

NESNE TABANLI PROGRAMLAMA

Program, herhangi bir elektronik cihaza bir işlem yaptırmak için yazılan komutlar dizisidir. Programlar; bilgisayar, cep telefonu, tablet, elektronik ev eşyaları, araba ve daha birçok yerde kullanılır.

Programlama; bilgisayar programlarının yazılması, test edilmesi ve bakımının yapılması sürecine verilen isimdir. Bu sürecin daha verimli geçmesi için amaca yönelik bir programlama dili seçilmelidir. Nesne Tabanlı Programlama ders kitabının içeriğinde C# programlama dilinin kullanımı anlatılacaktır

C# Programlama Dili

C# programlama dili, nesne tabanlı olarak geliştirilmiş bir dildir. C# programlama dili ile;

- Mobil uygulamalar,
- Konsol uygulamaları,
- Web servisleri,
- Dinamik kütüphaneler (DLL),
- Oyun tasarımı,
- Form uygulamaları yapılabilir.

Günlük hayatın birçok alanında büyük küçük pek çok şirket C# ile geliştirilmiş programlar kullanmaktadır. C# programlama dilinin çok tercih edilmesinin sebeplerinden bazıları şunlardır:

- Yazılması ve anlaşılması kolay kod yapısına sahiptir.
- Yeni teknolojileri destekler.
- Kullanışlıdır.
- Ekip çalışmasına elverişlidir.
- Kullanıcıyla etkileşimlidir.
- Grafik arayüzlü tasarımlar yapılabilir.
- Ağ üzerinden birbiriyle uyumlu çalışabilir.
- Çevrimiçi veya çevrimdışı kullanılabilir.
- QR kod okuyucu, kamera, yazıcı vb. cihazlarla etkileşimlidir.
- Verileri depolayıp işleyerek analiz yapabilir.
- Sosyal medya platformları ile etkileşimlidir.
- Cep telefonu uygulamaları ile etkileşimlidir.
- Yapay zekâ teknolojisi kullanılarak yüz tanıma, nesne tanıma, ses tanıma işlemleri yapılabilir.
- Birçok özelliğe sahip farklı programlar geliştirebilme imkânı sağlar.



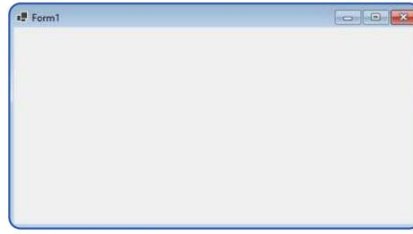
Uygulama-1

Adım 1: Kod editörü arayüzünü açarak başlangıç ekranından Create a new project'i seçiniz ve yeni bir proje oluşturunuz.

Sıra Sizde

1. "MerhabaDunya" isimli bir Form uygulaması oluşturunuz.
2. Oluşturduğunuz "MerhabaDunya" isimli projeyi kaydedip kapattıktan sonra var olan projeyi açma seçeneği ile yeniden açınız.

Form Ekranı



Görsel 1.9: Form ekranı

Form ekranı, programın görsel tasarımının yapıldığı yerdir Araç Kutusu (Toolbox) panelinde bulunan nesnelere form üzerinde istenilen pozisyona yerleştirilerek programın tasarımı yapılır.

Araç Kutusu (Toolbox)

Form üzerinde

tasarım için

kullanılabilecek

nesnelere bu

panelde bulunur.

Görsel 1.10'da

görüldüğü gibi tüm

nesnelere aynı anda

veya işlevlerine

göre çeşitli

kategorilerde

listelenebilir.

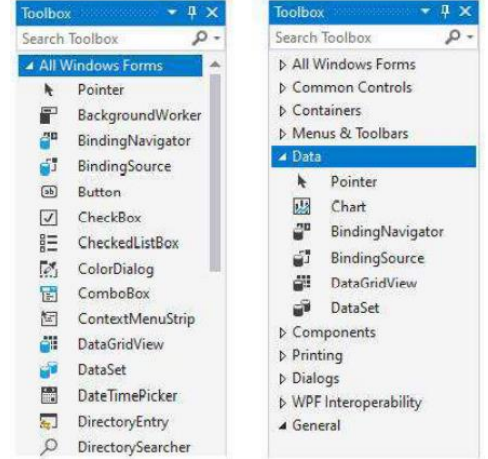
Örneğin, Data

bölümüne

tıklandığında Data

(Veri) ile ilgili nesnelere listelenir. Ayrıca Search Toolbox (Arama

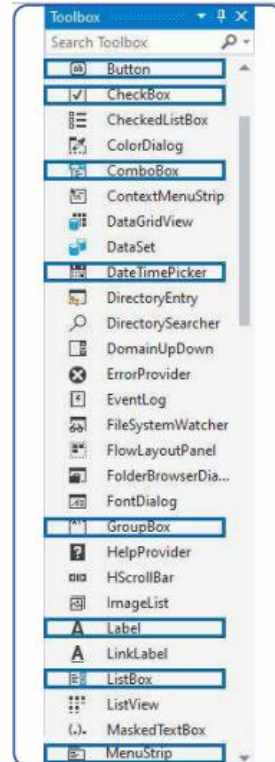
Çubuğu) ile istenilen nesnenin ismi yazılarak da nesne listelenebilir



Sıra Sizde

1. Toolbox panelinde Label ve TextBox nesnelere çift tıklayarak bu nesnelere formun üzerine yerleştiriniz.
2. Label ve TextBox nesnelere formun ortasına yerleştiriniz.

En Çok Tercih Edilen Toolbox Nesnelere



CheckBox: Kullanıcıya bir veya aynı anda birden çok seçeneği işaretleme imkânı sağlayan nesnedir.

ComboBox: Açılır liste ile açılan seçenekler arasından seçim yapılmasına olanak sağlayan araçtır. Listeye yeni eleman ekleme ve çıkarma işlemleri, tasarım ekranından veya program çalışırken kod ile yapılabilir.

DateTimePicker: Tarih ve saat seçme işlemlerine olanak sağlayan nesnedir.

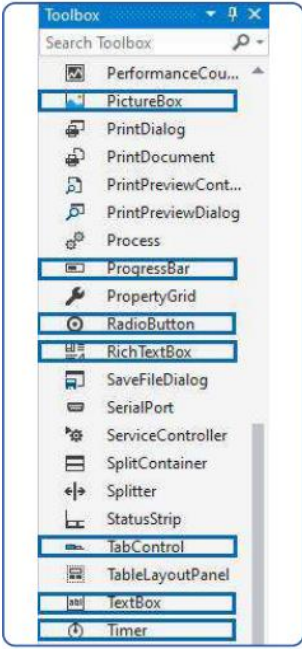
GroupBox: Form elemanlarını kendi aralarında gruplamak için kullanılan nesnedir.

Nesnelere gruplar hâlinde olduğu için daha anlaşılır tasarımlar yapılabilir.

Label: Form üzerinde bilgi vermek için kullanılan nesnedir.

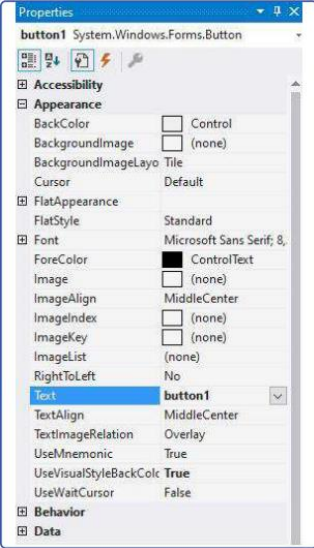
ListBox: Sunulacak seçeneklerin açık bir liste hâlinde gösterildiği nesnedir.

MenuStrip: Programda menü başlıkları ve alt başlıkları oluşturmak için kullanılan nesnedir



PictureBox: Form üzerinde resim göstermek için kullanılan nesnedir.
ProgressBar: Yapılan bir işlemin ne kadarının tamamlandığını göstermek için kullanılan nesnedir.
RadioButton: CheckBox nesnesinden farklı olarak birçok seçenek içinden sadece birinin seçilmesine imkân sağlayan nesnedir.
RichTextBox: Birden çok satır içine metin girişi yapılabilen nesnedir.
TabControl: Form elemanlarının gruplara ayrıldığı, grupların içindeki elemanları görmek için sekmelerin kullanıldığı nesnedir. Her sekme, bir grubu temsil eder.
TextBox: İçine tek satır metin girişi yapılabilen nesnedir. Bilgi girişi için en çok tercih edilen nesnedir.
Timer: Kodların zamanlanarak çalışmasını sağlayan nesnedir.

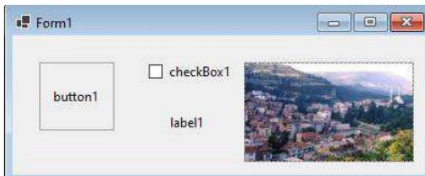
Özellikler (Properties)



Form nesnesinin ve diğer tüm nesnelerin özelliklerinin listelendiği, değiştirildiği ve ayrıca nesnelere ait olayların (events) listelendiği paneldir.
BackColor: Nesnenin arka plan rengini değiştirir.
BackgroundImage: Nesnenin arka planına resimler ekler. **Font:** Nesnenin yazı tipini, boyutunu ve kalınlığını değiştirir.
ForeColor: Nesnenin yazı rengini değiştirir.
Text: Nesnenin yazı metnini değiştirir.
TextAlign: Nesnenin yazısını hizalar.

Uygulama-2

Adım 1: Görsel 1.14'te görüldüğü gibi Form üzerine Button, CheckBox, Label ve PictureBox nesnelerini ekleyiniz. Toolbox panelinden sürükleyip bırak yöntemi ile nesnelere form üzerine yerleştiriniz.



Adım 2: Bu nesnelerin ortak ve farklı özelliklerini belirleyip defterinize not alınız. Form üzerindeki nesnelerin hepsi fare ile aynı anda seçildiğinde properties panelinde nesnelerin sadece ortak özellikleri listelenmektedir.

Olaylar (Events)

Her nesnenin form üzerinde bir görevi bulunmaktadır. Bazı nesnelere sadece programın işlevi ile ilgili bilgiyi ve görseli yansıtmak için kullanılır. Bazı nesnelere ise belli durumlarda (üzerine tıkladığında, bir tuşa basıldığında vs.) kod parçacıklarını çalıştırmak için kullanılır. Nesnelere, kullanıcı ile etkileşimi olaylar sayesinde sağlamaktadır. Günlük hayatta sosyal medya uygulamalarında, web sitelerinde, oyunlarda ve daha birçok alanda nesnelere tanımlanmış olay metodları çalışarak kullanıcıyla etkileşim sağlanır. Olay metodları tanımlanarak nesnelerin hangi durumda, nasıl kodlar çalıştırabileceği belirlenir. Örneğin; butona tıkladığında şifre kontrolünün yapılması, klavyeden sağ ok tuşuna basıldığında bir sonraki resmin gösterilmesi, PictureBox nesnesinin üzerine çift tıklama yapıldığında resme ait ilgili bilgilerin MessageBox ile göstermesi vb. Olaylar için metod tanımlanırken istenilen olayın adının hemen yanındaki boş beyaz kutucuğa çift tıklar ve otomatik olarak aşağıdaki gibi bir metod oluşturulur

```
private void button1_Click(object sender, EventArgs e)
{
    listBox1.Items.Add(textBox1.Text);
}
```

Görsel 1.15'te seçili button1 nesnesine ait bazı olaylar aşağıda verilmiştir.

Click: Nesneye fare ile tıklanması veya fare nesne üzerinden enter tuşuna basılmasıyla devreye giren olaydır.

MouseDown: Farenin tuşuna basılması olayıdır.

KeyDown: Klavyeden bir tuşa basılması olayıdır.

KeyUp: Klavyede basılan tuşun bırakılması olayıdır.

MouseClick: Click gibi nesnenin fare ile tıklanması olayıdır fakat MouseClick enter tuşundan etkilenmez. Ayrıca MouseClick, fareye ait koordinatlar gibi özel bilgileri de verir.

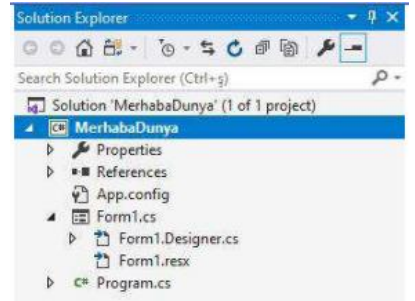
Uygulama-3



Adım 1: Form üzerine TextBox ve Button nesnelerini ekleyiniz
Adım 2: Düğmeye basıldığında TextBox da kişinin adı soyadı yazılacak.

Çözüm Penceresi (Solution Explorer)

Proje ile ilgili tüm dosya ve klasörleri listelenerek silme, kopyalama, taşıma, isim değiştirme işlemlerinin yapılabildiği paneldir. Projenin detaylı bir haritası gibi düşünülebilir. Çözüm penceresi ile projeye yeni sınıf (class), form ve başka öğeler eklenebilir. Veri tabanı dosyası, resim, müzik, video dosyaları projeye dâhil edilebilir. Projeye eklenen her form, çözüm penceresinde ayrı ayrı listelenmektedir. Listedeki Form1.cs ifadesine fare sağ tuşu ile tıkladığında açılan listeden View Code tıklanır ise Form1'e ait kod ekranı, View Designer tıklanır ise Form1'e ait tasarım ekranı açılır. Doğrudan Form1.cs ifadesine çift tıklanır ise tasarım ekranı açılır.



Kod veya tasarım ekranını göstermenin diğer yolları şunlardır:

1. Form ekranı üzerinde herhangi bir noktada fare sağ tuşu kullanılarak açılan listede View Code (Kodu Göster) tıklanarak kod ekranı, View Designer (Tasarımı Göster) tıklanarak tasarım ekranı görüntülenir.
2. Klavyeden F7 tuşuna basıldığında kod ekranı açılır. Klavyeden Shift+F7 tuşuna basıldığında tasarım ekranı açılır.

Uygulama-4

Adım 1: Form üzerine Görsel 1.20'de görüldüğü gibi bir tane Button nesnesi ekleyiniz.

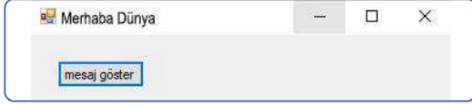
Adım 2: Özellikler penceresinden formun Text özelliğini "Merhaba Dünya" yapınız.

Adım 3: Özellikler listesinden button1'in Text özelliğini mesaj göster yapınız.

Adım 4: Olaylar listesinden button1 için Click olayı metodunu button1'in üzerine çift tıklayarak oluşturunuz.

Adım 5: Oluşturduğunuz metodun içine MessageBox.Show ("Merhaba Dünya"); kodunu yazınız.

Adım 6: Araç Çubuğunda bulunan ▶ Başlat - butonu ile programı çalıştırınız.



Uygulama-6

Adım 1: Form üzerine Görsel 1.22'de görüldüğü gibi iki tane Group-Box nesnesi ekleyiniz.

Adım 2: Üst tarafta bulunan GroupBox nesnesinin içine bir tane Label nesnesi ekleyiniz.

Adım 3: Alt tarafta bulunan GroupBox nesnesinin içine on tane Button nesnesi ekleyiniz ve bunları numaralandırınız.

Adım 4: Numaraların yazılı olduğu Button nesnelerinin arka plan rengini "MenuHighlight" yapınız.

Adım 5: Button nesnelerinin ve Label nesnesinin yazı tipi stilini "kalın", yazı boyutunu "10" yapınız.

Adım 6: Form nesnesinin başlığını "0-9" ve Form nesnesinin arka plan rengini "SandyBrown" yapınız.

Adım 7: Tıklanan Button nesnesine ait sayıyı Label nesnesinin Text özelliğine aktaran programı yazınız.



Görsel 1.22: Tuş takımı

```
private void button1_Click(object sender, EventArgs e)
{ label1.Text = "1"; }
private void button2_Click(object sender, EventArgs e)
{ label1.Text = "2"; }
private void button3_Click(object sender, EventArgs e)
{ label1.Text = "3"; }
private void button4_Click(object sender, EventArgs e)
{ label1.Text = "4"; }
private void button5_Click(object sender, EventArgs e)
{ label1.Text = "5"; }
private void button6_Click(object sender, EventArgs e)
{ label1.Text = "6"; }
private void button7_Click(object sender, EventArgs e)
{ label1.Text = "7"; }
private void button8_Click(object sender, EventArgs e)
{ label1.Text = "8"; }
private void button9_Click(object sender, EventArgs e)
{ label1.Text = "9"; }
private void button10_Click(object sender, EventArgs e)
{ label1.Text = "0"; }
```

Sıra Sizde

- Form üzerine 2 adet Button nesnesi ve 1 adet PictureBox nesnesi ekleyiniz.
- Button1'e tıkladığında PictureBox nesnesini görünmez hâle getiren, button2'ye tıkladığında PictureBox nesnesini görünür hâle getiren programı yazınız.

Uygulama-7

Sepete Ekle butonuna tıkladığında TextBox nesnesindeki değeri ListBox nesnesine aktaran Görsel 1.23'teki gibi bir tasarıma sahip programı yazınız.

```
private void button1_Click(object sender, EventArgs e)
{
    listBox1.Items.Add(textBox1.Text);
}
```



Görsel 1.23: Manav sepeti uygulaması

Sıra Sizde

- Görsel 1.23'teki program tasarımına **Sepeti Temizle** butonu ekleyiniz.
- Yeni eklediğiniz butona tıkladığında ListBox nesnesinin içindekileri tamamen temizleyen programı yazınız.

Sıra Sizde

- Form üzerine 2 adet Button nesnesi ekleyiniz.
- Button1'e tıkladığında "Button1'e tıkladınız.", button2'ye tıkladığında "Button2'ye tıkladınız." mesajlarını verdiriniz.
- Oluşturduğunuz kodları aşağıdaki kutucuğa yazınız.

Uygulama-8

Not: Bir nesne için sadece bir tane olay metodu oluşturulmaz. Nesne için tanımlanabilecek ne kadar olay varsa o kadar da olay metodu oluşturulabilir.

Mouse simgesi ile forma eklenen bir Button nesnesinin üzerine gelindiğinde "Mouse şimdi üzerindedir." mesajını, mouse simgesi Button nesnesinin üzerinden kaldırıldığında "Mouse artık üzerimde değildir." mesajını verdiriniz.

```
private void button1_MouseMove(object sender, EventArgs e)
{
    MessageBox.Show("Mouse Üzerimdedir.");
}
private void button1_MouseLeave(object sender, EventArgs e)
{
    MessageBox.Show("Mouse Artık Üzerimde Değildir.");
}
```

Sıra Sizde

- Form üzerine 3 adet Button nesnesi ekleyiniz.
- Form nesnesinin arka plan rengini turuncu yapınız.
- Button'ların hepsinin arka plan ve yazı renklerini birbiriyle uyumlu olacak şekilde tasarlayınız.
- Mouse simgesi ile button1'in üzerine gelindiğinde "Mouse benim üzerindedir.",
 - mouse simgesi ile button2'nin üzerine çift tıkladığında "Mouse iki kere tıklandı.",
 - button3'ün üzerinden mouse simgesi kaldırıldığında "Mouse üzerinde değil." mesajlarını verdiğiniz ve oluşturduğunuz kodları aşağıdaki kutucuğa yazınız.

Uygulama-9

Adım 1: Form üzerine Görsel 1.24'te görüldüğü gibi GroupBox nesnesi içine 4 adet Button nesnesi ekleyiniz.

Adım 2: Form nesnesinin arka plan rengini beyaz yapınız.

Adım 3: Kırmızı yazılı Button nesnesine tıkladığında Form nesnesinin arka plan rengini kırmızı, yeşil yazılı Button nesnesine tıkladığında Form nesnesinin arka plan rengini yeşil, mavi yazılı Button nesnesine tıkladığında Form nesnesinin arka plan rengini mavi, gri yazılı Button nesnesine tıkladığında Form nesnesinin arka plan rengini gri yapan programı yazınız.



Görsel 1.24: Form boyama

```
private void button1_Click(object sender, EventArgs e)
{ this.BackColor = Color.Red; }
private void button2_Click(object sender, EventArgs e)
{ this.BackColor = Color.Green; }
private void button3_Click(object sender, EventArgs e)
{ this.BackColor = Color.Blue; }
private void button4_Click(object sender, EventArgs e)
{ this.BackColor = Color.Gray; }
```

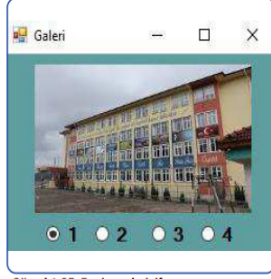
Uygulama-10

Adım 1: Form üzerine Görsel 1.25'te görüldüğü gibi dört tane PictureBox nesnesini üst üste olacak şekilde yerleştiriniz.

Adım 2: PictureBox nesnelerinin "Visible" özelliğini "false" yapınız (Visible özelliği ile nesnenin görünürlüğü belirlenir).

Adım 3: Dört tane RadioButton nesnesi ekleyiniz ve bunları numaralandırınız.

Adım 4: radioButton1 nesnesi işaretli ise sadece pictureBox1 nesnesini görünür hâle getirecek, radioButton2 nesnesi işaretli ise sadece pictureBox2 nesnesini görünür hâle getirecek, radioButton3 nesnesi işaretli ise sadece pictureBox3 nesnesini görünür hâle getirecek, radioButton4 nesnesi işaretli ise sadece pictureBox4 nesnesini görünür hâle getirecektir.



Görsel 1.25: Resim galerisi

```
private void radioButton1_CheckedChanged(object sender, EventArgs e)
```

```
{ pictureBox1.Visible = radioButton1.Checked; }
```

```
private void radioButton2_CheckedChanged(object sender, EventArgs e)
```

```
{ pictureBox2.Visible = radioButton2.Checked; }
```

```
private void radioButton3_CheckedChanged(object sender, EventArgs e)
```

```
{ pictureBox3.Visible = radioButton3.Checked; }
```

```
private void radioButton4_CheckedChanged(object sender, EventArgs e)
```

```
{ pictureBox4.Visible = radioButton4.Checked; }
```

Not: RadioButton nesnesi işaretlenince veya RadioButton nesnesinin işareti kaldırılınca CheckedChanged olayı devreye girer.

Değişkenler ve Temel Veri Türleri

Değişkenler, programlamanın en temel kavramıdır. Program çalışması sırasında çeşitli türlerde verileri hafızada saklayan değişkenler, ihtiyaca göre tekrar tekrar kullanılan veri tutuculardır.

Temel Veri Türleri

Tablo 1.1: Tam Sayı Veri Türleri

Tam Sayı Türü	Kapladığı Alan	Değer Aralığı
byte	1 Bayt	0,...,255
sbyte	1 Bayt	-128,...,127
short	2 Bayt	-32768,...,32767
ushort	2 Bayt	0,...,65535
int	4 Bayt	-2147483648,...,2147483647
uint	4 Bayt	0,...,4294967295
long	8 Bayt	-9223372036854775808
ulong	8 Bayt	0,...,18446744073709551615

Tablo 1.2: Ondalık Sayı Veri Türleri

Ondalık Sayı Türü	Kapladığı Alan	Değer Aralığı
float	4 Bayt	$\pm 1.5 \times 10^{-45}$,..., $\pm 3.4 \times 10^{38}$
double	8 Bayt	$\pm 5.0 \times 10^{-324}$,..., $\pm 1.7 \times 10^{308}$
decimal	16 Bayt	$\pm 1.5 \times 10^{-28}$,..., $\pm 7.9 \times 10^{28}$

Tablo 1.3: Metinsel Veri Türleri

Metinsel Veri Türü	Kapladığı Alan	Değer Aralığı
string	Sınırsız	Metinsel ifade tutar.
char	2 Bayt	Tek bir karakter tutar.

Tablo 1.4: Mantıksal Veri Türleri

Mantıksal Veri Türü	Kapladığı Alan	Değer Aralığı
bool	1 bit	True-False veya 1-0

1.6.2. Değişken Tanımlama

(Değişkenin Veri Türü) (Değişken Adı)

```
int sayi
```

```
double pi
```

```
string soru
```

```
char karakter
```

```
bool cevap
```

Not: Aynı veri türündeki değişkenler, aynı kod satırında ve aralarına virgül konularak tanımlanabilir.

Örnek:

```
int sayi,deger,sonuc;
```

```
string isim1.isim2.sovad;
```

1.6.4. Değişken İsmi Kuraları

1. Değişken isimlerinde boşluk kullanılmaz. Boşluk yerine alt çizgi (_) kullanılabilir.

Örnek: string ad soyad; şeklinde kullanım yanlıştır.

```
string ad_soyad; veya string adsoyad; şeklinde kullanım doğrudur.
```

2. ?, !, ;, %, +, -, . gibi özel karakterler kullanılmaz.

Örnek: string soru?; şeklinde kullanım yanlıştır.

```
string soru; şeklinde kullanım doğrudur.
```

3. Değişken isimleri sayı ile başlamaz.

Örnek: byte 1not; şeklinde kullanım yanlıştır.

```
byte not1; şeklinde kullanım doğrudur.
```

4. Değişken isimleri büyük ve küçük harfe duyarlıdır.

Örnek: ulong toplamTutar; şeklinde tanımlanan değişken ile ulong ToplamTUTAR şeklinde tanımlanan değişken aynı değildir.

5. Herhangi bir kodla aynı isimde değişken tanımlanamaz. Değişkenlerde if, else, random gibi programa ait ifadeler isim olarak kullanılmaz.

6. Zorunluluk yoktur fakat Türkçe karakterlerin (ç, ö, ü, ğ, ş vb.) kullanılması tavsiye edilmez.

1.6.5. Değişken Veri Türü Dönüştürme (Convert) İşlemleri

Değişkenlerin veri türlerini bazen değiştirmek gerekebilir. Sayısal bir ifade, bir nesnenin Text özelliğine aktarılmak istendiğinde program hata verecektir. İçeriği tamamen sayı olsa da metinsel bir ifadeyi sayısal veri türüne sahip bir değişkene aktarırken program yine hata verecektir. Bu tip durumlarda değişkenlerin veri türlerini dönüştürmek gerekir. Tür dönüşümünün sağlayacak hazır metodlar şunlardır:

ToString() >>>> Her türden değişkeni string türüne dönüştürür. ToString(), en sık kullanılan dönüştürme metodudur.

Convert.ToByte(metin)	>>>>>	Byte'a çevirir.
Convert.ToInt16(metin)	>>>>>	Short'a çevirir.
Convert.ToInt32(metin)	>>>>>	Int'e çevirir.
Convert.ToInt64(metin)	>>>>>	Long'a çevirir.
Convert.ToSingle(metin)	>>>>>	Float'a çevirir.
Convert.ToDouble(metin)	>>>>>	Double'a çevirir.
Convert.ToDecimal(metin)	>>>>>	Decimal'a çevirir.
Convert.ToChar(metin)	>>>>>	Char'a çevirir.
Convert.ToBoolean(metin)	>>>>>	Bool'a çevirir.

Örnek:

```
int sayi1=100;
```

```
string deger;
```

```
deger=sayi1; //Program bu noktada convert type 'int' to 'string' şeklinde hata verecektir.
```

Örnek:

```
int sayi1=100;
```

```
string deger;
```

```
deger=sayi1.ToString(); //Program artık hata vermeyecektir çünkü veri türleri uyumludur.
```

Uygulama-11

Adım 1: Form üzerine Görsel 1.28'de görüldüğü gibi bir CheckBox nesnesi ekleyiniz.

Adım 2: İki adet Label nesnesi ekleyiniz.

Adım 3: CheckBox nesnesi işaretli ise label2 nesnesine "True" değerini, CheckBox nesnesi işaretli değil ise label2 nesnesine "False" değerini gösteren programı yazınız.



Görsel 1.28: Lamba kontrol

```
private void checkBox1_CheckedChanged(object sender, EventArgs e)
```

```
{
```

```
bool secim;
```

```
secim = checkBox1.Checked; //Checked özelliği True veya False değerleri alır.
```

```
label2.Text = secim.ToString();
```

```
}
```

Sıra Sizde

Bir değişken tanımlayıp içine 500 sayı değerini atayınız. Butona tıkladığında bu değişkendeki değeri ekrana mesaj olarak verdiriniz.

1.7. Aritmetiksel Operatörler

Aritmetiksel operatörler, matematiksel işlemlerde kullanılan özel karakterlerdir (Tablo 1.5).

Tablo 1.5: Aritmetiksel Operatörler

Operatör Adı	Sembölü	Örnek
Toplama	+	2+8
Çıkarma	-	8-2
Çarpma	*	8*2
Bölme	/	8/2
Mod alma	%	8%2

Not: Bir sayı başka bir sayıya bölündüğünde kalan sayıya **mod** denir.

Örnek: $8\%2=0$ iken $8\%5=3$ sonucunu verir.



Uygulama-12

Girilen iki sayıyı toplayan programı Görsel 1.29'daki gibi tasarlayıp yazınız.



Görsel 1.29: Toplama uygulaması

```
private void button1_Click(object sender, EventArgs e)
{
    int sayi1, sayi2, toplam;
    sayi1 = Convert.ToInt16(textBox1.Text);
    sayi2 = Convert.ToInt16(textBox2.Text);
    toplam = sayi1 + sayi2;
    textBox3.Text = toplam.ToString();
}
```

Not: TextBox nesnesinin içine sayısal bir değer girilmiş olsa bile matematiksel işlemler yapılamaz çünkü TextBox nesnesi metinsel veri türüne sahip değerler alabilen bir nesnedir. Bu tip durumlarda Convert kodu ile veri türü dönüşümü yapılmalıdır. Ayrıca TextBox nesnesine sayısal veri türünde bir değer yazmak için ToString() metodu ile veri türü dönüşümü yapılmalıdır.



Uygulama-13

Aşağıdaki örnek kodları çalıştırınız.

1.Örnek Kodlar	2.Örnek Kodlar
<pre>string ad, soyad, topla; ad = "Zeynep"; soyad = "Sare"; topla = ad + " " + soyad; MessageBox.Show(topla);</pre>	<pre>textBox1.Text="25"; textBox2.Text="2" textBox3.Text= textBox1.Text + textBox2.Text</pre>

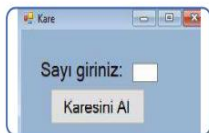
Not: Programlama dillerinde toplama operatörü sadece sayıları toplamak için kullanılmaz. Metinsel veri türüne sahip değişkenlerin arasında **artı (+)** operatörü konularak string değerlerini birleştirmek için de toplama operatörü kullanılır. Yukarıdaki 1. örnek kodlar çalıştığında ekrana **"Zeynep Sare"** mesajı gelecektir. 2. örnek kodlar çalıştığında ise textBox3'ün değeri **"252"** olur.yazmak için ToString() metodu ile veri türü dönüşümü yapılmalıdır.

Sıra Sizde

Girilen iki sayının toplama, çıkarma, çarpma ve bölme işlemlerini yapan programı tasarlayıp yazınız.

Sıra Sizde

Buton nesnesine çift tıklandığında TextBox nesnesine girilen sayının karesini alıp mesaj verdiren programı Görsel 1.30'da görüldüğü gibi tasarlayıp yazınız.



Görsel 1.30: Kare alma uygulaması

1.7.1. İşlem Önceliği

Matematiksel bir ifade yazılan kod satırında birden fazla aritmetiksel operatör kullanılabilir. Böyle durumlarda sonuçların doğru çıkması için operatörler arasındaki işlem önceliklerine dikkat edilerek kod yazılmalıdır.

Kodlar çalıştırılırken öncelikle varsa parantez içindeki işlemler, daha sonra parantez dışındaki işlemler yapılır. Çarpma, bölme ve mod alma işlemleri kendi aralarında, toplama ve çıkarma işlemleri de kendi aralarında aynı önceliğe sahiptir. Kendi aralarında aynı önceliğe sahip operatörlerin olduğu ifadelerde ise sırasıyla soldan sağa doğru işlem yapılır.

Örnek: $4*3+15/3-1$ işlemini yapınız.

Çözüm:
 $=12+5-1 \rightarrow$ Önce çarpma sonra bölme işlemi yapılır.
 $=17-1 \rightarrow$ Toplama işlemi ve sonrasında çıkarma işlemi yapılır.
 $=16$

Örnek: $(3+4-2)*(10/5*2)-4$ işlemini yapınız.

Çözüm:
 $=5*4-4 \rightarrow$ Önce parantez içindeki işlemler yapılır.
 $=20-4 \rightarrow$ Çarpma işlemi toplama işleminden öncelikli olarak yapılır.
 $=16$

Not: Kod yazımı sırasında karmaşayı ortadan kaldırmak ve en doğru sonuca ulaşmak için öncelikli olan ifadeleri parantez içine alabilirsiniz.

Sıra Sizde

Aşağıdaki işlemlerin sonucunu yazınız.

$7*5-3$	$(5*4)-(3+9)$
$20+30-4*10$	$(5*8/4)+3-5*2$
$9/3+2*2-5+1$	



Uygulama-14

İki sayıyı toplayıp mesaj verdiren programı yazınız (Değişkenlere istenilen değerler atanabilir.).

```
private void button1_Click(object sender, EventArgs e)
{
    int sayi1;
    int sayi2;
    int toplam;
    sayi1 = 5;
    sayi2 = 20;
    toplam = sayi1 + sayi2;
    MessageBox.Show(toplam.ToString());
}
```

Sıra Sizde

İki sayıyı toplayıp 2 ile çarparak çıkan sonuca 5 ekleyen ve oluşan değeri mesaj verdiren programı işlem önceliklerine dikkat ederek yazınız (Değişkenlere istenilen değerler atanabilir.).



Uygulama-15

Adım 1: Form üzerine Görsel 1.31'de görüldüğü gibi etiket fiyatı girişi için TextBox nesnesi ekleyiniz.

Adım 2: Dört adet Button nesnesi ekleyiniz.

Adım 3: Button nesnelerinin arka plan renklerini değiştiriniz.

Adım 4: İndirim oranlarını Button nesnelerinin üzerine yazınız.

Adım 5: TextBox nesnesine etiket fiyatı girilen ürünün seçilen indirim oranına göre indirimli fiyatını bulup Label nesnesinde gösteren programı yazınız.



```
private void button1_Click(object sender, EventArgs e)
{
```

```
    int etiketFiyati;
    double indirimliFiyat;
    etiketFiyati = Convert.ToInt32(textBox1.Text);
    indirimliFiyat = etiketFiyati-etiketFiyati * 0.10; //Yüzde 10 indirim
    label3.Text = indirimliFiyat.ToString();
}
```

```

private void button2_Click(object sender, EventArgs e)
{
    int etiketFiyati;
    double indirimliFiyat;
    etiketFiyati = Convert.ToInt32(textBox1.Text);
    indirimliFiyat = etiketFiyati-etiketFiyati * 0.25; //Yüzde 25 indirim
    label3.Text = indirimliFiyat.ToString();
}
private void button3_Click(object sender, EventArgs e)
{
    int etiketFiyati;
    double indirimliFiyat;
    etiketFiyati = Convert.ToInt32(textBox1.Text);
    indirimliFiyat = etiketFiyati-etiketFiyati * 0.50; //Yüzde 50 indirim
    label3.Text = indirimliFiyat.ToString();
}
private void button4_Click(object sender, EventArgs e)
{
    int etiketFiyati;
    double indirimliFiyat;
    etiketFiyati = Convert.ToInt32(textBox1.Text);
    indirimliFiyat = etiketFiyati-etiketFiyati * 0.75; //Yüzde 75 indirim
    label3.Text = indirimliFiyat.ToString();
}

```

Not: "etiketFiyati - etiketFiyati * 0.10" ifadesinde işlem önceliği çarpmada olduğu için önce çarpma işlemi sonra çıkarma işlemi yapılacaktır. Aynı ifade "etiketFiyati - (etiketFiyati * 0.10)" şeklinde yazılarak da işlem yapılabilir

Uygulama-16



- Adım 1:** Form üzerine Görsel 1.32'de görüldüğü gibi 1 adet GroupBox, 1 adet But-ton, 5 adet Label, 6 adet TextBox nesnesi ekleyiniz.
- Adım 2:** Türkçe ve Matematik netlerinin gösterileceği TextBox nesnelerinin Enabled özelli-ğini bilgi girişini engellemek için false yapınız.
- Adım 3:** Button nesnesi tıklandığında ilgili derslere ait net sayısını hesaplayan programı yazınız (Net sayısı, doğru sayısından yanlış sayısının dörtte biri çıkarılarak hesaplanır).

```

private void button1_Click(object sender, EventArgs e)
{
    double turkceDogru, matDogru;
    double turkceYanlis, matYanlis;
    double turkceNet, matNet;
    turkceDogru = Convert.ToDouble(textBox1.Text);
    turkceYanlis = Convert.ToDouble(textBox2.Text);
    turkceNet = (turkceDogru - (turkceYanlis / 4));
    textBox3.Text = turkceNet.ToString();
    matDogru = Convert.ToDouble(textBox4.Text);
    matYanlis = Convert.ToDouble(textBox5.Text);
    matNet = (matDogru - (matYanlis / 4));
    textBox6.Text = matNet.ToString();
}

```

ÖLÇME VE DEĞERLENDİRME

A) Aşağıdaki cümlelerde parantez içine yargılar doğru ise (D), yanlış ise (Y) yazınız.

- () Kod editörü platformu kullanılarak sadece C# programlama dili kodlanabilir.
- () string veri türüne sahip bir değişken int veri türüne dönüştürülebilir.
- () int a="52"; hatalı bir koddur.
- () string a="16"; hatalı bir koddur.
- () Değişken isimleri sayı ile başlamaz.
- () Bir nesne için aynı anda birden fazla olay metodu oluşturulabilir.
- () Button nesnesi için sadece Click olay metodu oluşturulabilir.

B) Aşağıda verilen cümlelerdeki boşlukları kutularda verilen ifadelerle tamamlayınız.

form	Tostring()	%	ToolBox	using
------	------------	---	---------	-------

- İsim uzayını projeye dâhil etmek için kodu kullanılır.
- Form üzerine eklenen nesnelere panelinden seçilir.
- Mod alma işlemi için aritmetiksel operatörü kullanılır.
- Sayısal veri türüne sahip bir değişken metodu ile string veri türüne dönüştürülür.

12. Aşağıdaki kod satırlarından hangisi çalışmaz? Kodun çalışmama sebebini yazınız.

```

int öğrenci no;
int 1.not;
string isim;
double yarıcap;
bool cinsiyet;
int sayı1=13.4;
char karakter = "kmr";
string tc = "00334412376";

```

14-Aşağıdaki kodlar çalıştırıldığında ekranda görülecek sayı kaçtır?

```

double a, b, c, d, islem;
a = 1;    b = 6;    c = 5;    d = 2;
islem = a * b / c + d;
MessageBox.Show(islem.ToString());

```

15. TextBox nesnesine girilen sayının %18'ini bulup mesaj verdiren programı yazınız

16. Yarıçapı girilen dairenin alanını ve çevresini hesaplayan programı yazınız.

17. Aşağıdaki form ekranını tasarlayıp çalışması için gerekli kodları yazın.

18-C# Windows Form ile Yazılı Ortalaması Hesaplama programını yaz