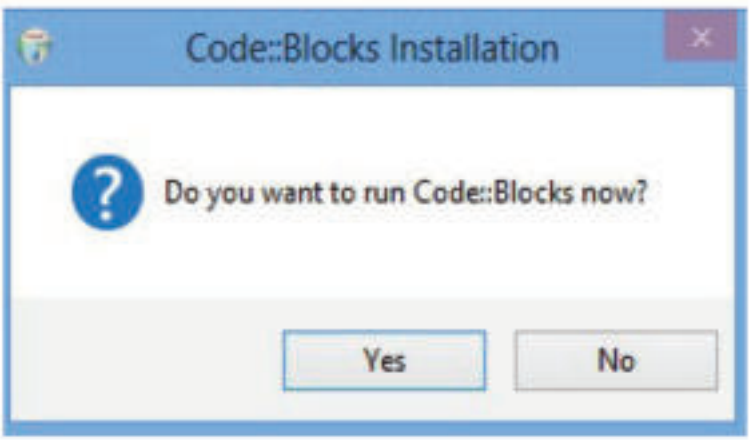
Şekil 4: Program dosyalarının depolanacağı klasörü seçin

İlk kurulum adımlarını tamamladıktan sonra, **Yükle - Install**'a tıklayın ve program dosyalarını hedef klasöre aktarmaya başlasın:



Şekil 5: Programın kurulumu

Bu işlem tamamlandığında, sonraki adıma geçmek için Sonraki seçeneğine tıklıyoruz, **Bitir - Finish** sonunda, Kod :: Blok başlangıç penceresinin göründüğü yer:



Şekil 6: Code :: Blocks başlangıç iletişim kutusu

Bu iletişim kutusundan, programın başladığı Evet –Yes seçeneğini seçiyoruz ve entegre programlama ortamı Code :: Blocks'un masaüstünü görüyoruz.

4.2.2 Code::Blocks'un arayüzü

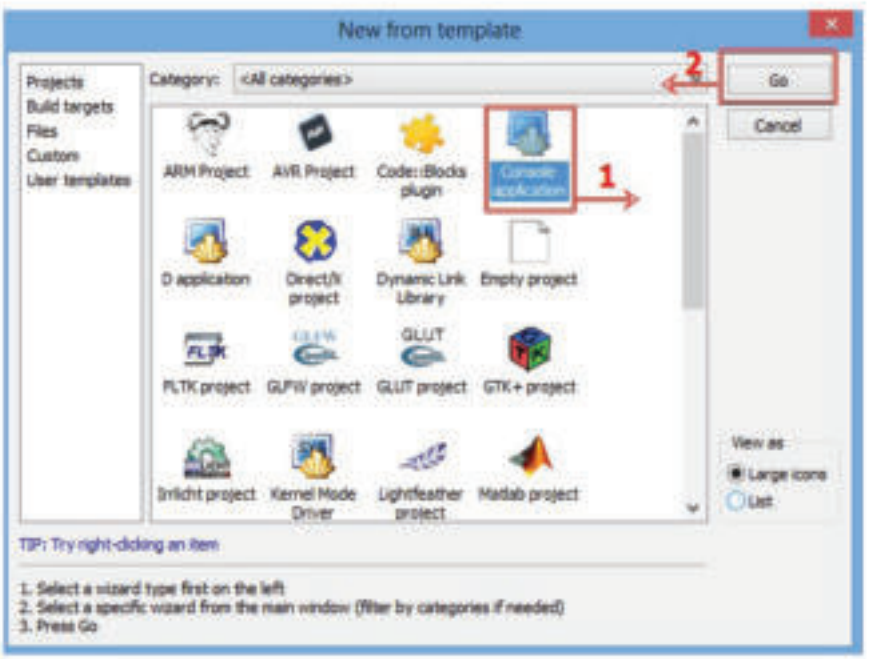
Programı başlattıktan sonra Code :: Blocks programının çalışma penceresi açılır. Arayüzüne bir göz atalım ve araçların işlevlerini anlatalım!

Şekil 7: Code::Blocks un arayüzü

Arayüzü aşağıdak unsurları içer r:

1. Komutlu menü çubuğu;
2. Yönet m bandı;
3. Kaynak Kodu (**Source Code**) g r ş düzenley c s ;
4. Hata mesajı b ld r m penceres .

Entegre Code :: Blocks programlama ortamında yen b r proje oluşturmanın lk adımı, aşağıdak penceren n göründüğü **Dosya - File → Yeni - New → Proje - Project** menüsüne tıklamakla başlar:

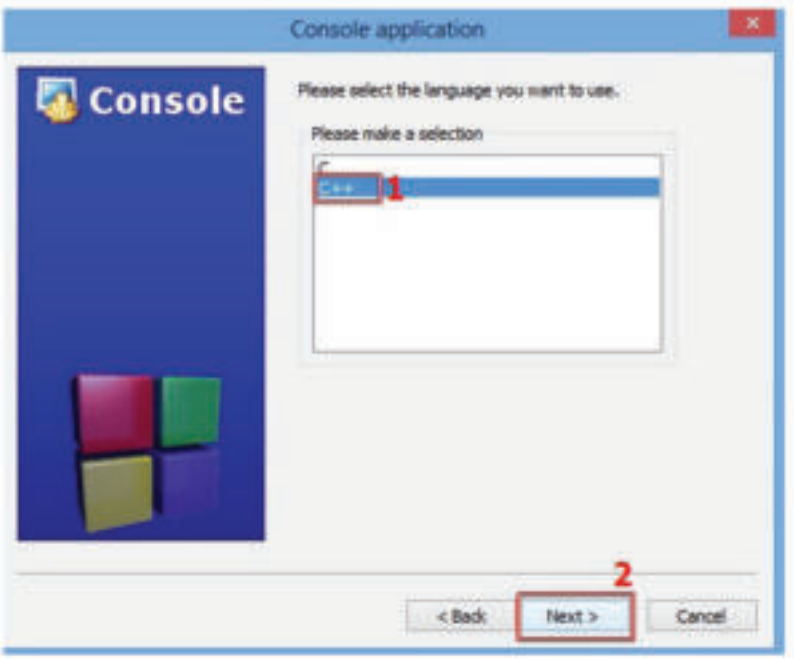


Şekil 8: Yeni bir proje oluşturma

Pencereden **Konsol Uygulaması'nı - Console aplication** seç n ve kinci adıma geçmek için **Git – Go** düğmesine tıklayın.

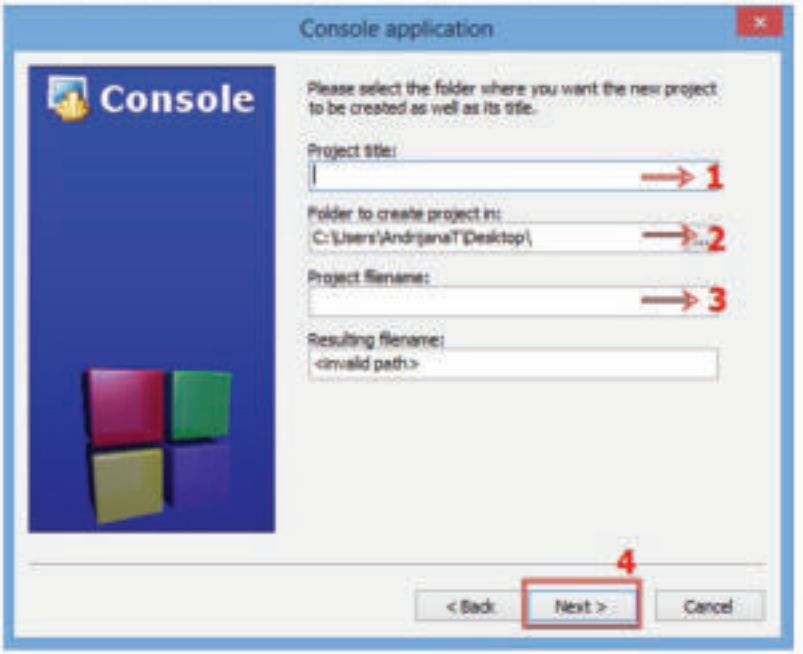
Şekil 9: Masaüstünü özelleştirme

İleri'ye - Next tıkladığınızda, aşağıdaki resimde gösterildiği gibi, hangi programlama dilini kullanacağımızı seçtiğimiz bir pencere görünür:



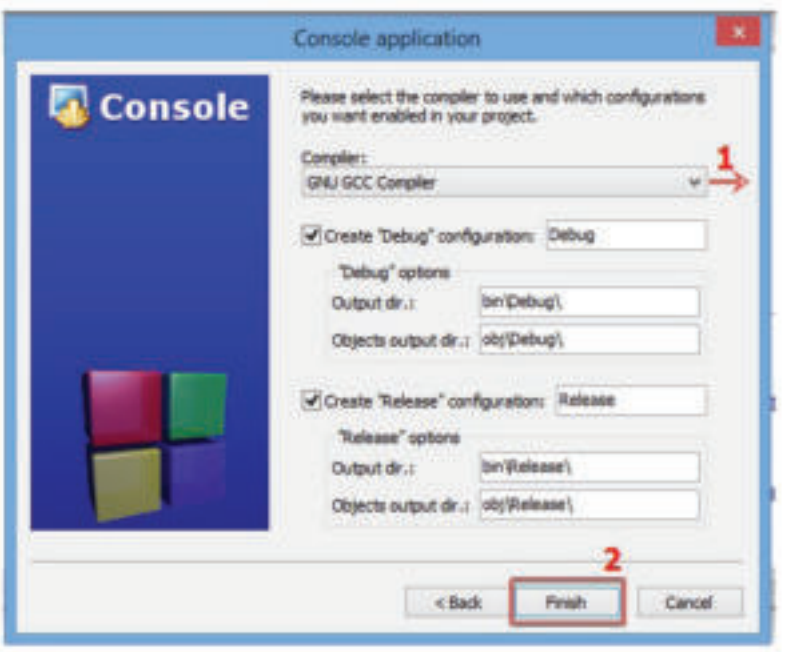
Şekil 10: Programlama dili seçimi

İleri - Next düğmesine tıklayarak, projenin adını ve bu projenin yerleştirileceği konumu, yani kaydedileceği yeri atamaya devam ediyoruz:



Şekil 11: Projeye bir isim ve konumun ayarlanması

Önce projeye isim veriyoruz, sonra projenin hangi klasöre yerleştirileceğini belirliyoruz, ayrıca oluşturulmakta olan proje dosyasını da isimlendiriyoruz ve son olarak **Next** butonuna tıklıyoruz. **İleri - Next** düğmesine tıklayarak derleyicinin seçimine geçiyoruz:



Şekil 12: Derleyici seçimi

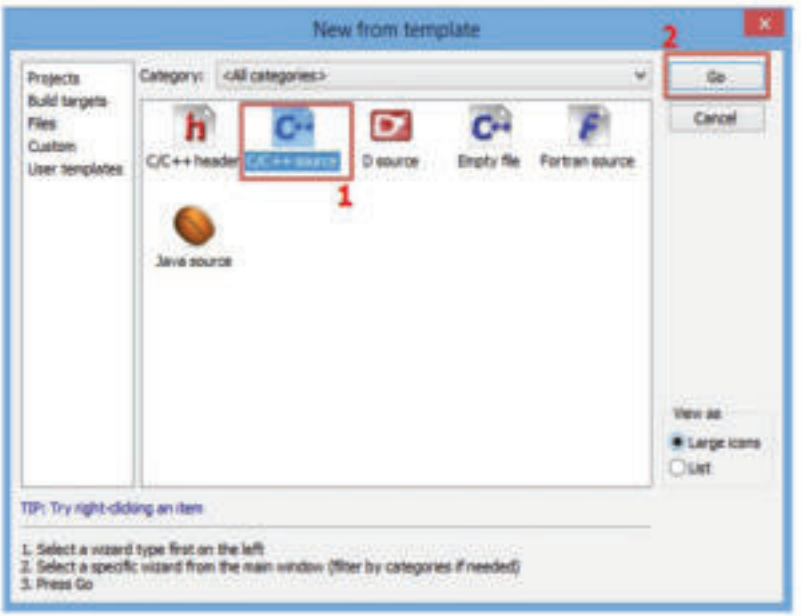
Bitir - Finish düğmesine tıklamakla, entegre programlama ortamı düzenleyicisinde kaynak kodu oluşturma işlemini başlatır.

4.2.3 Yeni bir kaynak kod dosyası oluşturun

İpucu

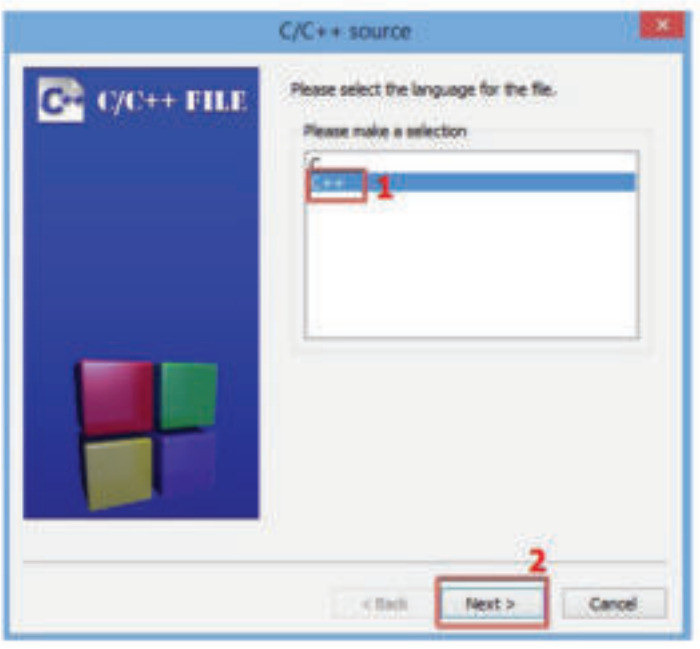
Bilgisayarınızın masaüstünde, oluşturulan tüm projeleri kaydedeceğiniz bir klasör oluşturun.

İlk olarak, kaynak kodunu yazmak için **yeni bir** dosya oluşturun. Adım adım prosedürü başlatarak **Dosya - File → Yeni - New → Dosya - File** menüsü aracılığıyla yeni bir dosya oluşturulur. İlk adım, dosya türünü seçmektir:



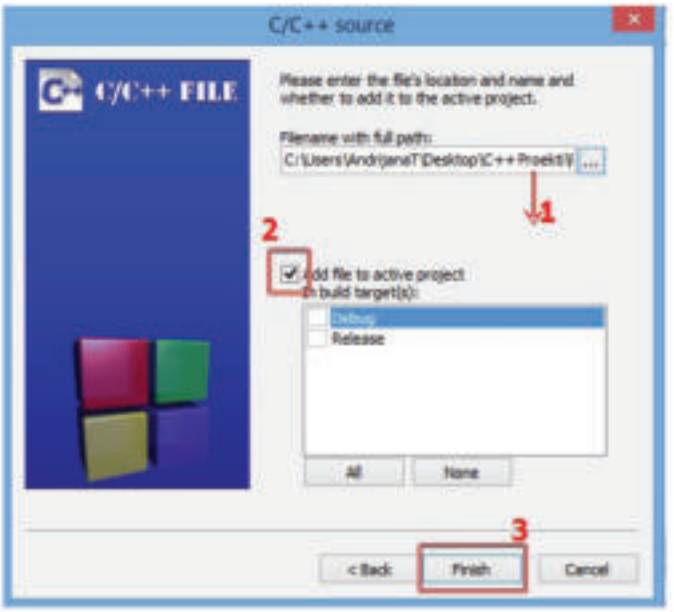
Şekil 13: Bir C ++ dosyası seçme

C / C ++ Kaynak - Source dosyasını seçin ve **Git – Go**'ya tıklayın. Ardından, programlama dilinin seçimi olan ikinci adıma geçiyoruz:



Şekil 14: Programlama dili seçimi

C ++ programlama dilini seçin ve prosedürün bir sonraki adımına geçmek için **Next** düğmesine tıklayın:



Şekil 13: Bir C ++ dosyası seçme

Pencerede Browse düğmesine tıklıyoruz ve açık pencereden yeni oluşturulan dosyanın adını ve konumunu ayarlıyoruz. **Finish** butonuna tıklayarak **Source Code** editörüne giriyoruz ve programı yazmaya başlıyoruz.

Ezberle!

Entegre bir programlama ortamı, bir program oluşturmak için gereken araçları içeren bir yazılım paketidir. Bu tür araçlar şunlardır: kaynak kodu düzenleyicisi, kod kitaplıkları, derleyiciler ve test platformları. Code::Blocks, ücretsiz bir entegre açık kaynak programlama ortamıdır.

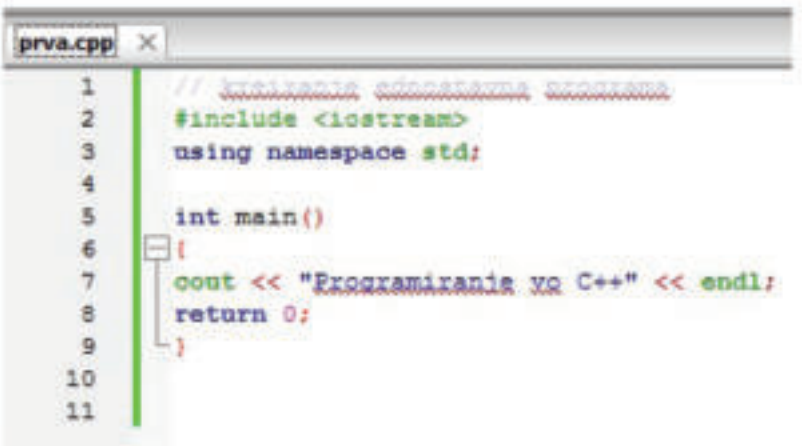
Sorular

1. Entegre programlama ortamı nedir?
2. Entegre programlama ortamının temel amacı nedir?
3. Entegre programlama ortamı hangi araçları içerir?
4. Kaynak kod pencerenin hangi kısmına yazılır?
5. Derleyicinin programlama masaüstünde ödevi nedir?
6. Hata ayıklayıcının rolü nedir?
7. Code::Blocks entegre programlama ortamının masaüstü penceresini açıklayın!

4.3 Program kodlarının görünümü

Program kaynak kodunu yazmaya başlamak için programlama dilinin **yapısını** ve **sözdizimini** bilmemiz gerekir. Bir programlama dili, bir program oluşturmak için kullanılan bir dizi kural, sembol ve özel karakter olduğundan, sorunsuz ve doğru çalışacak bir program oluşturmak amacıyla izlenmesi gereken sözdizimi (dilbilgisi) ve anlambilim (mantık) kuralları vardır.

Kaynak kodu, kaynak kodu editörlerinde yazılır. Aşağıda örnek bir kaynak kodu verilmiştir:



Şekil 1: Kaynak kod örneği

Kaynak kodu düzenleyicisinden her satırın numaralandırıldığını görüyoruz. Numaraların yanında bu kodun çevrildiğini gösteren yeşil bir çizgi var. Önceden, çevirme yapılmadan önce bu satır sarıydı. Her satırdaki kodun anlamını açıklayalım!

Satır numarası	Kaynak kodun anlamı
S1	// işaretleriyle, bir yorum yazmaya başlanılır veya bundan sonra yazılacakların bir açıklaması yapılır
s2	iostream kütüphanesi ekleniyor
s3	using namepace std kütüphanenin standart öğeleri kullanılacağını belirtir
s4	Boş satır, kodun derlenmesinde işleme alınmıyor
s5	Ana işlev main()
s6	Açılan süslü parantez main() işlevindeki emirlerin uygulanmasının başlangıcını ve kapanan süslü parantez bu işlevin sonunu bildirir.
s7	Cout ekrana yazdırma emridir << yazdırma operatörleridir “Programıranje vo C++” ekrana yazılacak metindir. Endl satır sonu emridir. “;” emrin sonunu belirtir.
s8	return 0; Programın başarılı bir şekilde tamamlandığına dair işletim sistemine bir mesajdır.
s9	Süslü parantezin kapatılması, ana işlev main() in sonunu bildirmektedir.

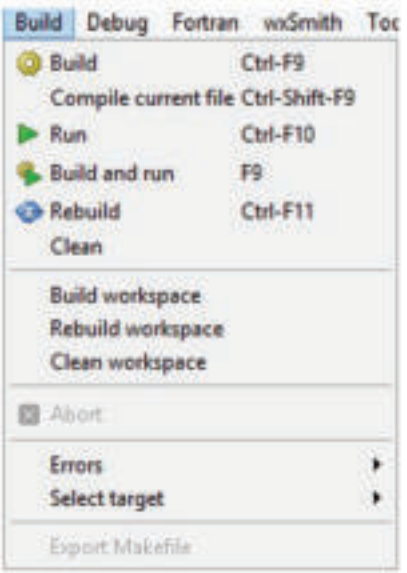
Şekil 1: Kaynak kod örneği

İpucu

Her ifadeden sonra noktalı virgül (;) yerleştirilir ve bu ifadenin burada bittiğini gösteren bir işarettir.

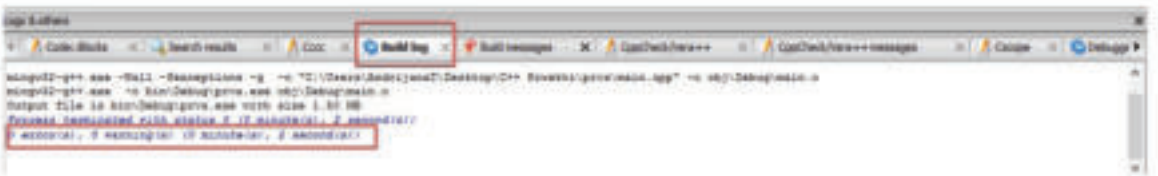
Geliştiriciler, kaynak kodu yazarken, kaynak kodu analiz ederken kaosa ve kafa karışıklığına katkıda bulunmamak için yazılı komutların netliğine dikkat ederler. Örneğin, derleyiciye için komutların tek satırda mı yazılacağı yoksa ayrı satırlarda mı düzenleneceği hiçbir şey ifade etmez, çünkü noktalı virgül (;) komutun sonu olduğunu açıkça belirtir.

Kodun derlenmesi **Build → Build** menüsü veya **Ctrl + F9** klavye tuşlarının bir kombinasyonu kullanılarak tek bir aşamada gerçekleşir:



Şekil 2: Build - Oluştur komutu

Bu aşamada hatalar oluşursa, geliştiriciler bunları çözmeye devam eder ve kaynak kodunu yeniden derler. Bir hata bulunduğunu belirten mesaj, masaüstünün altında ayrı bir Derleme çerçevesinde görüntülenir:



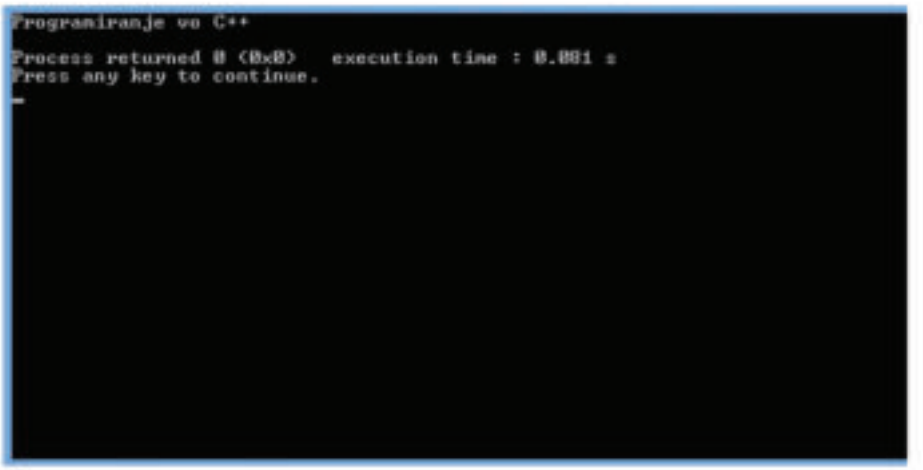
Şekil 3: Build Log penceresi

Programı **Build → Run** komutuyla veya **Ctrl + F10** klavye tuşlarının bir kombinasyonuyla çalıştıracağız. Bu işlemden sonra, oluşturulan klasörde üç dosya oluşur: kaynak, nesne ve çalıştırılabilir dosya.Her ifadeden sonra bir noktalı virgül (;) yerleştirilir ve bu diğer durumlar dışında, ifadenin burada bittiğini gösteren bir işarettir.

Build komutunu ve **Run - Çalıştır** komutunu çağırmak, menü çubuğunun hemen altında bulunan araç çubuğu aracılığıyla da yapılabilir:

Şekil 4: Build ve Run - Çalıştır araç simgeleri

Run - Çalıştır'a tıklamak, görüntülenen programı başlatır ve resimde gösterildiği gibi ayrı bir pencerede çalışır.



Şekil 5: Yürütülebilir pencere

Programın çalışmasıyla bu pencerede çalıştırılan programın mesajı geliyor. Ondan sonra bilgisayarın oluşturduğu "**Press any key to continue**." mesajı yazıyor. Çıkmak için herhangi bir tuşa tıkla anlamındadır.

Ezberle!

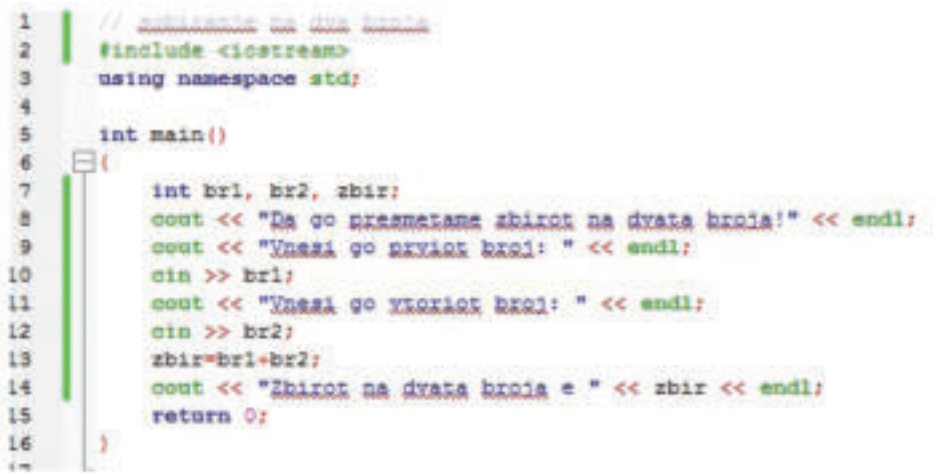
Bir program yazarken, programlama dilinin **sözdizimine**, **anlamsallığına** ve **yapısına** daima dikkat etmeliyiz. Kaynak kodu, her satırın numaralandırıldığı bir düzenleyicide yazılır. Kaynak kodu derlenir ve bağlanır ve tek aşamalı olarak gerçekleşir. Hata oluşursa, mesajlar **Build log** penceresinde görünür. Programın çalışması ayrı bir pencerede görüntülenir ve sonuç olarak üç dosya görüntülenir: kaynak, nesne ve çalıştırılabilir dosya.

Sorular

1. Kaynak kodu nedir?
2. Bir kaynak dosyayı hangi komut deriyor?
3. Kaynak dosya derlendikten sonra kaç dosya oluşur? Hangileri?
4. Bir program nasıl çalıştırılır?

4.4 Hazır örnek programların yürütülmesi

Önceki öğretim biriminden, yalnızca ekranda bir mesaj yazmayı gerçekleştiren basit programdaki temel ve standart öğelerin anlamını ve işlevselliğini öğrendik. Diğer unsurları tanımak için, programın yürütülmesi sırasında etkileşimli hareket etme, yani klavyeden veri veya değer girme fırsatına sahip olacağımız bir program oluşturacağız. Kaynak kodu düzenleyicide, entegre **Code::Blocks** programlama ortamında, aşağıdaki komut dizilerini yazın:

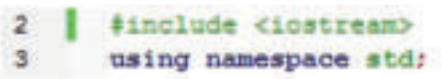


```
1   // Sobiranje na dva broja
2   #include <iostream>
3   using namespace std;
4
5   int main()
6   {
7       int br1, br2, zbir;
8       cout << "Da go presmetame zbirot na dvata broja!" << endl;
9       cout << "Vnesi go prviot broj: " << endl;
10      cin >> br1;
11      cout << "Vnesi go vtoriot broj: " << endl;
12      cin >> br2;
13      zbir=br1+br2;
14      cout << "Zbirot na dvata broja e " << zbir << endl;
15      return 0;
16  }
```

Şekil 1: Toplamı hesaplama programı için kaynak kod

Programın ilk satırına bir **yorum** yazdık, yani ne tür bir program oluşturacağımız ve o programın ne hesaplayacağına dair bir **açıklamadır**. Bizim durumumuzda iki sayının toplamını hesaplayacağız.

Her program, hangi **kütüphaneleri** dahil edeceğimizi ve hangi komutları kullanacağımızı tanımladığımız bir giriş ile başlar:



```
2   #include <iostream>
3   using namespace std;
```

Şekil 2: Programa giriş

Bu durumda, **iostream** kitaplığı dahil edilir, bu da sonucun verilmesi ve görüntülenmesi için standart komutların uygulanacağı anlamına gelir. Aşağıda, programı oluşturan komutların çalıştırıldığı ana işlev yer almaktadır:

Şekil 3: Ana işlev

Bu programda klavyeden verğeceğimizden dolayı, önce ana programda değşkenler, yan değerler klavyeden gelen grdye bağlı olan değşkenler tanımlıyoruz. Klavyeden gireceğimiz değerlerin **tamsayıları int (integer)** ile tanımlıyoruz. Değşkenlere şim verdik: **br1, br2** ve **zbir**.

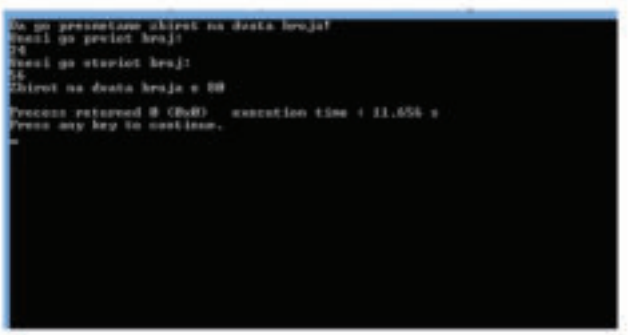
Sekiz (8) numaralı satırda, ekranda bir metin mesajı getiren **cout** komutunu yazılıdır. Komutun ardından yazdırma operatörler(<<) ve ardından tırnak şaretleriyle yazılmış bir ekranda görüntülenecek mesaj gelir. **Endl** komutu ile satırın sonunu ve noktalı virgül (;) simgesiyle ifadenin, yan komutun sonunu kastedilmektedir. Dokuz (9) numaralı satır aynı şlevsellğe ve anlama sahiptir.

On (10) numaralı satırda cin komutu yazılı ve ardından operatörler (>>) “>>”, klavyeden verğime komutunu belirtir. Elbette, komut noktalı virgül (;) ile bitiyor. Aynı prosedür on bir (11) ve on iki (12) numaralı satırlarda tekrarlanır.

On üçüncü (13) numaralı satırda, girilen iki sayının toplamını hesaplamak için formül yazılır veya harf harfine çevrilde **zbir** değşkeni **br1** ve **br2** toplamından bir değer alır. Ancak bu satır ekranda görüntülenmiyor, sadece tercüme ediliyor. **zbir** değeri komut dizisinin sonraki satırında görüntülenecektir.

Böylece, **cout** ve baskı operatörlerinin (<<) bulunduğu on dört (14) satırında, tırnak içinde yazılan cümle, ardından **zbir** değşkeninin değeri yazılacaktır. Bu, komutun endl ile bittiği yerdir. Ana şlev **return 0** ile sonlandırıyoruz; böylece CPU’ya programın başarıyla yürütüldüğünü gösterir. Parantezleri kapatarak **main** şlevindeki komutların sonunu şaretleriz.

Kaynak kodunu entegre programlama ortamının editörüne yazdıktan sonra, derleme ve hata ayıklamaya geçiyoruz, yan kaynak dosyayı çevirerek ve yürütülebilir bir dosyaya bağlayarak. **Build** menüsü ve **Build** komutu aracılığıyla çevirişaşamasını başlatıyoruz. Derleme günlüğünde herhangi bir hata görünmezse, dosyayı yürütmek için **Çalıştır - Run** komutuyla devam ederiz. Aksi halde hataları düzeltir ve süreci tekrar ederiz. Yürütülebilir dosya aşağıdaki pencerede yürütülür:



Şekil 4: Toplam hesaplama programının yürütülmesi

Bu ekranda aslında çalışmalarımızın sonucunu ve sürecin tanımlanan adımlara göre gidip gitmediğini görüyoruz.

Ancak bazen işler planlandığı gibi gitmez. Geliştiricinin çözmesi ve işlemin devam etmesi gereken çevre hataları olabilir. Bilgisayar hatalarına **baglar (Bugs)** denir, dolayısıyla hata ayıklama işlemine **hata ayıklama - debug** denir. Bu işlem kaynak kodu oluşturarak uygulayacağız.

Code :: Block'dak kaynak kodu düzenleyicisinde aşağıdak komut dizilerini yazalım:

```
1   // proizvod na dva broja
2   #include <iostream>
3   using namespace std;
4
5   int main()
6   {
7       int br1, br2, proizvod;
8       cout << "Da go presmetame proizvodot na dvata broja!" << endl;
9       cout << "Vnesi go prviot broj: " << endl
10      cin >> br1;
11      cout << "Vnesi go vtoriot broj: " << endl;
12      cin >> br2
13      proizvod=br1*br2;
14      cout << "Proizvodot na dvata broja e " << proizvod << endl;
15
16      system ("PAUSE");
17      return 0;
18  }
```

Şekil 5: Çarpımın hesaplanması için kaynak kod

Programı oluştururken yaptığımız hataları görmeye çalışalım. Bunları listeleyelim!

Kaynak dosyayı çevirme sürecinde aşağıdak sonucu alacağız:

Şekil 6: Hatalar bildirisi

Derleme günlüğü - Build log alanında, programı çevirirken hatalar olduğunu görüyoruz. **Build messageye** tıklayarak hataları tek tek görebiliriz:

Şekil 7: Hataların ayrı ayrı satırlarda gösterimi

Pencereden hatalarımız olduğunu fark ediyoruz:

- on (10) numaralı satırda cin komutunu çağırmadan önce ";" işaretleksik;
- on üçüncü (13) numaralı satırda, "**Proizvod**" un değerin tanımlamadan hemen önce ";" işaretleksiktir;
- sonunda **system ("PAUSE")** komutunu tanımıyor çünkü iostream kütüphanesine ait değildir.

Hata çözümüne yaklaşıyoruz, yanlış eksik yerlere ";" işaretini ekliyoruz ve giriş bölümünde sistem komutunun çalışmasını sağlamak için **cstdlib** kütüphanesini ekliyoruz **system ("PAUSE").**

Şekil 8: Düzeltmelerden sonra kaynak kodu

Şimdi hata olmadığı anda, çalıştırılabilir dosyanın ve programın sonucunun ayrı bir ekranda çalıştırmaya devam ediyoruz.

Ezberle!

Önişlemci komutları "#" işaretiyle başlar. # Include ile kaynak kodda komutlarını kullanacağımız kütüphaneleri duyuruyoruz. Main, her C ++ programının sahip olduğu ana işlevdir ve açık bir parantez "{" ile başlar ve kapalı bir parantez "}" ile biter. İki parantez arasında değişkenler tanımlanır, ifadeler yazılır, işaretler, semboller vb. Bir parçası olarak yazılır. Bug adı verilen hatalar çeviri yaparken ve bağlanırken ortaya çıkabilir, hataları algılama ve düzeltme işlemine hata ayıklama -debuger denir.

Sorular

1. Her program C ++ programlama dilinde nasıl başlar?
2. Programa bir giriş tanımlayın!
3. (///) iki eğik çizgi ile başlayan satırlara ne denir?
4. Standart kütüphaneleri açıkla!
5. Ana işlevin adı nedir ? Neler içerir?
6. Programdaki hatalar ne denir? Nasıl ayıklanır?
7. Return 0 ne anlama gelir?
8. Komutlar hangi işaret ile birbirinden ayrılır?

4.5 C++ programlama dilinin temel öğeleri

Hatırlayalım!

Günlük iletişimimizde bir dil kullanırız, yani bilgi, veri, fikir, görüş alışverişinde bulunuruz. Birbirimizi nasıl anlayacağız? Hangi dili kullanıyoruz? Konuştuğumuz dili ne oluşturur? Dönüştürmek için harfleri nasıl birleştiririz.

Bu sorular, bilgisayarla basit şeyler yaparken de ortaya çıkar. Bilgisayarlar ayrıca iletişim kurar: kullanıcı-bilgisayar-kullanıcı veya bilgisayar-bilgisayar. Kullanıcı, veri ve bilgi girdiği, emir verdiği, programları kullandığı vb. giriş cihazlarının uygulaması yoluyla bilgisayarla iletişim kurar. Bu faaliyetler, bilgisayar tarafından anlaşılabilir **ikili dile** çevrilir ve bunun tersi de, bilgisayarın bir etkinliğe yanıt vermesi gerektiğinde, ikili ifadeler kullanıcının anlayabileceği dile çevrilir ve böylece sonucu ekranda görürüz.

Bilgisayar ağları oluştuğunda, bilgisayarlar arasındaki iletişimin meydana geldiği ve veri, bilgi, belge, program ve benzerlerini değiş tokuş edebildiği söylenebilir.

Doğal diller gibi programlama dillerinin de kendi alfabeleri vardır. **Programlama dillerinin alfabesi şunlardan oluşur:**

- alfabenin büyük ve küçük harfleri;
- sıfır (0) dan dokuz (9) a kadar rakamlar;
- özel işaretler;
- boşluk işareti;
- iki veya üç karakter kombinasyonu.

Bu öğelerin uygun kombinasyonu ve uygulaması **ifadeler** yani **komutlar** oluşturur, bu nedenle p**rogramlama dilinin yapı öğeleri** olarak adlandırılır. Bu nedenle, programlama dilinin yapı taşları şunlardır:

- Ayrılmış kelimeler;
- tanımlayıcılar;
- operatörler,
- noktalama işaretleri;
- Yorumlar.

Bu yapı taşlarının her birinin, programlama dilinin yapısında ve sözdiziminde yeri vardır. Bunları açıklayıcı bir şekilde açıklayalım! Ayrılmış kelimelere ayrıca anahtar kelimeler de denir. Önceden **tanımlanmıştır** ve kaynak dosyayı çevirirken, çevirmen tam olarak ne demek istediklerini bilir.

Bunlar örneğin: int, delete, if, then, else, true, false, while ve başka. Programlama dilindeki tanımlayıcılar programcı tarafından tanımlanır. Aşağıdaki kurallara göre birleştirilebilen karakter kombinasyonlarıdır:

- isim bir harf veya küçük harfle başlar;
- büyük ve küçük harfler farklılık gösterir, örneğin: “Toplma” ile “toplam” aynı değildir;
- isim bir rakam içerebilir ancak bir rakamla başlamaz;
- isim ayrılmış bir kelime olamaz;
- isim !, @, # gibi özel bir karakter içeremez.

Programlama dillerindeki **operatörler**, bir programda gerçekleştirilen aritmetik, mantık ve diğer işlemleri belirtmek için kullanılır. Noktalama işaretleri, bir ifadenin sonunu belirtmek için “;” işareti gibi programlama dilinin öğelerini ayırmak içinkullanılır.

Programcı, programı oluştururken bir **yorum** veya **yorumlar** yazabilir. Yorum yazmak iki eğik çizgi “//” ile başlar. Aslında yorumlar, programın neye yönelik olduğunu açıklar veya tek tek ifade veya komut satırlarını açıklar. Yorumlar derlenmez yani çalıştırılmaz, sadece öğenin, komut satırının veya bir bütün olarak programın anlamı için bir kılavuz veya açıklama ödevi görürler.

Farklı programlama dilleri, söz dizimindeki farklılıklarla tanınır yani farklı ayrılmış sözcükler, kullanım kuralları vb.

C ++ programlama diliyle çalıştığımız için, yapı taşları şunlardır:

1. **veriler** - programlama dilleri ile farklı veri türleri işlenebilir ve tüm veriler şu özelliklere sahiptir: isim, tür ve değer. Farklı veri türlerine örnekler: char, int, bool, float, double, vb.

2. **sabitler**, değerlerini değiştirmeyen miktarlardır;

3. **değişkenler** - değerini değiştiren isimlerdir;

4. **bildirimler** aslında değişkenin bir açıklamasını verir, yani değişkenin ne tür veri içereceği, örneğin: **int x, float a**, vb.;

5. **ifadeler** - çözülecek ödeve bağlı olarak bazı hesaplamalar yapmak için kullanılır. Örnek: sum + i vb..;

6. **deyimler**, programda çalıştırılması gereken komutlardır, örneğin: **sum = sum + i, while, for, if –else, switch, break,** vb.;

3. **fonksiyonlar - işlevler - C ++** 'da ana fonksiyon **main "()"** dir. İfadeler, yani emirler, main işlevinin küçük parantez arasına yazılır, burada her bir ifade ";" işaretiyle biter. Programcı tarafından oluşturulabilen işlevler vardır, ancak en yaygın kullanılanları önceden var olan işlevlerdir;

4. **modüller** - program kaynak kodunu yazarken her öğe farklı bir renkte görüntülenir. Renk sayesinde kelimeyi veya ifadeyi doğru yazıp yazmadığımızı anlarız.

Programlama dilinin yapı taşlarının birleştirilme şekli, kursları ve sırası **programlama dilinin yapısını tanımlar**. Örneğin, bir programın yapısı şu şekildedir:

Şekil 1: C ++ Yapısı

Ezberle!

Programlama dillerinin yapı taşları şunlardır: Ayrılmış sözcükler, tanımlayıcılar, operatörler, noktalama işaretleri ve yorumlar. C ++ programlama dili aşağıdaki temel unsurlarla karakterize edilir: veriler, sabitler, değişkenler, bildirimler, ifadeler, işlevler ve modüller

Sorular

1. C ++ programlama dilinin yapı taşlarını listeleyin!
2. C ++ programlama dilinin temel öğelerini listeleyin!
3. Ayrılmış sözcükler nelerdir? Bazılarını listeleyin!
4. Tanımlayıcı nedir? Nasıl yaratılırlar
5. Oluşturulma kurallarına göre doğru oluşturulmuş bazı tanımlayıcı örnekler verin
6. Fonksiyonlar nelerdir? Programın hangi işlevi olmalıdır
7. Yorumlar nedir? Yorum için bir örnek verin!

4.6 Beyanlar

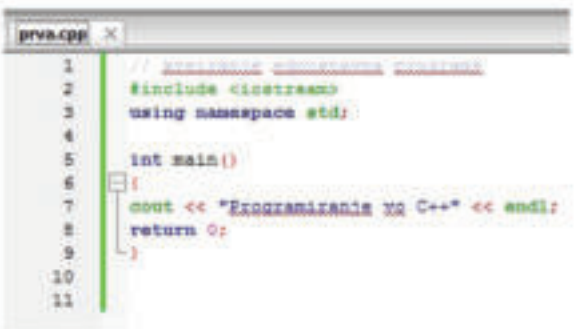
İfadeler, herhangi bir programlama dilinin temel öğeleridir. İfadeler olmadan bilgisayar, nihai sonuca ulaşmak için hangi adımların atılması gerektiğini bilemezdi. Bu nedenle ifadeler, bir bilgisayara ne yapması gerektiğini veya **bir çözüme ulaşmak için hangi eylemleri gerçekleştirmesi gerektiğini söyleyen komutlar olarak tanımlanabilir.**

Programdaki ifadeler bir işaret ile ayrılmıştır; "İşaret;" bize bu ifadenin sonu olduğunu söyler.

Programlama dillerinde birçok ifade vardır ve hepsinin farklı anlamları, işlevleri ve amaçları vardır. Programlama yaparken genellikle ekrandan ekrana ifadeler ve değer atama ifadeleriyle karşılaşırız.

4.6.1 Ekran görüntüsü bildirimi

Hazır programları incelerken karşılaştığımız ilk program, ekrana metin veya mesaj yazan basit bir programdı:



Bu kaynak dosyayı çevirip bağladıktan sonra, çalıştırılabilir pencerede "C ++ 'da Programlama" mesajı görüntülenir. Örneğe göre:

```
cout << "Programiranje vo C++" << endl;
```

Şekil 1: Ekran görüntüsü bildirimi

İfade, ekranın görüntülenmesine izin veren bir komuttur. Bu ifadenin unsurlarına bir göz atalım:

Eleman	Anlamı
cout	Ekran görüntüsü ifadesi
<<	Ekrana yazdırma operatörü
""	Tırnak işaretleri içinde yazılan her şey ekranda yazılacak
C++ ta programlama	Tırnak işaretleri içindeki ekrana yazılacak mesaj
endl	Satır sonu
;	İfadenin tamamlanması

Tablo 1: Ekran görüntüsü için ifade öğeleri

cout terimini başka şekillerde de kullanabiliriz. Bakalım cout ifadesi aşağıdaki şekillerde sunulursa ekranda ne gösterecek:

cout terimi	Ekran görüntüsü
cout<<"a Sayısı";	Tırnak işaretlerinde bulunan mesaj ekrana yazdırılacak
cout<<br;	Br değişkeninin değeri ekrana yazdırılır
cout<<55	Ekranda 55 sayısı yazdırılacak
cout<<3*br	Ekrana çarpımın sonucu gelecek

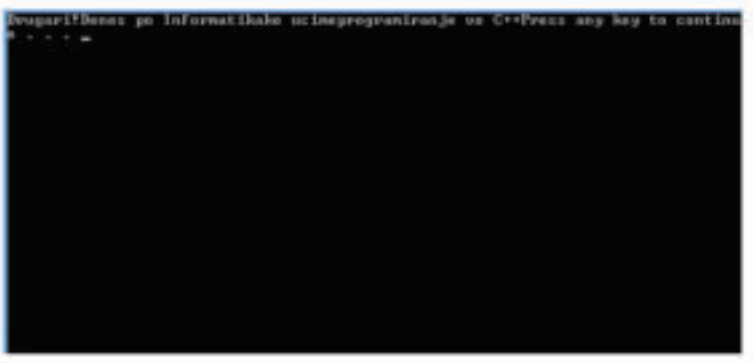
Tablo 2: cout ifadesini kullanmanın farklı yolları

Entegre programlama ortamı **Code :: Blocks** aracılığıyla, aşağıdaki örnek kaynak kodunda olduğu gibi, ekranda bir mesaj görüntüleyecek bir program oluşturmak için program yazalaım.

```
1   // programa za pishuvanje na ekran
2   #include <iostream>
3   #include <cstdlib>
4   using namespace std;
5   int main()
6   {
7   cout << "Drugari!";
8   cout << "Denes po Informatika";
9   cout << "ke ucime";
10  cout <<"programiranje vo C++";
11  system ("pause");
12  return 0;
13  }
14
```

Şekil 2: Ekrandaki metni görüntülemek için kaynak kodu

Bu programın sonucu aşağıdaki resimde gösterilmektedir:



Şekil 3: Yazılan ekran mesajı

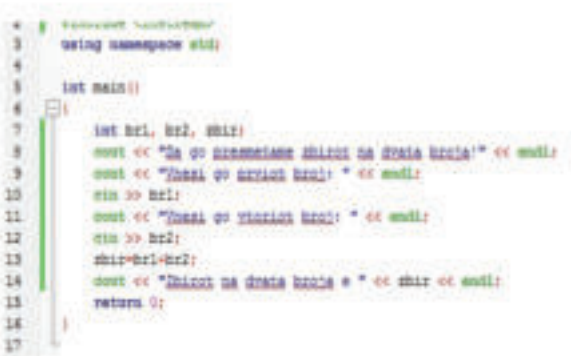
Resimden mesajın tek satırda yazıldığını, tüm bölümlerin birleştirildiğini ve bu şekilde görüntülendiğini fark ediyoruz, görünürlük ve netlik gösterm yor. Bunları ayrı satırlara ayırmak için **endl** komutunu kullanacağız:

Örneğin kaynak kodunda, yazdırılacak birden çok deyim olduğunu ve istelendiği gibi tek tek çalıştırılacağını fark ettik. Bu şekilde yazılan ifadelere bir dize veya ifade dizisi denir.

Şekil 4: endl komutunu çalıştırma

4.6.2 Değer atama beyanı

Girilen iki sayının toplamını hesaplarken değer atama ifadesini uyguladık. Değişkenin değerinin atanması klavyeden girilerek veya hesaplama sırasında elde edilen bir değerden olabilir.
İki sayının toplamını hesaplamak için programın kaynak koduna bakalım!



Resimden, programın açıklaması olarak bir yorumla başladığını fark ediyoruz. Yorumu, ön işleme olayları ve **namespace** standart ifadelerin kullanılacağına dair bildirim izler. Ardından **main ()** ana işlev başlar.

Şekil 5: Bir değişkene değer atama

Ana işlev, sırayla yürütülecek bir dizi ifadeyi gösterir. Değişkenler ve ekran üstü gösterim ifadesini açıklamaya ek olarak, burada atama değer ifadesinde kullanılır. Klavyeden girerek bir değişkene bir değer atayan bir ifade olarak **cin** ifadesini uygulama prosedürü aşağıdaki şekilde gösterilmiştir:

```
cout << "Vnesi go prviot broj: " << endl;
cin >> br1;
```

Şekil 6: Klavyeden değer atama

Bu nedenle **cin**, klavye aracılığıyla bir değişkene bir değer atayacağımız, yani klavyeden veri gireceğimiz bir ifadedir. Daha sonra **giriş operatörleri ">>"** ve ilk sayı için tanımlanan değişken, yani **br1** gelir.

br1 değişkeni, kullanıcı tarafından klavyeden girilen bir değere sahip olacaktır. Ama önce bu ifadeyi uygulamadan önce br1 değişkeni tanımlanmalıdır. Önceki şekilde **br1**, **br2** ve **zbir** değişkenleri **integer** tamsayı değerler, yani tamsayılar olarak tanımlanmıştır. Bu nedenle, değerleri klavyeden br1 ve br2'ye atamak için kullanıcı tamsayılar girecektir. Bu bir **veri giriş tekniğidir**. Zbir değişkenine klavyeden bir değer atanmayacaktır, ancak değeri aşağıdaki şekilde gösterildiği gibi iki sayının toplamının hesaplanmasının sonucu olacaktır:

```
zbir=br1+br2;
cout << "Zbirot na dvata broja e " << zbir << endl;
```

Şekil 7: Hesaplama sonucunda bir değişkene değer atama

Ekrandan, değişkenin değerinin kullanıcının klavyeden gireceği **br1** ve **br2** değerlerine bağlı olduğunu belirledik. İki sayının toplamı **zbir** değişkeninin değerini belirler. Son olarak, sonuç **cout** ifadesi ile ekranda görüntülenir.

Ezberle!

İfadeler, bir bilgisayarın bir çözüme ulaşmak için hangi eylemleri gerçekleştirmesi gerektiğini belirleyen komutlardır. Her ifadeden sonraki noktalı virgül (;) ifadenin sonunu belirtir. Cout ifadesi ekranı görüntülemek için kullanılır. Ardından operatörler (<<), mesaj için tırnak işaretleri gelir ve noktalı virgülle biter. Klavyeden bir değişkene bir değer atamak için bir ifade, yani veri girmek için cin ifadesi kullanılır. Cin ifadesini veri girişi için operatörler (>>) takip eder, değişken ve noktalıvirgül ile biter. Bu tür programlara etkileşimli programlar denir. Ancak klavye girişine ek olarak, değişken bir hesaplama sonucunda bir değer alabilir.

Sorular

1. İfade nedir?
2. Ekrana yazdırmak için hangi ifade kullanılır ?
3. Endl komutu neyi etkinleştirir?
4. Bir değer atamak için hangi ifade kullanılır?
5. Klavyenin değeri nasıl belirlenebilir?

4.7. Programların geliştirilmesi

Aşağıdaki ödevlerde, ekranda birden çok ifadenin **görüntülendiği** programlar oluşturacağız. Program kodundaki yazılı ifadeler sırayla tek tek çalıştırılacaktır ve bu çalıştırma yöntemine **sıralı çalıştırma tekniği** denir.

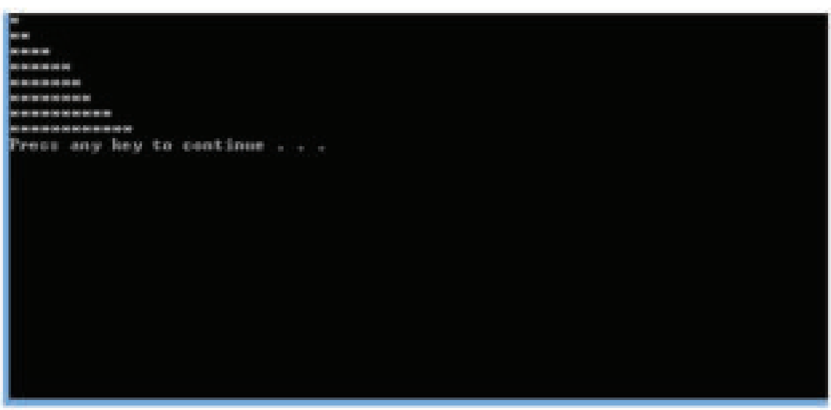
Örneğin, ekran görüntüsü ifadesinin yardımıyla bir dik üçgenin şeklini yazdırabiliriz. Burada sıralı çalıştırma tekniğini kullanacağız. Bunun için **Code :: Blocks**'ta yeni bir proje oluşturuyoruz ve programı yazmaya başlıyoruz:

```
//naplasteni iskazi za prikaz na ekran
#include <iostream>
#include <cstdlib>
using namespace std;

int main()
{
    cout << "*" << endl;
    cout << "***" << endl;
    cout << "*****" <<endl;
    cout << "*******" << endl;
    cout << "*********" << endl;
    cout << "***********" << endl;
    cout << "*************" << endl;
    cout << "***************" << endl;
    system ("pause");
    return 0;
}
```

Şekil 2: Ekran için katmanlı ifadelerin görüntüsü

Kaynak kodun neticesi aşağıdaki resimde gösterilmektedir:



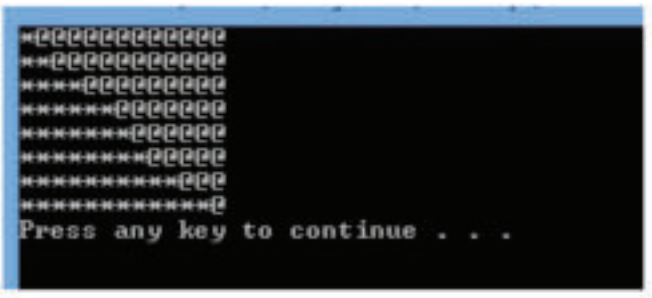
Şekil 2: Katmanlı ifadelerin görüntüsü

Aynı şekilde bir kare oluşturmaya çalışalım ve kareyi görüntülerken köşegen görünmelidir. Bu amaçla, halihazırda kullanılan yıldızlara ek olarak başka bir işaret kullanacağız. Bu programın kaynak kodu şöyle görünecektir:

```
1  //[redacted]
2  #include <iostream>
3  #include <cstdlib>
4  using namespace std;
5
6  int main()
7  {
8      cout << "*@@@@@@@@@@@@" << endl;
9      cout << "**@@@@@@@@@@@" << endl;
10     cout << "****@@@@@@@@@" <<endl;
11     cout << "******@@@@@@@" << endl;
12     cout << "*******@@@@@@" << endl;
13     cout << "********@@@@@" << endl;
14     cout << "**********@@@" << endl;
15     cout << "************@" << endl;
16     system ("pause");
17     return 0;
18 }
19
```

Şekil 3: Kare görünümü için kaynak kod

Programın çalıştırılması sonucu, farklı bir ekranda resimde gösterildiği gibi görünür.



Şekil 4: Çalıştırılan dosyadan elde edilen kare görünüm

Aynı prensipte, yaratıcı olmaya ve ekran gösterim ifadeleri yardımıyla nesneler çizmeye çalışalım. Örneğin, adınızın ilk harfini, kalp vb.

Ezberle!

Tümü sıralı olarak yürütülen program kodunda deyimler yazmak, sıralı yürütme tekniği olarak adlandırılır.

4.8 Aritmetik işlemler ve ifadeler

Aritmetik işlemler ve ifadeler genellikle C ++ programlama dilinde programlar oluştururken çok sık kullanılır. Aritmetik işlemler ve ifadeler temelde matematiksel hesaplamaları içerir.

Bu nedenle, **C ++ programlama dilindeki aritmetik ifadeler, matematiksel işleçlerle ilişkili iki veya daha fazla sayısal değerin kaydı olarak tanımlanabilir.**

Matematiksel operatörler, belirli bir eylemi temsil eden sembollerdir. C ++ 'da kullanılan temel aritmetik operatörler şunlardır:

Aritmetik işlemler	Anlamı
+	Toplama
-	Çıkarma
*	Çarpma
/	Bölme
%	Bölümden kalan

Tablo 1: C ++ 'da aritmetik işlemler

Not!

C ++ programlama dilinde üs operatörü yoktur. Ancak cmath kitaplığında tanımlanmış yerleşik bir pow işlevi vardır.

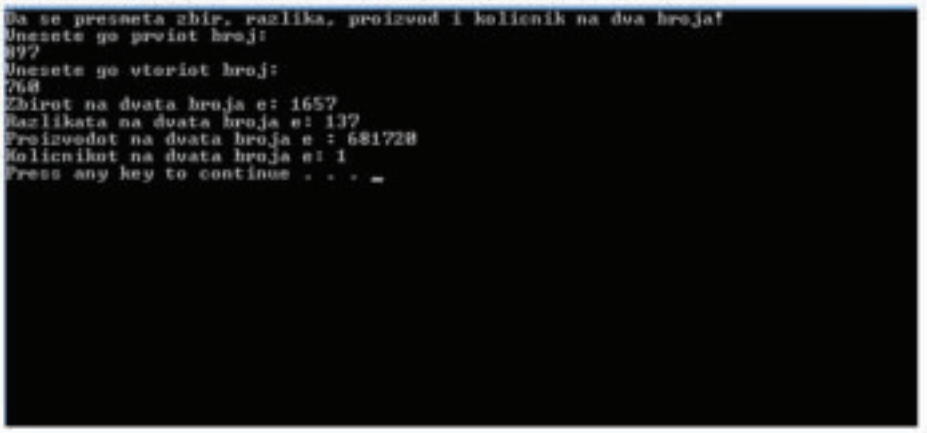
Aritmetik operatörlerin nasıl uygulandığını görmek için basit matematiksel işlemler gerçekleştirecek basit bir program oluşturacağız.

```
1
2   #include <iostream>
3   #include <cstdlib>
4   using namespace std;
5
6   int main()
7   {
8       int broj1, broj2, zbir, razlika, proizvod, kolicnik;
9       cout << "Da se presmeta zbir, razlika, proizvod i kolicnik na dva broja!" << endl;
10      cout << "Vnesete go prviot broj: " << endl;
11      cin >> broj1;
12      cout << "Vnesete go vtoriot broj: " << endl;
13      cin >> broj2;
14      zbir=broj1+broj2;
15      cout << "Zbirot na dvata broja e: " << zbir << endl;
16      razlika=broj1-broj2;
17      cout << "Razlikata na dvata broja e: " << razlika <<endl;
18      proizvod=broj1*broj2;
19      cout << "Proizvodot na dvata broja e : " << proizvod << endl;
20      kolicnik=broj1/broj2;
21      cout << "Kolicnikot na dvata broja e: " << kolicnik << endl;
22      system ("pause");
23      return 0;
24  }
```

Şekil 1: Aritmetik işlemleri kullanan kaynak kod

Programın diğer örneklerinde olduğu gibi programın giriş kısmı,**namespace** tanımlanan kütüphaneleri ve komutları içerir. **Main () ana işlevi** aslında çözüme ulaşmak için çalıştırılacak ifadeleri içeren ana programdır.

Gördüğünüz gibi ana program, değerlerin atanacağı değişkenleri tanımlar. Onlara atanacak değerler **integer**, yani tamsayıdır. **br1** ve **br2** değişkenleri klavye girişi ile kullanıcı tarafından atanan bir değere sahip olurken, **zbir**, **razlika**, **proizvod** ve **kolicnik** değişkenler hesaplama sonucunda bir değer alacaktır. Hesaplamalarda temel aritmetik işlemler kullanılır. Hesaplamanın sonucunu çalıştırılan pencerede görüyoruz:



Şekil 2: İki sayı ile aritmetik işlemler

Ezberle!

Aritmetik işlemler, bir hesaplama gerçekleştirmek için matematiksel işleçlere bağlanan iki veya daha fazla sayısal değerin kayıtlarıdır. C ++ programlama dilinde aşağıdaki aritmetik işlemler kullanılır: (+) toplama için, (-) çıkarma için, (*) çarpma için ve (/) bölme için. İfadeler, değişkenler ve operatörlerin bir kombinasyonu ile oluşturulur.

4.9. Sabitler ve değişkenler

Sabitler ve değişkenler, programlama dillerinin öğeleridir. Bir program oluştururken sabitlerin ve değişkenlerin kullanılması, bunların doğru uygulamaları için kurallara uyulmasını gerektirir. Onlarla tanışalım!

4.9.1 Sabitler

Sabitler, programın yürütülmesi sırasında değer değişmeyen verilerdir. C ++ programlama dilindeki sabit iki şekilde oluşturulur. Bir yol, **#define** komutunu kullanmaktır, ikinci yol ise **const** kelimesini sabitin türüne ve adına eklemektir. Örneğin:

```
1   // definiranje konstanta u C++
2   #include <iostream>
3   #include <cstdlib>
4   using namespace std;
5   #define pi 3.141
6   int main()
7   {
8       const int n=5;
9       cout << n*pi << endl;
10      system ("pause");
11      return 0;
12  }
13
```

Şekil 1: Sabitleri tanımlama

Programda tanımladığımız ilk sabit **3.141** değerine sahip sabit **pi**'dir. **#define** komutunu kullanarak tanımladık ve **main ()** fonksiyonunun başlangıcı duyurulmadan önce yerleştirildi. Bu komut **main ()** işlevinin başlangıcından önce yerleştirildiğinden, önişlemci aşamasında çalıştırıldığı kabul edilir ve bu nedenle sonunda ifadenin sonunu işaret eden noktalı virgül içermez.

İkinci sabit **n**'dir. Main () ana işlevinde tanımlanmıştır. Yani programda gösterildiği gibi ismine ve int türüne ek olarak const kelimesi de kullanılır, yani **const int n = 5**. Bazı durumlarda, sabit türü belirtilmezse, int türünde olduğu varsayılır. Hesaplamanın sonucu gerçekte 5 * 3.141 = 15.705'tir ve cout ifadesi aracılığıyla ekranda görüntülenir. Bu nedenle, sabitlerin değerinin ifade edildiği veri türüne göre birbirinden farklı olduğu sonucuna varabiliriz.

4.9.2 Değişkenler

Matematik, fizik, kimya ve diğer benzer bilimler okurken çoğu zaman bir anlamı olan birçok değişkenle karşılaşırız. Örneğin: çevre **L** ile gösterilir, alan **P** ile, yarıçap r ile, köşegen **d** ile, sıcaklık **T** ile, hız a ile gösterilir vb. **Değişkenler, programın yürütülmesi sırasında değerleri değişebilen verilerdir**. Değişkenler tür ve ada göre veya tür, ad ve değere göre tanımlanır. Örneğin:

```
int I, j;
int m=15;
float e=2,78;
int a=5, b=10
```

4.9.2.1 Değişken türü

Programda kullanılacak her bir değişken tanımlanmalıdır yani türünü tanımlamalıyız. Aşağıdaki tablo, en sık kullanılan değişken türlerin anlamları ile gösterir:

Değişken türleri	Anlamı
Char	Simge veya tamsayı değeri olabilen bir karakter değişkeni
Int	Değeri tamsayı olabilen değişken ... -2, -1,0,1,2, ...
Bool	Doğru (true) ve yanlış (false) iki değerden birine sahip olabilen mantıksal bir değişken
Float	Basit hassasiyetle gerçek sayı
Double	Çift hassasiyetli gerçek sayı

Tablo 1: Değişken türleri

4.9.2.2 Değişkene bir değer vermek

Değişken türü ve adına ek olarak, değişkene aşağıdaki örneklerde gösterildiği gibi "=" operatörünün yardımıyla bir **değer** atanabilir: **c = 8, a = 10, a = b * 3, x = x + 6, x = d = f = 2** vb. Bu prosedür, **başlangıç değerinin verilmesi** veya **eklenmesi** olarak adlandırılır. Bu, c değişkenine sekiz (8) değerinin verileceği, **a** değişkeninin **b * 3** çarpımının hesaplanmasından kaynaklanan bir değer alacağı, **x** değişkeninin **x + 6** toplamının, **f** değişkeninin (2) değerini alacak, **d** f'nin değerini, **x** se **d**'nin değerini alacaktır. Örneğin:

```
1  // inicijalizacija i deklaracija na promenliva
2  #include <iostream>
3  #include <cstdlib>
4  using namespace std;
5
6  int main()
7  {
8      //inicijalizacija
9      int a=6, b=18;
10     int suma;
11
12     //deklaracija
13     suma=a+b;
14     cout <<"Suma= " << suma <<endl;
15     system ("pause");
16     return 0;
17 }
18
```

Şekil 2: Değişkenlerin başlatılması ve bildirilmesi

Bu programda ana fonksiyondaki **dokuz numaralı satır (9) a ve b** değişkenlerini tanımlar ve başlatır: **int a = 6, b = 18. On (10) numaralı satır, suma** değişkenini tanımlar, ancak bu, başlatılmamıştır. Hesaplamayı yaptıktan sonra bir değer alacak: **suma = a + b** ve sonuç ekranda görüntülenecektir.

Ezberle!

Değişkenler ve sabitler bir programda kullanıldığında bildirilmelidir, bu da türlerini ve adlarını belirleme anlamına gelir. Değişkenler (=) eşit operatörün yardımıyla bir değer alır. Bir değişkene bir başlangıç değeri atanmasına başlama değeri denir.

Sorular

1. Sabit nedir ve değişken nedir? Aralarındaki fark nedir?
2. Beyan nedir? Birden çok değişken tek bir ifadeyle bildirilebilir mi? Örnek
3. Başlama değeri nedir? Örnek veriniz!

4.10 Programa veri girmek için ifadeler (teknikler)

Programdak ver g r teknğ n n özünü anlamak ç n b rkaç program oluşturacağız.

Ödev

G den b sayının kares n hesaplamak ç n b programın oluşturukması!

```
1   // presmetka na kvadrat na vnesen broj
2   #include <iostream>
3   #include <cstdlib>
4   using namespace std;
5
6   int main()
7   {
8       int broj;
9
10      cout << "*************************************************" << endl;
11      cout << " Da se presmeta kvadratot na vneseniot broj" << endl;
12      cout << "*************************************************" << endl;
13      cout << "Vnesi eden broj" << endl;
14      cin >> broj;
15      cout << "Kvadratot na brojot " << broj << " e: " << broj*broj << endl;
16      system ("pause");
17      return 0;
18  }
19
```

Şekil 1: Girilen bir sayının karesinin hesaplanması

G den b sayının kares n hesaplamak ç n programı oluştururken, programın g r şında bulunan açıklama ve ön şlemc komutlarına ek olarak, ana şlevde ekranda görüntülemek fadeler ve klavyeden ver g rmek ç n fadeler kullandık. Aslında burada ver g r teknğ n kullandık. Klavye ver ler n n g rdeceğ değ şken önce ne tür olduğunu tanımlar ve ardından uygun şek lde uygular. Yürütmen n sonucunu özel b ekranda görüyoruz:

```
*************************************************
 Da se presmeta kvadratot na vneseniot broj
*************************************************
Vnesi eden broj
5
Kvadratot na brojot 5e: 25
Press any key to continue . . .
```

Şekil 2: Sayının karesinin hesaplamanın sonucu

Ödev

Bir dairenin alanını ve çevresini hesaplayın! Bu, hesaplamada hazır bir matematiksel formül kullanmamız gerekir. Örneğin: Bir dairenin alanı P = r * r * p formülüyle hesaplanırken, çevre L = 2 * r * p formüllüyle hesaplanır.

```
1  // [illegible]
2  #include <iostream>
3  #include <cstdlib>
4  using namespace std;
5
6  #define pi 3.141
7
8  int main()
9  {
10     float radius, P, L;
11
12     cout << "*************************************************" << endl;
13     cout << " Da se presmeta plostina i perimetar na kruznica" << endl;
14     cout << "*************************************************" << endl;
15     cout << "Vnesi go radiusot na kruznicata: " << endl;
16     cin >> radius;
17     P=radius*radius*pi;
18     cout << " Plostinata na kruznicata iznesuva: " << P << endl;
19     L=2*radius*pi;
20     cout << "Perimetarot na kruznicata iznesuva: " << L << endl;
21     system ("pause");
22     return 0;
23 }
```

Şekil 3: Dairenin alanı ve çevresinin hesaplanması

Hazır matematik formüllerinden başlayarak **pi**'nin sabit değeri **3,141**'dir ve bu nedenle önişlemci bölümünde **#define** komutunu kullanarak onu bir sabit olarak tanımlarız. Ana fonksiyondaki değişkenler **float** olarak tanımladık, yani bir tamsayı ve ondalık kısım içeren gerçek sayılar, çünkü **pi** ondalık bir değere sahiptir ve hesaplamanın sonucunun ondalık bir sayı olması beklenir. Ayrıca, sıralı olarak yürütülen ve ayrı bir pencerede nihai sonucu veren ekran üzeri gösterim ile değer atama ifadeleri geçerlidir:

```
*************************************************
 Da se presmeta plostina i perimetar na kruznica
*************************************************
Vnesi go radiusot na kruznicata:
5
 Plostinata na kruznicata iznesuva: 113.076
Perimetarot na kruznicata iznesuva: 37.692
Press any key to continue . . .
```

Şekil 4: Dairenin alanı ve çevresinin hesaplanmasının sonucu

Ödev

Matematiksel formülü kullanarak bir üçgenin alanını hesaplayın: P = (a * h) / 2

```cpp
// presmetka na plostina na triagolnik
#include <iostream>
#include <cstdlib>
using namespace std;

#define pi 3.141

int main()
{
    float a, h, P;
    cout << "*************************************************" << endl;
    cout << " Da se presmeta plostina na triagolnik" << endl;
    cout << "*************************************************" << endl;
    cout << "Vnesi ja goleminata na stranata a: " << endl;
    cin >> a;
    cout << "Vnesi ja visinata na triagolnikot: " << endl;
    cin >> h;
    P=a*h/2;
    cout << "Plostinata na triagolnikot iznesuva: " << P << endl;
    system ("pause");
    return 0;
}
```

Şekil 5: Üçgen alanının hesaplanması

Programı oluşturmak ve kesin çözümü elde etmek için, bir üçgenin alanını hesaplamak için matematiksel formülü uyguladık. Değişkenlerin türünü belirledik, ödeve giriş olarak ekran üzeri ifade deyimi yardımıyla metni yazdırdık, değişkenlerin değerlerini klavye ile girdik ve formülle hesaplama yaptık. Bunu yaparken, dosyayı çalıştırırken aşağıdaki sonucu elde ettik:

```
*************************************************
 Da se presmeta plostina na triagolnik
*************************************************
Vnesi ja goleminata na stranata a:
7
Vnesi ja visinata na triagolnikot:
4
Plostinata na triagolnikot iznesuva: 14
Press any key to continue . . .
```

Şekil 6: Üçgen alanının hesaplanmasının sonucu

Ödev

Klavyede bir sayı girerken bir öncesini ve bir sonrasını gösterecek bir program oluşturun.

4.11 Karşılaştırmalı ifadeler

Günlük yaşamlarımızda, olaylar olur ve biz bu durum hakkında yeni kararlar almamız gerekir, ve bu kararlardan yola çıkarak hangi yoldan gideceğimize karar vermeliyiz. Örneğin, zorunlu okulu bitirdikten sonra, eğitimimize üniversitede devam edip etmeme ya da burada bitirme konusunda karar vermemiz gereken bir dönüm noktası gibi. Burada iki olasılığımız var. İlk fırsat, eğer üniversitede eğitimimize devam edersek, bilgimizi genişletmek ve dolayısıyla iyi ücretli ve profesyonel işler bulmakta rekabet edebilmek için geniş fırsatlar vardır. İkinci olasılık ise eğitime devam etmeme ve bugüne kadar kazanılan bilgilerde kalmaya karar vermek ve böylelikle piyasada rekabet etme fırsatı azalmaktadır. Bununla birlikte, bunlar hayattaki zor kararlardır, ancak bu öğretim içeriğinde konuşacağımız konuya giriş için iyi bir örnektir.

Not!

Kendinizi karar vermeniz gereken bir durumda buldunuz mu? Nasıl karar verdin? Seni böyle düşündüren ne oldu? Durumu açıklayın!

Bilgisayarlarda da benzer şeyler olur. Örneğin: bir program koşulun karşılanıp karşılanmadığına karar verdiğinde ve tatmin olursa hangi eylemin yapılacağına ve bunun tersi, koşul tatmin edilmezse ne yapmalı?

Bu tür ifadeler **koşullu ifadeler** olarak adlandırılır ve bir alternatif olduğunda kullanıcı bunları seçer. Bu, iddia doğruysa bir işlem, doğru değilse başka bir işlem yapılacağı anlamına gelir. Örneğin, Programları oluşturmaya başladığımızda, ifadelerin (komutların) birbiri ardına sırayla yürütüldüğünü belirttik. Bu tür koşullu ifadeleri uygularken, programın sırası kaybolur, çünkü yürütülmesi tanımlanan ifadenin doğruluğuna bağlıdır.

C ++ programlama dilindeki bu koşullu ifadelere Boolean ifadeleri olarak da adlandırılır. Bu tür ifadeleri oluştururken karşılaştırma operatörleri kullanılır:

C++ ta operatörler	Anlamı	Matematik operatörü
==	eşittir	=
!=	eşit değildir	≠
>	büyüktür	>
<	küçüktür	<
>=	büyük yada eşittir	≥
<=	küçük yada eşittir	≤

Tablo 1: Karşılaştırmalı operatörler

4.11.1 İki olasılık arasından seçim yapma yapısı

Koşullu ifadelere sahip programlar oluştururken, en yaygın olarak **iki seçenekli yapı kullanılır.** İfadenin değerine bağlı olarak, bu yapıda iki olasılık arasından bir seçim yapılır. İfade **doğru** veya **yanlış** değerine sahip olabilir, yani ifadenin koşulu doğruysa bir komut çalıştırılır ve doğru değilse başka bir komut çalıştırılır veya program burada sona erer.

Yukarıdakilere dayanarak, koşulun karşılanıp karşılanmadığına bağlı olarak iki tür ifade dallanması olduğunu belirleyebiliriz: **tek** ve **çift**.

Tek dallanma, ödev koşulu karşılanırsa, ardından bir emir yürütmek için ve koşul karşılanmazsa, program sona erdiğinde, yani "**eğer koşul, sonra komut**". Bu amaçla, **if ... then** ifadesi uygulanır.

Örneğin: kullanıcı yıl girdiğinde, program onun reşit olup olmadığını göstersin.

```
1  // if...else ornekler
2  #include <iostream>
3  #include <cstdlib>
4  using namespace std;
5
6  int main()
7  {
8      int godini;
9      cout << " Kolku godini imate? " << endl;
10     cin >> godini;
11
12     if (godini <=18)
13         cout << " Maloletni ste!" << endl;
14     else
15         cout << "Polnoletni ste!" << endl;
16
17     system ("pause");
18     return 0;
19 }
```

Şekil 1: If ... Then ifadesi

Önemli!

İf ifadesinin ardından noktalı virgül eklenmez çünkü sıradaki komutlar bu ifadede yürütülmeldir. Bir noktalı virgül ekleyerek ifadenin sonunu işaretleyeceğiz.

Ana programdan tamsayı yani **int** değer türünde **godini** değişkeni tanımlandığını görebiliriz. "Kolku godini imate (**Kaç yaşındasınız**)?" yazısı görüldükten sonra, kullanıcı klavyeden verileri girer ve **Enter** tuşuna basar. Daha sonra girilen verinin onsekizden (18) küçük veya ona eşit olup olmadığı kontrol edilir ve bu koşul karşılanırsa ekranda

“Reşit değilsiniz!” Mesajı görüntülenir. Koşul yerine getirilmezse, yani on sekizden (18) büyük veriler girilirse, herhangi bir işlem yapılmayacaktır.

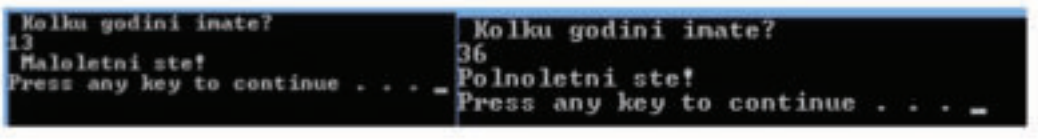
Aslında, **çift dallanma**, bir koşul yerine getirilirse, o zaman tanımlanan siparişi yürütmek ve bu karşılanmazsa, başka bir emri yani “**koşul doğru ise sipariş1, aksi takdirde sipariş2**” yürütmek anlamına gelir. Bunun için **If ... else** deyimi uygulanır.

Örneğin daha önce oluşturduğumuz ödeve başka bir komut tanımlayarak devam edelim.

```
1   // if...then naredba
2   #include <iostream>
3   #include <cstdlib>
4   using namespace std;
5
6   int main()
7   {
8       int godini;
9       cout << " Kolku godini imate? " << endl;
10      cin >> godini;
11
12      if (godini <=18)
13          cout << " Maloletni ste!" << endl;
14
15      system ("pause");
16      return 0;
17  }
18
```

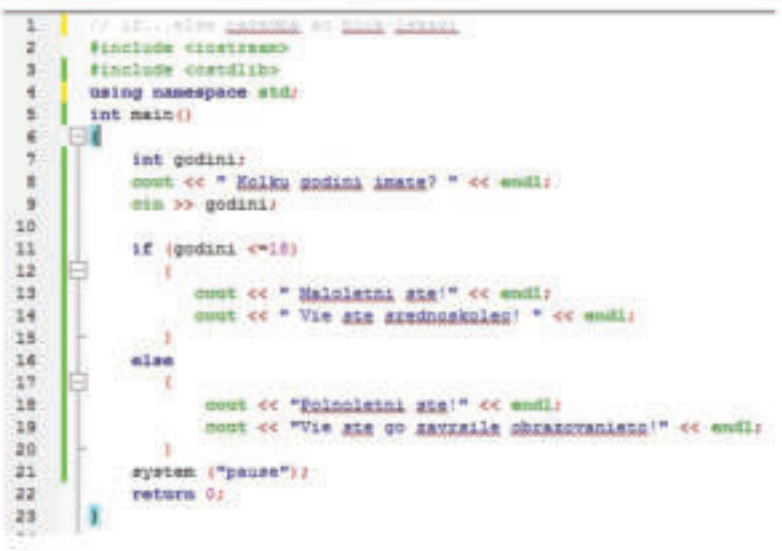
Şekil 2: If ... Else komutu

Bu nedenle, “**godini**” verisi girilirken, program girilen verinin onsekiz (18) ‘den küçük veya ona eşit olup olmadığını kontrol eder. Koşul karşılanırsa, yani girilen veri sıfır (0) ile onsekiz (18) aralığındaysa ekranda “**Maloletni ste! Reşit değilsiniz**” mesajı görüntülenecektir. Koşul yerine getirilmezse, yani girilen veriler on sekiz (18) rakamından büyükse, ekrana “Polnoleten ste! Reşitsiniz” yazsın:



Şekil 3: If ... else yürütmenin sonucu

If ... else deyimini kullanırken, komut birden çok ifade içerebilir. Bu komut yazma şekline bir ifade bloğu denir ve bunlar parantezler arasına yerleştirilir. Örneğin, çalıştığımız programda blok ifadeler yazmak için:



Şekil 4: If ... else içindeki blok ifadeler

Bu, "**godini**" değişkeni için veri girildikten sonra, girilen verilerin onsekizden (18) **küçük veya ona eşit** olup olmadığının kontrol edileceği anlamına gelir. Koşul yerine getirilirse, ekranda birbiri ardına iki mesaj görüntülenir ve koşul yerine getirilmezse, koşulun yerine getirilmesine bağlı olarak ekranda yeniden iki komut daha görüntülenir. İfade bloğu her zaman iki parantez arasında yazılır.

4.11.2 İki olasılık arasından seçim yapabilen programların geliştirilmesi

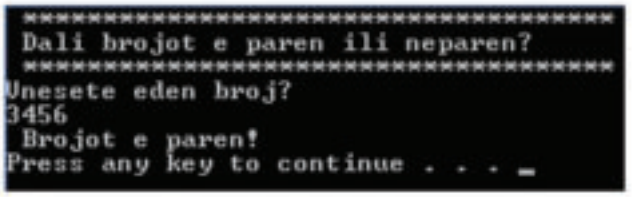
Bir koşul belirleyeceğimiz birkaç program oluşturalım. Verileri girerek koşulun karşılanıp karşılanmadığını kontrol edeceğiz ve cevaplara göre komutlar çalıştırılacak.

Ödev

Matematiksel formülü kullanarak bir üçgenin alanını hesaplayın: P = (a * h) / 2

Şekil 5: Sayının çift mi yoksa tek mi olduğunun kontrol edilmesi

Bu programda klavyeden sayı girildikten sonra çift veya tek sayı olup olmadığı kontrol edilir. Matematik kuralına göre "ik(2) sayısına bölünen her sayının sonucu tamsayı olur ve kalansızdır, o zaman çift sayıdır", koşulu belirlerz. Burada bölümün gerkalanını gösteren "%" (mod) operatörünü uyguluyoruz. Girilen sayının ikye bölünmesi sıfıra eşit bir kalana sahip olmalı, ardından ekrana "Brojot e paren! - Çift sayı" Mesajı yazılmalıdır. Koşul karşılanmazsa, "Brojot e neparen! - Tek sayı" Mesajını yazılsın. Programın yürütülmesinin sonucu aşağıdakpencerede görüntülenir:



Şekil 6: Sayının çift mi yoksa tek mi olduğunu kontrol etmek için programın sonucu

Ödev

İksayı gireceğimiz bir program oluşturun. Program bu iksayıdan hangsayının büyük olduğunu tespit edecek.

```
1   // Koi broj e pogolem
2   #include <iostream>
3   #include <cstdlib>
4   using namespace std;
5   int main()
6   {
7       int broj1, broj2;
8       cout << " ************************************ " << endl;
9       cout << " Koi broj e pogolem? " << endl;
10      cout << " ************************************ " << endl;
11      cout << "Unesete go prviot broj: " << endl;
12      cin >> broj1;
13      cout << "Unesete go vtoriot broj: " << endl;
14      cin >> broj2;
15      if (broj1>broj2)
16            cout << " Prviot broj e pogolem od vtoriot!" << endl;
17      else
18            cout << " Vtoriot broj e pogolem od prviot!" << endl;
19      system ("pause");
20      return 0;
21  }
```

Şekil 7: Hangi sayının daha büyük olduğunu kontrol eden bir program

Programda, iksayıyı girdikten sonra, ilk sayının ikinciden büyük olup olmadığını kontrol etmek için bir koşul oluşturduğumuz **If ... else** ifadesinkullanıyoruz. Koşul yerine getirilirse ekrana birincisayının ikinciden büyük olduğu mesajı yazılır ve karşılanmazsa ikincisayının birinciden büyük olduğu mesajı yazılır. Kontrolün sonucu aşağıdakgibidir:

```
**************************************
 Koj broj e pogolen?
**************************************
Unesete go prviot broj:
345
Unesete go vtoriot broj:
123
 Prviot broj e pogolen od vtoriot!
Press any key to continue . . .
```

Şekil 8: Hangi sayının daha büyük olduğunu kontrol etmenin sonucu

Ödev

Bir sayının pozitif mi yoksa negatif mi olduğunu kontrol edecek bir program oluşturun!

```
1  // [illegible]
2  #include <iostream>
3  #include <cstdlib>
4  using namespace std;
5  int main()
6  {
7      int broj;
8      cout << " ************************************ " << endl;
9      cout << " Dali brojot e pozitiven ili negativen? " << endl;
10     cout << " ************************************ " << endl;
11     cout << "Vnesete eden broj?" << endl;
12     cin >> broj;
13     if (broj<0)
14           cout << " Brojot e negativen!" << endl;
15     else
16           if (broj==0)
17                cout << "Brojot ne e nitu pozitiven nitu negativen!" << endl;
18           else
19                cout <<"Brojot e pozitiven" << endl;
20
21     system ("pause");
22     return 0;
23 }
```

Şekil 9: if ... else ile sayının pozitif mi, negatif mi yoksa sıfıra eşit mi olduğunun kontrolü

Bu programdan, klavyeden bir sayı girdikten sonra, girilen sayı sıfırdan küçükse bir koşul oluşturduğumuzu, ardından negatif bir sayı olduğuna dair bir mesaj yazdığımızı fark edebiliriz. Aksi takdirde, koşul yerine getirilmezse, girilen sayının sıfıra eşit olup olmadığını kontrol etmeye devam edin ve ardından bu koşulun ekranda sayının ne pozitif ne de negatif olduğunu göstermesi sağlanır, aksi takdirde girilen sayının pozitif olduğunu belirten bir mesaj yazılır. Burada, önceki koşulun yerine getirilmesine bağlı olarak yürütülen **If ... else ifadesini iki kez** uyguladığımızı not ediyoruz. Bu tekniğe **gömme tekniği** denir. Hesaplamanın sonucunu ekranda görüyoruz:

```
**************************************
 Dali brojot e pozitiven ili negativen?
**************************************
Unesete eden broj?
2346
Brojot e pozitiven
Press any key to continue . . .
```

Şekil 10: Sayının pozitif, negatif olup olmadığının kontrol edilmesinin sonucu

Ezberle!

İki olasılık arasından seçim yapılırken koşullu ifadeler kullanılır. Mantıksal veya Boolean ifadeleri olarak adlandırılırlar ve bunları oluşturmak için oşoullu peratörlerini kullanırlar. İki olasılık seçeneği içeren program yapısı tek dallı ve çift dallı olabilir. İlk durumda if ifadesi kullanılırken, ikinci durumda if ... else ifadesi kullanılır. Yapısındaki bu ifadeler birden çok ifade içerebilir ve süslü parantez içinde yazılır.

Sorular

1. Koşullu ifadeleri tanımlayın!
2. Karşılaştırma için kullanılan operatörleri listeleyin!
3. İki olasılık arasından seçim yapılabilen bir programda dallanma ne olabilir? Aralarındaki fark nedir?
4. If ve If ... Else! Sözdizimini yazın!

4.12 Bir koşul karşılanana kadar bir döngüde tekrarlama yapısı

Şimdiye kadar farklı yapıya, farklı ifadelere, farklı amaçlara ve çözümlere sahip birçok program oluşturduk. C ++ programlama dilinde programlar oluştururken, ifadeler sırayla birer birer yürütülür. İfadelerin bu şekilde yürütülmesi, **fiziksel yürütme** sırası olarak adlandırılır.

If ... Else ifadesini kullanarak programlar oluşturduğumuzda, tanımlanan koşulun yerine getirilmesine bağlı olarak ifadelerin yürütme sırası değişti. Çoğu kez bazı ifadelerin tekrarlanması, yani birkaç kez çalıştırılması gerekir. İfadelerin bu tekrarı bir **döngüdür** ve bu yapılara **döngüsel yapılar** veya **tekrar için yapılar** denir. İfadenin bir döngüdeki yürütme sırasına **mantıksal sıra** denir, çünkü koşulun yerine getirilmesine bağlıdır.

Buna göre döngü, bir veya bir grup koşul karşılanana kadar bir veya daha fazla talimatı birçok kez çalıştırır. Döngüleri çalıştıran en yaygın kullanılan ifadeler şunlardır: while, do - while ve forcycles.

Yineleme yapılarını uygularken iki durum mümkündür:

- döngünün kaç kez tekrarlanacağı önceden bilinmektedir;
- tekrarların sayısı bir koşula bağlıdır ve bu sayı önceden bilinmemektedir. Koşul, döngünün başında veya döngünün sonunda bulunabilir.

While döngüsü, belirli bir koşulun yerine getirilmesiyle biten en basit döngüdür, yani while komutu, koşul test eder, koşul yerine getirilene kadar bir dizi komutu çalıştırır. Sözdizimi aşağıdaki gibidir:

```
While (koşul)
{
komut 1;
komut 2;
komut 3;...
}
```

Görüntülenen sözdiziminden, **while** komutu **döngü -loop** adı verilen bir kontrol yapısıdır. Komut, yani döngü her geçildiğinde çallıştırılması gereken komutlar, **döngünün gövdesi** olarak adlandırılır. Döngünün yürütülmesine başlamadan önce, ilk olarak koşulun karşılanıp karşılanmadığı kontrol edilir. Koşul yerine getirilirse, ifadelerin çalıştırılması ile başlar ve koşul doğru değilse hiç başlamaz.

Örneğin:

```
1  // ...
2  #include <iostream>
3  #include <cstdlib>
4  using namespace std;
5  int main()
6  {
7      int i=1;
8      while (i<=10)
9      {
10         cout << i <<"\n";
11         i++;
12     }
13     system ("pause");
14     return 0;
15 }
```

Şekil 1: Bir (1) ile on (10) arasındaki sayı dizisini yazmak için program

Programda, ana işlevde "**I**" değişkeninin "tamsayı" tipiyle tanımlandığını ve bir (1) değeriyle başlatıldığını fark ettik. **While** yineleme yapısını kullanarak, while döngüsünde verilen koşul karşılanana kadar sıra numaralarını bir (1) ila on (10) arasında yazılsın. Ekranda aşağıdakiler görüntülenecektir:

```
1
2
3
4
5
6
7
8
9
10
Press any key to continue . . .
```

Şekil 2: Bir (1) ile on (10) arasındaki sayı dizisinin yazılması

Not!

Bu programda sayı dizisi dikey olarak gösterilir. Sayıları yatay olarak yazdırmak için, "\ n" ifadesini silip tırnak içinde virgül ekliyoruz.

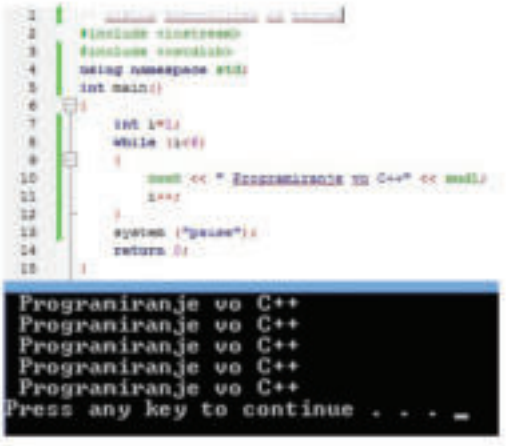
Ekranda belirli bir sayıya kadar karakterleri görüntüleyen benzer bir program aşağıdaki programda gösterilmiştir:

Şekil 3: Bir karakter dizisi yazma

Komutlar sırasında **while** döngüler kullanan şunlar olabilir:

- Sayaç kontrollü döngüler;
- olay kontrollü döngüler.

Sayaç kontrollü döngüler, döngüyü kontrol eden bir değişken kullanır. Bu amaçla, sayacın kontrol değişken başlatılır ve her yürütme döngüsünde sayaç artırılır. Örneğin:



Şekil 4: Sayaç kontrollü döngü

Ödeve göre, "C ++ 'da Programlama" cümlesi koşul karşılanana kadar yazılacaktır: **beş kez yazmak.** Her turda, sayaç bir (1) artar.

Olay kontrollü döngülerde sonlandırma koşulu, döngünün gövdesi çalışırken meydana gelen bir olaya bağlıdır. Örneğin: girilen sayıların çarpımını hesaplamak için ve giriş klavyeden sıfır (0) girilerek duracaktır:

```
#include <iostream>
#include <cstdlib>
using namespace std;
main()
{
    int broj, p=1, br=1;
    cout << "*************************************************************"<< endl;
    cout << "Vnesot na podatoci i presmetkata ke zavrsat so vnes na 0!" << endl;
    cout << "*************************************************************"<< endl;
    cout <<" Vnesete eden broj: ";
    cin >> broj;
    while (broj!=0)
    {
        p=p*broj;
        br++;
        cout << "Vnesete nov broj: ";
        cin >> broj;
        cout << p << endl;
    }
    if (br==0)
    p=0;
    cout << "Proizvodot na vnesenite broevi e: " << p << "." << endl;
    system ("pause");
    return 0;
}
```

Şekil 5: Olay kontrollü döngü

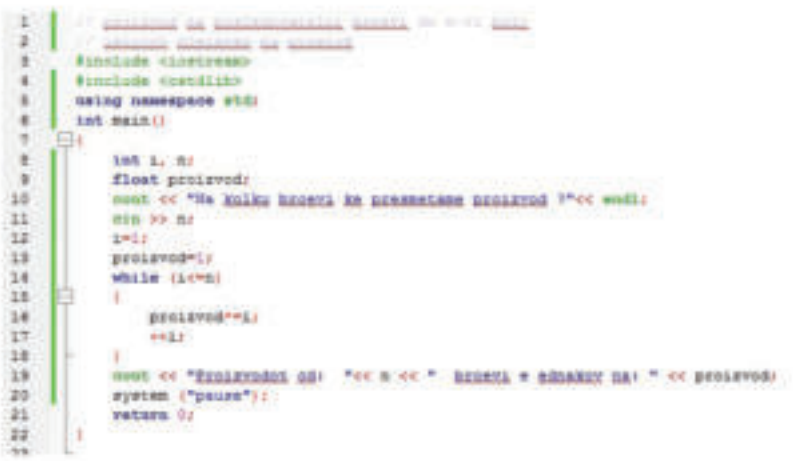
Bu While döngüsü programı klavyeden sıfır (0) girilene kadar çalışacaktır. Ve bu, sıfır (0) girmenin döngüyü sona erdirecek bir olay olduğu anlamına gelir. Bu nedenle bu tür döngülerin olaylar tarafından kontrol edildiği söylenir. Kodun çalıştırılmasının sonucu aşağıdaki görüntüde gösterilmektedir:

Şekil 6: Döngü yürütme sonucu

4.12.1 Koşul sağlanana kadar tekrarlanacak yapıda basit bir program yapmak

Bir koşulu sağlamak için yinelemeli yapılarda, koşul **döngünün başında** veya **döngünün sonunda** tanımlanabilir. İlkinde, ilk önce koşulun karşılanıp karşılanmadığı kontrol edilir, böylece ifade dizileri yürütülebilir ve ikincisinde, koşul karşılanana kadar ifadeler yürütülür. Örneğin, n'inci sayıya kadar ardışık sayıların çarpımını üç şekilde hesaplayacağız:

Birinci yol: koşul, döngü tekrarlama ifadesinin başlangıcındadır:



Şekil 7: Koşul döngü başlangıcında tanımlanır

İkinci yol: koşul yerine getirilene kadar ifade dizileri yürütülecektir:

Şekil 8: Koşul karşılanıncaya kadar ifadelerin yürütüldüğü program

Ödev

Aşağıdaki programları oluşturmak için koşul karşılanana kadar döngü yineleme yapılarını uygulayın:

1. n'inci sayıya kadar çift sayıları ekranda yazsın!
2. n'inci sayıya kadar olan tek sayıların toplamıhesaplansın!
3. n'inci sayıya kadar çift sayıların çarpımı hesaplansın!

Ezberle!

İfadelerin yürütülme sırası hem fiziksel hem de mantıksal olabilir. İfadelerin birkaç kez tekrarlanması bir döngüdür ve yapılar tekrar için döngüsel yapılardır. Döngüler için en sık kullanılan ifadeler şunlardır: while, do-while ve for. While kullanılırken, önce koşul kontrol edilir ve ardından ifadeler yürütülür. Do-while'da, koşul karşılanana kadar ifadeler yürütülür.

Sorular

1. Tekrarlama yapıları nelerdir? Tanımlayın!
2. Döngüler için en sık kullanılan ifadeler nelerdir?
3. while ve do-while sözdizimlerini listeleyin!
4. while ve do-while uygularken döngüleri nasıl kontrol edersiniz?
5. Döngünün kaç kez tekrarlanacağı bilinir mi? Açıklamak!

TEKRARLAYALIM! UYGULAYALIM!

Ödev 1

Aşağıdaki ekran görüntüsünü alacak bir program yazın:
İlk ders: Bilişim Teknolojileri
C++ 'da programlama

Ödev 2

Aşağıdaki komutları uyguladıktan sonra a ve b değişkenlerinin değerleri nelerdir?

```
a = 15
b = 23
a = a + b
b = a-b
a = a-b
```

Ödev 3

Aşağıdaki kodu çalıştırdıktan sonra ekranda ne görüntülenecek:

```
int a, b,c;
cin >> a;
cin >> b;
cin >> c;
cout << a << "," << b << "," << c << ". " << endl;
```

kullanıcı aşağıdaki değerleri girerse: art arda beş (5), altı (6) ve yedi (7)?

Ödev 4

Aşağıdaki komut dizisini uyguladıktan sonra ekranda ne görüntülenecek?

```
int a=8;
int b=10;
cout << (a+b)/2
```

Ödev 5

Bir sonraki döngü kaç kez gerçekleştirilecek?

```
int i=2
Do
{
cout << i << " " ;i ++;}
While (i<=10)
```

Ödev 6

Bir dikdörtgenin alanını hesaplamak için bir program yazın. Kenarların boyutlarını ve dikdörgenin alanını ekranda gösterecek. Bir dikdörtgenin alanını hesaplamanın formülü şudur: p= a*b.

Ödev 7

İki sayının aritmetik ortalamasını hesaplayan bir program yazın. Sekiz sayının aritmetik ortalaması nasıl hesaplanır?

Ödev 8

Girilen sayının çift mi yoksa tek mi olduğunu kontrol eden program oluşturun.

Ödev 9

Bir karenin çevresini hesaplayan bir program oluşturun. Girilen “a” değeri pozitiv sayıysa karenin çevresşnş ekrana yazdıracak aksi takdirde “Hesaplama yapılamaz! Pozitif bir sayı girin! “ ekrana yazsın.

Ödev 10

Do - while döngüsünü kullanarak, n’inci sayıya kadar olan çift sayıların toplamı hesaplansın.

Ödev 11

Klavyeden sıfır (0) rakamı girilene kadar, klavyeden ard arda girilen sayıların toplamı hesaplansın.

Çevrimiçi yaşam

Web Bloglarına Giriş
Blog kavramı ve uygulaması
Bir blog oluşturmak
Blog içeriği hakkında yorum yapma
TEKRAR EDELİM! UYGULAYALIM!

5. Web günlüklerine (bloglar) giriş

Hatırlayalım!

Programlar nelerdir? Programlama için tanımlar verin! Bir program nasıl geliştirilir? Hangi programlama dillerini biliyorsunuz?

Modern toplum ve kitlesel küreselleşme eğilimi, **internet teknolojisinin** eğitim sürecinde ve aynı zamanda genel olarak günlük yaşamda uygulanması ihtiyacını empoze etmektedir. İnternet teknolojisinin uygulanması, kullanıcıların iletişim, mesajlaşma, bilgi paylaşımı, çevrimiçi alışveriş, resim indirme, müzik, video içeriği ve gibi çeşitli hizmetlerin kullanmasına olanak tanır.

Bu nedenle **İnternet**, iletişim kurmak ve bilgi alışverişi yapmak için çeşitli şekillerde birbirine bağlanan, dünyanın dört bir yanındaki milyonlarca bilgisayardan oluşan dünyanın en ünlü ağıdır. Bununla birlikte, İnternette en çok kullanılan hizmet iletişimdir, yani e-posta yoluyla veri ve bilgi alışverişi, çevrimiçi iletişim hizmetleri, sosyal ağlar, son zamanlarda **web günlükleri** veya **blogların** uygulanmasıdır. Bu bir iletişim olduğundan, kullanıcı internetin etik kullanım kurallarına uymalıdır:

Kişisel verileri ve üye verilerini paylaşmayın

- Ailenizin adı ve soyadı, adresi, telefonu veya çevrimiçi yabancıların fotoğrafları gibi bilgileri paylaşmayın, çünkü bilgisayarın diğer tarafında kimin olduğunu bilmiyoruz. Çevrimiçi iletişim kuran herkes iyi niyetli değildir;
- Ebeveyinlerinin izni olmadan bir şey satın almak veya sipariş etmek için asla kredi kartı numarasını çevrimiçi olarak girmeyin;
- Kötüye kullanılmamak için çevrimiçi hizmetlere erişmek için kullandığınız şifreleri açıklamayın;
- Çevrimiçi sohbet ederken saldırgan ve müstehcen kelimeler kullanmayın;
- Çevrimiçi iletişim kurarken, sizinkinden farklı olsalar bile diğer insanların görüşlerine saygı gösterin. Çevrimiçi iletişimde dostça olun;
- Çevrimiçi metin yazarken, büyük harflerle yazmayın, büyük harflerle yazmak, bağırmak gibidir.

Son zamanlarda sosyal ağların hayata geçmesiyle birlikte kullanıcının internette yayınladığı paylaşımlar sonucunda **dijital parmak** izi terimi karşımıza çıkıyor. Günlük sörf yaparken korunduğumuza ve anonim olduğumuza inanıyoruz, ancak öyle değil. Sosyal ağ kullanıcılarının faaliyetlerinin birçok kişi tarafından izlendiğini gösterir.

Kullanıcı tarafından üstlenilen faaliyetler yoluyla: Sayfaları ve gönderileri beğenmek, bağlantıları, durumları, resimleri, videoları paylaşmak, içerik hakkında yorum yapmak gibi etkinlikler kullanıcının dijital **parmak izi oluşturulur,** yani her bir kullanıcının faaliyetlerinin bir veritabanı oluşturulur, bunun aracılığıyla ne tür bir kişi olduğu, ilgi alanları ve ilgi alanları ve onu nasıl etkileyeceği öğrenilir.

Bu nedenle, **dijital parmak izi, yanlışlıkla veya kasıtlı olarak geride bırakılan olumlu ya da olumsuz herhangi bir internet bilgisidir.**

Dijital parmak izinin artıları ve eksileri vardır, ancak kontrol edilmeleri gerekir. Örneğin, dijital parmak izinin olumlu tarafı, internette bir tarif aradığımızda ve bu tarif aracılığıyla birçok yemek sayfasına ulaşabiliyoruz. Ancak internetin sorumsuzca kullanılmasının dezavantajları var. Örneğin, özel olduğunu düşündüğümüz, herkese açık hale getirilen fotoğraflar, belirli zamanlarda kişisel veriler iyi korunmuyor vb. Bu nedenle, çevrimiçi bir şey göndermeden önce, kullanıcının gönderinin faydaları ve kendisine nasıl zarar verebileceği hakkında dikkatlice düşünmesi gerekir. Her gönderi öncesinde, kullanıcı göndermeden önce doğru kararı vermek için pozitif ve negatif değerleri tartmalıdır. Pozitif bir dijital parmak izi için en yaygın ipuçları şunlardır:

- sosyal ağlardan veya e-postadan çıkarken çıkış yapın;
- şifreleri söylemeyin;
- şifreleri herkese açık olarak yazmayın;
- şifre oluştururken harf, sayı ve simge kullanın;
- arşivi sık sık silin;
- kişisel bilgileri çevrimiçi olarak paylaşmayın.

Bu konuda öğretmenler ve öğrenciler arasında bağlantı sağlayan, dersleri, alıştırmaları, videoları paylaşan ve bundan daha fazlasını öğrenmek isteyenler için ek materyaller ekleyen güçlü bir çevrimiçi araç olarak **web günlüklere**, yani **bloglara** odaklanacağız.
Blogların sunduğu işlevleri ve olanakları bir arama motoru yardımıyla anlamak için, farklı yazarlara ve farklı konulara ait blogları bulmaya çalışalım. Özellikle çalıştığımız konu olan **bilişim** içeriğe sahip birçok blog var.

Anahtar terimler

Web günlük , blog, blogger, giriş, kullanıcı adı, şifre, yorum, kayıt, dijital baskı.

5.1 Blog kavramı ve uygulaması

Bir **blog** veya **web günlük,** içeriğin kronolojik sırayla görüntülendiği, yani en son haberler veya içeriğin önce görüntülendiği ve geri kalanın aşağıya gittiği çevrimiçi bir günlüktür. En son gönderilerden daha eski gönderilere giderek sıralanır.

Blog kelimesi, ağ anlamına gelen **web** ve giriş anlamına gelen **log** kelimesinden gelir.

Aslında bloglar, kullanıcıları, iş gruplarını ve benzer düşünen web yöneticilerini birbirine bağlayan bir iletişim aracıdır.

Blogun yazarına **blogger** denir. Blogger yalnızca bir blog oluşturmakla kalmaz, aynı zamanda içeriğini sürekli olarak günceller ve düzenler. Blog içeriği oluşturma, güncelleme ve düzenleme, **blog oluşturma** olarak adlandırılır. Bir blog yazarı, bilgisayar ve İnternet teknolojileri hakkında bilgi sahibi olan ve belirli bir konuyu tartışmaya hazır olan herhangi biri olabilir.

Blog içeriği farklı olabilir: Metinler, notlar, tartışmalar, fotoğraflar, videolar, internet'teki diğer konumların adresleri vb. Blogcular ne yazmak, yorum yapmak veya blog yazmak istediklerini baştan bilirler. En yaygın bloglama konuları şunlardır:

- eğitim kılavuzları ve eğitim konuları;
- ünlüler, biyografileri ve başarıları;
- spor;
- sağlık ve güzellik;
- ebeveynler ve çocuklar için konular;
- bilgisayar oyunları;
- evcil hayvanlar ve bakımı;
- tarifler;
- siyaset;
- farklı alanlara yönelik bir rehber;
- yaşam deneyimleri;
- hizmetler ve ürünler;
- seyahat;
- moda;
- ev dekorasyonu vb.

Tabii ki, kullanıcıların ilgisini çeken başka birçok konu var. Blog yazarı, hangi konunun blog oluşturacağına karar verir. Çoğu zaman, blog yazarları aşina oldukları, önceden bilgi sahibi oldukları konuları, bilgilerini genişletmek ve derinleştirmek için tartışmaların yardımıyla, kamuya konu veya konuları tartışan belirli bir hedef gruba yöneliktir.

Tartışmalar eğlenceli ve öğretici olabilir ve internet nüfusu için büyük önem taşıyan içerikler de bulunabilir.
Blog oluşturma olanakları hakkında bilgi edinmek için bir blog oluşturup, blog topluluğunun bir parçası olacağız.

Ezberle!

Bir web günlük veya blog, içeriğin ters kronolojik sırada görüntülendiği, elektronik dergi biçimindeki bir internet sitesidir. Blogger, web günlüğün veya blogun yazarıdır. Blog oluşturma, düzenleme, yorum yapma ve sürdürme sürecidir. Çevrimiçi iletişimde etik davranış kurallarına bağlılık, blog yazma ve dijital parmak izi için de geçerlidir. Dijital parmak izi, internette çeşitli hizmetleri uyguladıktan sonra bıraktığımız izdir.

Sorular

1. Blog nedir?
2. Bloger nedir?
3. Bir blogun yazarı kim olabilir?
4. Bloglama nedir?
5. Blog oluştururken hangi konular kullanılabilir?
6. Blogun içeriğini kim yazıyor?
7. Blogların etik kullanımı için birkaç kural listeleyin!
8. Blogları okul amaçları için bir araç olarak kullanabilir miyim? Nasıl?
9. Dijital parmak izi nedir?

5.2 Blog oluşturmak

Blog oluşturmak basit bir süreçtir. Özellikleri, özellikleri ve işlevleri uygulamanın en basit, en hızlı ve en ucuz yolu ücretsiz blog hizmetleridir. Bloglama için kullanılabilen birçok bloglama hizmeti vardır:

1.Blogger;
2.Drupal;
3. Ghost;
4. Joomla;
5. Magento;
6. Medium;
7. Squarespace;
8. Tumblr;
9. Typepad;
10. Weebly;
11. Wix;
12. WordPress.

En popüler blog hizmetleri **Blogger** ve **WordPress**'tir.

Blogger, **Google** paketinin bir parçası olduğu için en eski **Google** platformlarından biridir, kişisel **Google hesabımızı,** yani **Gmail**'i kullanarak ona kolayca erişebiliriz. Bu nedenle, Blogger'da blog yazmaya başlamak için önce kendi kullanıcı adımıza ve şifremize (**Gmail hesabına)** sahip olmamız gerekir.

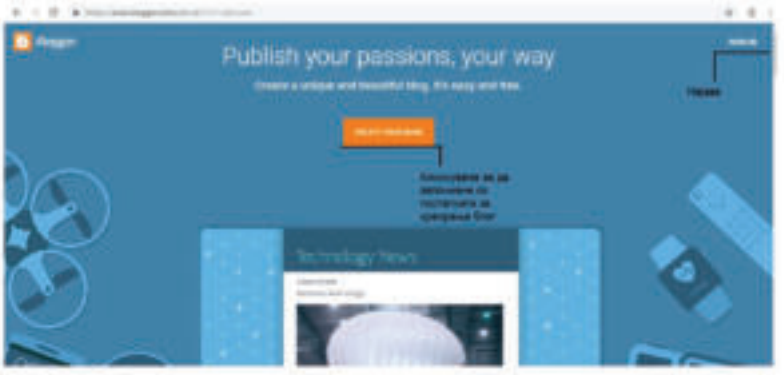
Şekil 1: Blogger logosu

Blogger çevrimiçi olarak barındırıldığı için, kullanıcı şuna benzer bir alt alan adı da alır: stead.blogspot.com.

Blogger özellikleri şunlar için kullanılabilir: günlük yazmak, resim göndermek, kendi kağıtlarınızı yazmak, haber portalı, kılavuzlar vb. Bir yazarın oluşturabileceği ve sürdürebileceği blog sayısında herhangi bir sınırlama yoktur. Ayrıca, kullanıcı blog ile reklam, çeşitli ürünlerin tanıtımı ve benzer yollarla para kazanabilir.

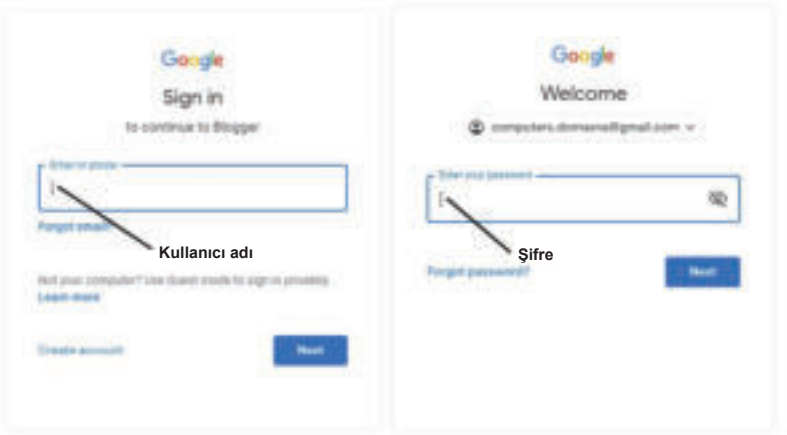
5.2.1 Blog oluşturma adımları

Ücretsiz Blogger hizmetinde blog oluşturma sürecini başlatmak için, önce bir Gmail hesabı oluşturmamız, ardından aşağıdaki gibi web tarayıcısını (Browser) başlatmamız gerekir: Google Chrome, Internet Explorer, Mozilla Firefox, Opera vb.. Tarayıcının adres çubuğuna, aşağıdaki penceren in göründüğü ücretsiz blog hizmetinin adresini yazıyoruz, www.blogger.com:



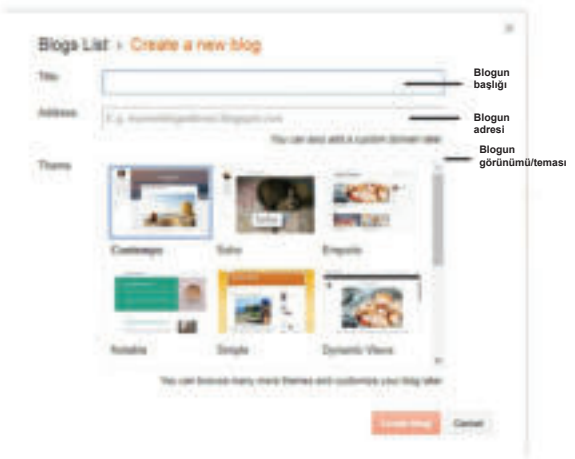
Şekil 2: Blog oluşturma sayfasına bakıldığında ana ekran

Kullanıcının kendi hesabı (**Account**) yoksa, **Giriş - Sign in** yap seçeneğine tıklayın ve **Gmail** kayıt işlemini başlatın veya eğer bir **Gmail** hesabı varsa, oturum açma penceresini açıp Yeni Blog Oluştur seçeneğini seçin.



Şekil 3: Oturum açma penceresi

Duyuru yapıldıktan sonra kullanıcı bloga bir isim, bir adres yazar ve bir konu, yani blogun görünümünü seçer. Bu, aşağıdaki görüntüdeki alanları doldurarak yapılır:



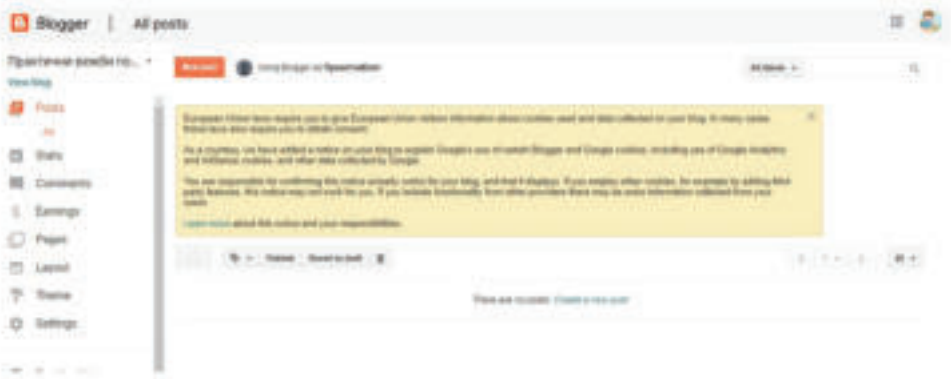
Şekil 4: Bir blog oluşturma

Not!

Seçenek olarak verilen konulara ek olarak, kullanıcı teklif listesinin dışında bir konu ekleyebilir.

Kullanıcı blog adresini girdiğinde, giriş kutusunun yanında sol tarafta bir mesaj belirir:

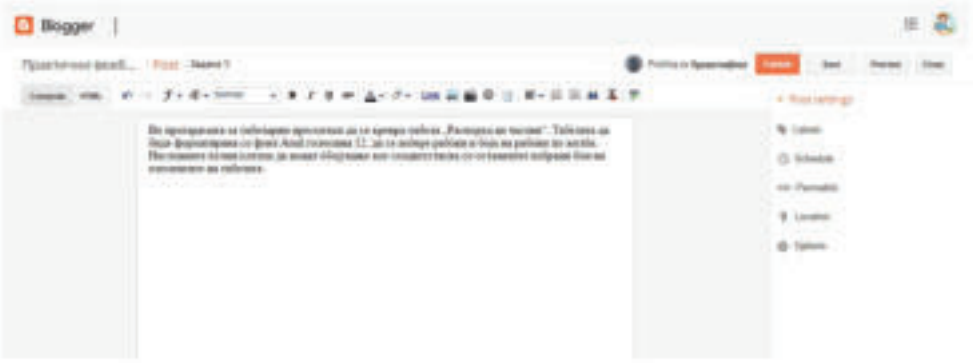
Checking Address Availability, yani girilen adresin mevcut olup olmadığı yani adresin başka bir kullanıcı tarafından kullanılıp kullanılmadığı kontrol edilir. Ardından, blogun içeriğiyle alakalı bir blog teması veya görünümü seçilir. Son olarak, **Create Blog - Blog Oluştur** düğmesi seçilir ve blogun Kontrol Paneline (**Dashboard**) erişilir. Bu kontrol paneli aracılığıyla, kullanıcı blogun içeriğini düzenler, yeni içerik ekler, yorumları yönetir vb.



Şekil 5: Bir blogun gösterge tablosu (Dashboard)

5.2.2 Blog içeriğini düzenleme

Gösterge Panosu (**Dashboard**) göründükten sonra, kullanıcı blogun içeriğini düzenlemeye başlayabilir. Bir gönderi (**Post**) oluşturmak için **New post - Yeni gönderi** düğmesine tıklanıp, aşağıdaki pencere görüntülenir:



Şekil 6: Bir blog yazısı (Post) oluşturma

Yayınla - Publish düğmesine tıklamayarak blogun içeriği yayınlanır.

Gönderi **Save - Kaydet** düğmesine tıklanarak kaydedilir, preview - önizleme düğmesi tıklanarak yayınlamadan öncekihalini görebiliriz, gönderiyi **kapat - close** seçeneğiyle yayından çekebiliriz. Pencerenin sağ tarafında, gönderinin (Post) ayarlarının (Post settings) ek düzenlemesi için seçenekler olduğunu fark ediyoruz, örneğin: yer imleri eklemek, yayının **Yayınla - Publish** düğmesine tıklayarak otomatik olarak görüntülenip görüntülenmeyeceğiveya kullanıcının bir tarih ayarlayıp duyuru zamanının planlanması, duyurunun yapıldığı yerin eklenmesive gibi ayarlar yapılabilir.

Not!

Gönderi tekrar yayınlandığında, Ana ekran olarak Gösterge Tablosu (Dashboard) görünür. Gösterge Tablosuna sadece blog yöneticileri erişebilir. Gönderi (Post) bağlantısına tıklayarak tüm yazı listesini görebilirsiniz.

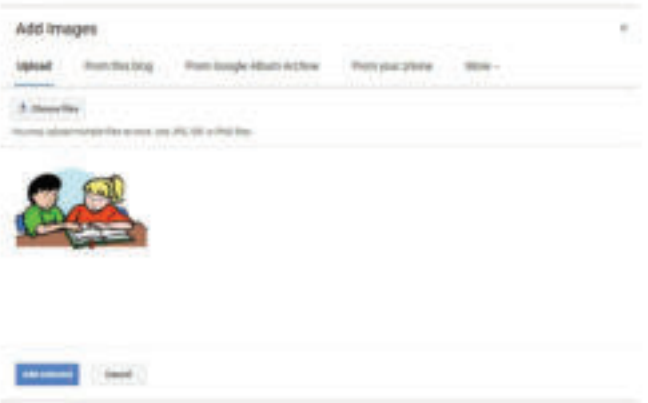
Bir gönderi oluşturma penceresinden, gönderinin içeriğini yazmanın, düzenlemenin ve biçimlendirmenin **MS Word** uygulaması gibi aynı olduğu fark edilir. Kullanıcı, yazı tipini, yazı tipi boyutunu, yazı tipi rengini, yazı tipi stillerini, bağlantılar, resim gibi ayarları ekleyip düzenleyebilir.

5.2.3 Bir bloga resim ve video ekleme

Gönderi oluşturma penceresi açıkken, önce imleci görüntünün ekleneceğiyere yerleştirin ve görüntü yükleme aracını seçin:

Şekil 7: Görüntü ekleme aracı

Ardından, bir görüntünün eklendiği aşağıdaki pencere açılır:



Şekil 8: Gönderi için bir resim seçin

Dosyaları Seç seçeneğiyle bilgisayardan yüklenecek bir görüntü seçilir. Seçilen görüntü, pencerenin sol alt köşesinde bulunan **Seçileni Ekle** düğmesine tıklanarak yayınlanır:

Şekil 9: Bir gönderideki görüntüyü düzenleme

Not!

Bir bilgisayardan resim eklemeye ek olarak, bir resim gönderisi oluştururken blogda zaten kullanılan resimleri, Google arşivinden resimleri ve telefonunuzdan resimleri de kullanabilirsiniz.

Pencereden, eklenen görüntüyü seçerken, aşağıdaki gibi düzenleme seçenekleri görünür: Görüntünün boyutlarını düzenleme, görüntüyü sıralama, görüntüye etiket ekleme, ek ayarlar ve silme.

Bir bloga video ekleme prosedürü, fotoğraf eklemeyle aynıdır. Bir gönderiye video eklemek için, videonun bilgisayarınıza indirilmesi ve kaydedilmesi gerekir. Yüklenecek videoyu seçin seçeneğinin tıklanması, video yüklenmeye başlar. Yükleme tamamlandığında Yayınla'yı - Publish tıklayın.

Deneyin!

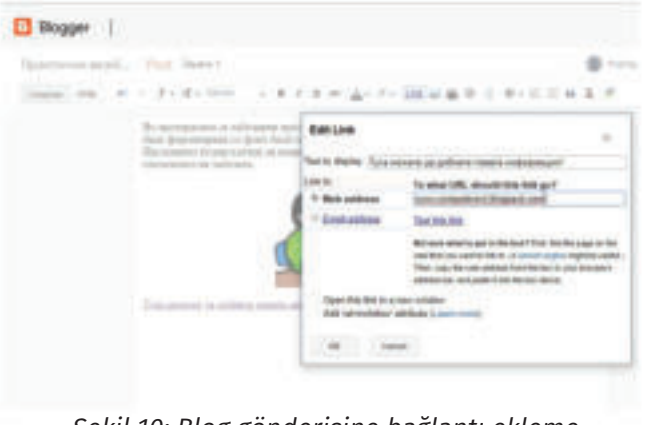
Programlar nelerdir? Programlama için tanımlar verin! Bir program nasıl geliştirilir? Hangi programlama dillerini biliyorsunuz?

5.2.4 Bloga bağlantı ekleme

Hatırlayalım!

Bağlantılar nedir? Bir metnin bağlantı olup olmadığını nasıl anlarsınız? Bağlantıların bazı özelliklerini yazın!

Blogdaki gönderide, kullanıcı ayrıca bağlantılar ekleyebilir, yani internette bulunan diğer sayfalara bağlantı verebilir. Bu amaçla, gönderide bağlantı olması gereken metni seçin ve bağlantı eklemek için pencereden bağlantı butonuna tıklayın:



Şekil 10: Blog gönderisine bağlantı ekleme

Bağlantıyı Düzenle (Edit Link) penceresinde, **Görüntülenecek Metin - Text to display** alanında, kullanıcının bağlantı yapmak istediği seçilen metin görünür. Ardından **Web Adresi** butonu seçilir ve alana kullanıcının bağlamak - linklemek istediği sayfanın adresi girilir. Bir e-posta adresine bağlanmanız gerekiyorsa, **E-posta Adresi** düğmesini seçin ve son olarak **Tamam** düğmesini tıklayın. Gönderiyi yayınlamak için **Yayınla'**yı tıklayın.

5.2.5 Blog temasını / düzenini değiştirme

Kullanıcı, blog içeriğini düzenlerken, blog için zaten bir tema seçmiş olmasına rağmen bunu değiştirebilir. Blog temasını / görünümünü değiştirme prosedürü Gösterge Panosu'nda (**Dashboard**) ve **Tema'yı** seçerek yapılır.

Şekil 11: Blog teması seçme / tema görünümü

Tema / düzen seçildikten sonra, blogun neye benzeyeceğini gösterir ve düzen kabul edilebilirse, temayı eklemek için **Ekle** düğmesine tıklıyoruz.

Son olarak, blog gönderilerini ve düzenini görüntülemek için kontrol panelinde **Blogu Görüntüle** bağlantısına tıklanır.

Ezberle!

En sık kullanılan blog hizmeti Blogger'dir ve Google'a aittir. Kullanıcı, bir oturum açma kullanıcı adı ve şifresi içeren bir Gmail hesabına sahip olmalıdır. Gönderiler, blogda yayınlanan içeriktir. Kontrol paneli aracılığıyla, blogun yazarı ayarlar oluşturabilir, gönderiler ekleyebilir, gönderileri düzenleyebilir, blogun konularını / görünümünü değiştirebilir, blog trafiğini inceleyebilir ve daha fazlasını yapabilir.

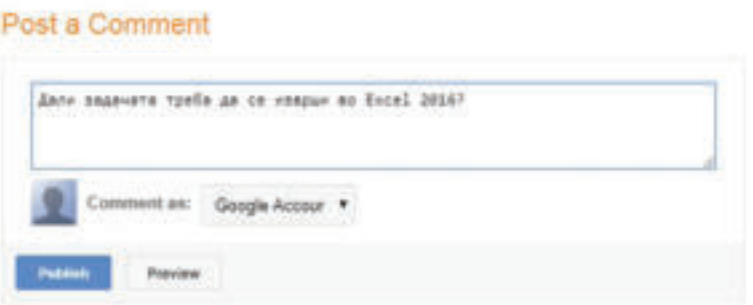
Sorular

1. Birkaç blog hizmetini listeleyin!
2. En popüler blog hizmeti hangisidir?
3. Kontrol paneli neden bir blog oluştururken kullanılır?
4. Gönderi nedir?
5. Gönderi oluşturma prosedürü nedir?
6. Blog içeriği ne olabilir?

5.3. Bloglardaki içeriği yorumlama

Konunun başında belirtildiği gibi, bloglar belirli bir temaya sahip toplulukların parçası olmamızı sağlar. Aslında, bilgi yaygınlaştırmaya, deneyimleri paylaşmaya, bir soruna çözüm bulmaya, yeni uygulamaları göstermeye ilgi göstersek herhangi bir konudaki tartışmaya katılabiliriz. Belirtilen nedenlerden ötürü gönderilere, yani blogdaki yazılara yorum yapabiliriz.

Blog içeriği hakkında yorum yapabilmek için öncelikle web tarayıcısının adres çubuğuna blogun adresini girerek blog içeriğini değerlendirmeye çağırmamız gerekir. Bunu yaparken içeriğini dikkate alıyoruz. Bir gönderiye yorum eklemek için **Yorum**'a tıklayın ve resimde gösterildiği gibi bir yorum formu görünecektir:



Şekil 1: Bir blog gönderisine yorum ekleme

Yorumun metni boş alana yazılır ve yorumun yapılacağı bir profil seçilir. Ardından iki seçenek vardır: **Yayınla** ve **Önizle**. Yayınla seçilirse yorum, gönderinin içeriğinin altında yayınlanır, Önizleme ile yorum gözden geçirilir.

Blog yöneticisi veya yaratıcısı, yayınlanan yorumlara yanıt verebilmek veya tartışmaya katılabilmek için, önce kontrol panelini (**Dashboard**) ve **Yorum** seçeneği aracılığıyla ardışık bir inceleme yapar.

Şekil 2: Blog içeriğiyle ilgili yorumları görüntüleyin

Yoruma tıklayarak, blog yöneticisinin yoruma yanıt vermesine olanak tanıyan bir **Yanıt** seçeneği mevcuttur. Boş alana cevap yazılır ve **Yayınla**'ya tıklanır.

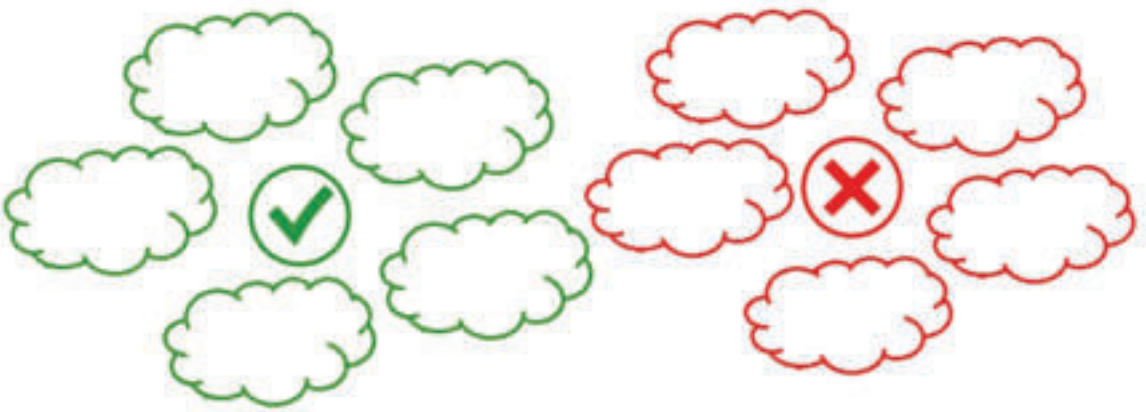
Ezberle!

Blog yazarken, hem kullanıcılar hem de yönetici yorum ekler. Yorumlar aracılığıyla kullanıcılar bir konuşma başlatır, deneyimlerini paylaşır, bilinmeyen şeyler hakkında sorular sorar, soruları cevaplar. Yorumlar, farklı bir konuda topluluğun parçası olmasını sağlar.

TEKRAR EDELİM! UYGULAYALIM!

Ödev 1

İnternet hizmetlerinin etik kullanımı için izin verilen kuralların yeşil ve yasa dışı olanların kırmızı ile yazılması için aşağıdaki diyagramı doldurun.



Ödev 2

Olumlu bir dijital parmak izi bırakmak için ipuçlarını yazın.

- _____
- _____
- _____

- _____
- _____
- _____

Ödev 3

Web günlüğün ne olduğunu açıklayın!

_____

_____

_____

Ödev 4

Doğru ifadeler elde etmek için cümleleri tamamlayın.

Blogları düzenleyen kişilere ----------

ve blog düzenleme işlemine ---------- denir.

Ödev 5

Haberler / içerikler web günlüğünde şu şekilde yayınlanır:

a) blogda yayınlanan resim sayısına göre,
b) özel bir sıra olmadan,
c) kronolojik sırayla yani yayın tarihin e göre.

Ödev 6

Bir bloglama sürecine başlamadan önce,

a) Kullanıcının bir e-posta adresine sahip olması gereklidir, örn. e-posta;
b) kullanıcının bir e-posta adresi olabilir, ancak bu mecburi değildir;
c) bir e-posta adresi gerekli değildir.

Ödev 7

Sahibin (blogun yazarı) kullanıcı ismi şu şekilde kullanılır:

a) imza için isim,
b) bloga giriş için isim,
c) tartışma konusu.

Ödev 8

Blog yazıları şöyle adlandırılır:

a) durumlar;
b) gönderiler;
c) tweetler.

Ödev 9

Blog ziyaretçisi, ekle seçeneği ile bir veya daha fazla blog yazısı için mesaj bırakabilir:

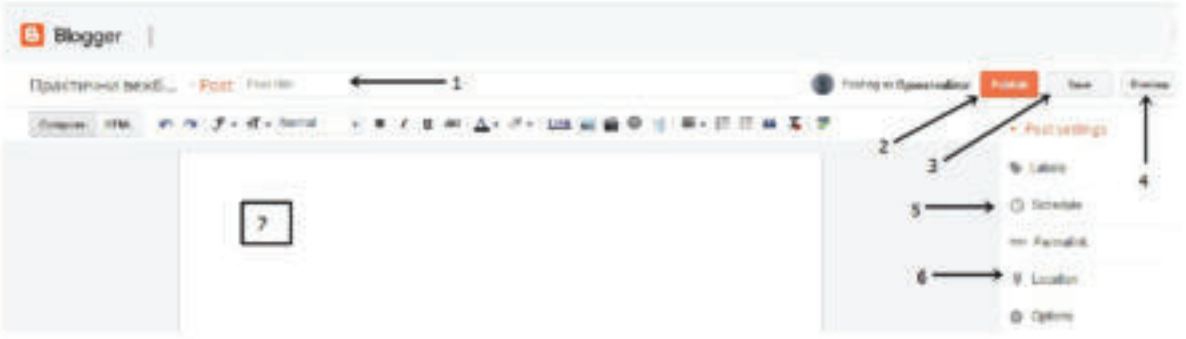
a) bağlantı;
b) blog arşivi;
c) yorumlar.

Ödev 10

Yeni yazılmış bir blog gönderisini yayınlamak için ne gerekiyor?

Ödev 11

Her elemanın anlamını açıklayın!



1._____
2._____
3._____
4._____
5._____
6._____
7._____
8._____

Ödev 12

İnternette belirli blogları yazarken veya ziyaret ederken nasıl davranmalısınız?

_____

_____

_____

Proje çalışması

Ücretsiz blog hizmetlerinden birinde bir web günlük / blog oluşturun. Okuldaki proje faaliyetlerinden birine karşılık gelen veya müfredattaki konulara karşılık gelen bir konu seçin, örneğin: eko-bütünleşme, okulda etnik gruplar arası entegrasyon, spor ve spor faaliyetleri, sanat ve etkinlikler, oyunlar, programlama vb. Blogun içeriğiyle eşleşen blogun temasını / düzenini dikkatlice seçin. Blog farklı içerik türleri içermelidir: metin, bağlantılar, videolar, resimler vb. Blog adreslerini arkadaşlarınızla paylaşın, böylece gönderilere yorum bırakabilir ve ardından aşağıdaki soruları yanıtlayabilirsiniz:

1. Blogunuzun adresi nedir?
2. Blogunuzun başlığı nedir?
3. Blog kimin içindir?
4. Blogdaki son yazı hangi tarihte yazılmıştır?
5. Blogdaki ilk yazı hangi tarihte yazılmıştır?
6. Blogda yazılar dışında başka hangi bilgiler var?

ANAHTAR KELİMELER SÖZLÜĞÜ
(profesyonel terminoloji)

Aktif hücre	Halihazırda işaretlenmiş ve veri girmeye hazır olan hücreye etkin hücre denir.	Konu 1
Analiz	Analiz, problem ödevinin ayrı unsurlara bölünmesidir.	Konu 3
İkili sayılar	İkili sayılar 0 ve 1'dir. Bunlar ikili dilini oluşturur, veriler, bilgiler ve komutlar 0 ve 1 dizilerine çevrilir.	Konu 2
Blog	Bir blog veya web günlük, içeriğin kronolojik sırayla görüntülendiği, yani en son haberlerin veya içeriğin önce görüntülendiği ve geri kalanın aşağıya gittiği çevrimiçi bir günlüktür.	Konu 5
Grafik	Grafik, verilerin grafiksel bir temsilidir.	Konu 1
Grafik ekranı	Grafik görüntüsü	Konu 3
Debager	Debager, bir hata algılayıcı ve hata ayıklayıcıdır.	Konu 4
Dijital baskı	Dijital parmak izi, yanlışlıkla veya kasıtlı olarak geride bırakılan tüm internet bilgileridir (pozitif ve negatif).	Konu 5
Editör	Bir kaynak kodu düzenleyicisi, bir düzenleyicidir.	Konu 4
Eliminasyon	Ortadan kaldırma, önemli olmayan ve sorun durumuna nihai bir çözüme götürmeyen gerçekleri atma sürecidir.	Konu 3
Gönderi - yazı	Blogda yayınlanan yazılardır.	Konu 5
Kaynak kod	İsteğe bağlı bir program, çalıştırılabilir bir bilgisayar programına çevrilmesi gereken metin komutlarının bir listesidir.	Konu 4
Yönetici programı	Çalıştırılabilir bir program, bilgisayarda bir program olarak çalıştırılabilen bir dosyadır.	Konu 4
Tanımlama	Bir değişkene tanımlama sırasında değer atamaya tanımlama denir	Konu 4

Etkileşimli programlar	Etkileşimli programlar, program ile kullanıcı arasında bir tür iletişimin olduğu programlardır.	Konu 3
İfadeler (komutlar)	İfadeler, bilgisayara ne yapacağını söyleyen komutlardır.	Konu 3
Kodlama	Verileri, bilgileri ve programları bilgisayar tarafından anlaşılır bir dile çevirmeye kodlama denir.	Konu 2
Sütun	Sütunlar, bir tablonun dikey parçalarıdır	Konu 1
Yorum	Yorumlar, blog etkileşimi için, yani toplulukla iletişim kurmak için en basit araçtır.	Konu 5
Derleyici	Derleyici, kaynak dosyayı bir çalıştırılabilir dosyaya çeviricisidir.	Konu 4
Sabit	Sabit, sabit ve değişmeyen bir miktardır.	Konu 3
Koordinatlar	Bir noktanın konumunu doğru bir şekilde belirlemek için kullanılan geometrik öğeler.	Konu 3
Kriptografi	Kriptografi, yalnızca amaçlananların anlayabileceği bir şekilde mesaj gönderme ve alma yöntemlerinin araştırılmasıyla ilgilenen bilimsel bir disiplindir.	Konu 2
Mantıksal düşünme	Mantıksal düşünme, iddiaların, yani ifadelerin doğruluğuna saygı duyarak çözüme ulaşmanın bir yoludur.	Konu 3
Olay	Programlama dilinde bir olay, fare aktivitesi veya klavye üzerindeki uygun tuşa basarak programda etkileşimi gerçekleştiren yeni bir yapıdır.	Konu 3
Komşu olmayan hücreler	Bitişik olmayan hücrelerdir	Konu 1
Ardışık hücreler	Üst üste veya yan yana bulunan hücrelerdir.	Konu 1
Programlama	Programlama dillerini kullanarak program yazmaya programlama denir	Konu 2

Değişken	Girilen verilere göre değişen değerlerdir.	Konu 3
Çalışma kitabı	Çalışma kitapları, bir elektronik tablolama programında oluşturulan belgelerdir. Örneğin: Ms Office Excel, Open Office.Org, Apple ve Work Numbers.	Konu 1
Çalışma sayfası	Elektronik tablolama programındaki her çalışma kitabı çalışma sayfalarından oluşur.	Konu 1
Satır	Satırlar tablonun yatay kısımlarıdır	Konu 1
Sıralama	Verileri belirli bir koşula göre sıralamak	Konu 2
Ücretsiz yazılım	Özgür Yazılım, kısıtlama olmaksızın kullanılabilen, üzerinde çalışılabilen ve değiştirilebilen, değiştirilmiş veya değiştirilmemiş formda hiçbir kısıtlama olmaksızın veya minimum kısıtlama ile kopyalanabilen ve yeniden dağıtılabilen yazılımdır.	Konu 4
Karşılaştırmalı ifade	Karşılaştırmalı bir ifade, değerler arasında karşılaştırmaya izin veren bir sıra veya ifadedir	Konu 2
Veri yapıları	Veri yapıları, tek isim altında tanımlanan veri kümeleridir	Konu 1
Tablo	Tablo, verileri sütunlar ve satırlar halinde düzenlemenin bir yoludur.	Konu 1
Hücre	Bir sütun ve bir satırın kesişme noktasına hücre denir	Konu 1
Formüller	Formüller, bir hesaplama yapmak için kullanıcı tarafından oluşturulan aritmetik ve mantıksal ifadelerdir.	Konu 1
Fonksiyonlar	Fonksiyonlar, ad ve hesaplama argümanlarını içeren hazır formüllerdir.	Konu 4
Döngü	Bir koşul karşılanana kadar bir dizi komutu tekrarlamak bir döngüdür.	Konu 4

Davis, S.(2014). Beginning Programming with C++ for Dummies. Wiley Publishing

Дејтел П, Дејтел Х. (2010). C++ Како се програмира-седмо издание. Арс Ламина

Freun S, Starks J, Schmieder E. (2016). Microsoft Office 365 Excel 2016 comprehensive. University Indianapolis

Gardner S. (2005). Buzz Marketing with Blogs for Dummies. Wiley Publishing

Marji M. (2014). Learn to program with Scratch. William Pollock

www.bebras.org

https://education.microsoft.com

www.bbc.co.uk

www.code.org