



GÖRSEL PROGRAMLAMA

Yrd. Doç. Dr. Durmuş ÖZDEMİR

durmus.ozdemir@dpu.edu.tr

Hafta 3

Bu Hafta

Veri Tiplerine Genel Bir Bakış

Veri Tiplerini Dönüştürme

Değer Atama

Atama Operatörleri

Kıyaslama Operatörleri

Mantıksal Operatörler

Aritmetik Operatörler

Döngüler

Veri Tiplerine Genel Bir Bakış

Veri Tiplerini doğru tanımlarsanız;

→ Oluşturacağınız programın kullandığı **disk alanını ekonomik bir şekilde kullanmasını** sağlarsınız.

→ Ayrıca dolaylı ve doğrudan **programın performansını** yükseltebilirsiniz.

→ Değişken Tanımlama:
veri_tipi değişken_adı; (Örnek: string isim;)

→ Değer Atama:
doğrudan → string isim="Osman";
başka bir satırda → isim="Osman";

Tam Sayı Veri Tipleri

Veri Tipinin Adı	Kapladığı Alan	Değer Aralığı	Kullanım Amacı Örnek:	Örnek Uygulama:
byte	1 Byte	0 → 255	Genellikle kişilerin yaş bilgilerini tutmak için uygundur	byte yas; yas=35; MessageBox.Show(yas.ToString());
short	2 Byte	-32768 → +32768	Genellikle sicil numaraları için...	short b,c,d; b=22; c=3; d=Convert.ToInt16(b-c); MessageBox.Show(d.ToString());
int	4 Byte	-2,147,483,648 ile +2,147,483,647	Özellikle maaş alanları için kullanılabilir	int b,c,d; b=22; c=3; d=b/c; MessageBox.Show(d.ToString());
long	8 Byte	-9,223,372,036,854,775,808 +9,223,372,036,854,775,807	Uzun tamsayı ihtiyaçları.....	long b,c,d; b=40055; c=5465; d=b*c; MessageBox.Show(d.ToString());

Ondalıklı Veri Tipleri

Veri Tipinin Adı	Kapladığı Alan	Değer Aralığı	Kullanım Amacı Örnek:	Örnek Uygulama:
float	4 Byte	$-3,4 \times 10^{38}$ ile $+3,4 \times 10^{38}$	virgülden sonra 6-7 digit ile sınırlıdır	float pi; <code>pi=Convert.ToSingle(23)/Convert.ToSingle(7);</code> <code>MessageBox.Show(pi.ToString());</code>
double	8 Byte	$-1,7 \times 10^{308}$ ile $+1,7 \times 10^{308}$	virgülden sonra 15-16 digit ile sınırlıdır	double pi; <code>pi=Convert.ToDouble(23)/Convert.ToDouble(7);</code> <code>MessageBox.Show(pi.ToString());</code>
decimal	16 Byte	+/- 79,228,162,514,264,337,593,543,950,335	virgülden sonra 28 digit ile sınırlıdır	double pi; <code>pi=Convert.ToDecimal(23)/Convert.ToDecimal(7);</code> <code>MessageBox.Show(pi.ToString());</code>

Diğer Veri Tipleri

Veri Tipinin Adı	Kapladığı Alan	Değer Aralığı	Kullanım Amacı Örnek:	Örnek Uygulama:
char	2 Byte	0 ↔ 65536	genellikle tek bir karakter atanması için kullanılır.	char harf; <code>harf='D';</code> <code>MessageBox.Show(harf.ToString());</code>
string	2 Byte x karakter adedi	0 ↔ 2 milyar	uzun karakterlerin tanması için kullanılır	string harf; <code>harf='Murtaza 1900';</code> <code>MessageBox.Show(harf);</code>
bool	2 Byte	True / False	ikili sonuçlar içerecek	bool medeni_hal; <code>medeni_hal=true;</code> <code>MessageBox.Show(medeni_hal.ToString());</code>
DateTime	8 Byte	0:00:00 – Ocak 1, 0001 ile 11:59:59 PM – Aralık 31, 9999	Tarih ve saat bilgilerini tutulur.	DateTime tarih; <code>tarih = DateTime.Now;</code> <code>MessageBox.Show(tarih.ToString());</code>
Object	her şeyin üstüne +4 Byte	istenilen değer. örneğin siz DateTime yerine kullanırsanız 12 byte yer kaplar	Her türlü veri tipi	object tarih; <code>tarih = DateTime.Now;</code> <code>MessageBox.Show(tarih.ToString());</code>

Bazı Durumlara Göre Veri Tipi Tanımlama

Veri Tipinin Adı	Kullanımı:	Kullanım Amacı Örnek: Örnek Uygulama:
Const	Eğer tanımlayacağınız değer PI sayısı gibi sabit ise Const ile deklarasyon yapabilirsiniz.	double alan, yarıcap, cevre; yaricap = 5; Const double pi=3,14; alan=pi * yarıcap^2; cevre=2*pi*yarıcap; MessageBox.Show("alan:"+alan+"cevre:"+cevre);
Public	Bu ifade ile tanımlanan değişkenlere her yerden erişilebilir.	
Private	Sadece tanımlandığı class içinden erişilebilir.	
Internal	Tanımlandığı öğeye sadece Assembly içinde erişilebilir. DLL ya da EXE üzerinden erişilebilir.	
Protected	Tanımlandığı classtan ya da bu classtan üretilen classlardan(miras) erişilebilir.	
Protected Internal	Assembly içinde ve miras alabilen yapılarda bu değere ulaşılabilir.	
Struct	Bir değişken grubu oluşturmak için kullanılır. İlgili class içerisinde tanımlanarak, farklı yapılar içerisinde kullanabilirsiniz.	public struct müşteri { public int maas; public int kira; public int yol; public int kalan;}

Struct ile bir değişken grubu oluşturmak

```
public struct musteri
{
    public int maas;
    public int kira;
    public int yol;
    public int kalan;
}
```

Programın her hangi bir yerinde kullanmak için ideal bir yapıdır.

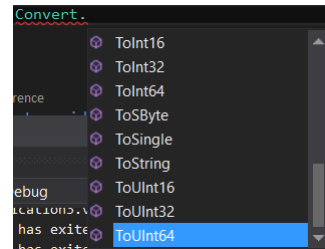
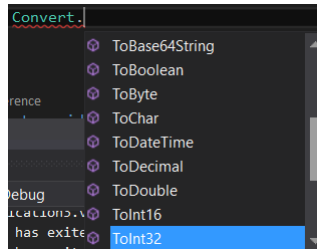
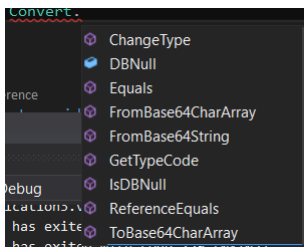
Hem tanımlama işlemini kısaltır.

Hemde hangi alanı ne şekilde tanımladığınızı hafızanızda tutmak zorunda kalmazsınız...

```
private void button3_Click(object sender, EventArgs e)
{
    musteri m = new musteri();
    m.kira = 3000;
    m.maas = 7500;
    m.yol = 3000;
    m.kalan = m.maas - m.kira - m.yol;
    MessageBox.Show(m.kalan.ToString());
}
```

Veri Tipini Dönüştürme

Veri Tipi Dönüşümü	Açıklama	Hangi tiplere Uygulayabiliriz ?
Convert.ToString(deger)	Değeri string'e çevirir	Boolean, Char, Date, Object,Int
Convert.ToInt32(deger)	Değeri Integer'a dönüştür.	Boolean, String, Object
Convert.ToDateTime(deger)	Değeri Date'e dönüştür.	String, Object
Convert.ToDouble(deger)	Değeri Double'a dönüştür.	Boolean, String, Object



Veri Tipini Dönüştürme

```
double a = 35.658;
int b = 0;
b = Convert.ToInt32(a);
MessageBox.Show(b.ToString());
```

```
string ds = null;
string ts = null;
DateTime d;
DateTime t;
ds = "April 20, 1981";
ts = "10:30:00 PM";
d=Convert.ToDateTime(ds);
t=Convert.ToDateTime(ts);
MessageBox.Show(d + " "+t);
```

Değer Atama

- Doğrudan Atama

```
int a = 5;
int b = 25;
int c;
c = a * b;
MessageBox.Show(c.ToString());
```

- Başlangıç Değeri

```
int a;
MessageBox.Show(a.ToString());
```

//hata verir

- Son Atanan Değer

```
int a=25;
a = 30;
a = 43;
MessageBox.Show(a.ToString());
```

//Aynı blok içerisinde
// son atanan değer
// geçerli 43 olur

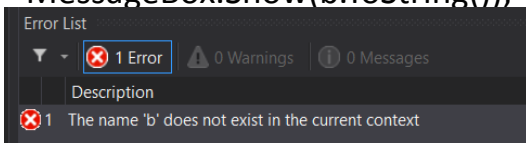
Local ve Global Olma Durumları

//LOCAL DECLARATION

```
int i;
for(i = 1; i<=50;i++)
{
    int b;
    b = b + i;
}
```

```
MessageBox.Show(b.ToString());
```

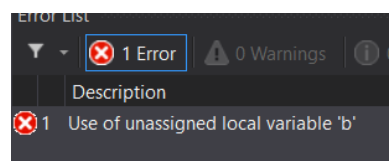
Visual C#
.net'te
değişkenler
sadece
tanımlandıkları
blokta
geçerlidir.



//GLOBAL DECLARATION

```
int i,b;
for(i = 1; i<=50;i++)
{
    b = b + i;
}
MessageBox.Show(b.ToString());
```

b için
Başlangıç
Değeri
verilmemiştir.



Local ve Global Olma Durumları

//GLOBAL DECLARATION

```
int i,b=0;
for(i = 1; i<=50;i++)
{
    b = b +i;
}
MessageBox.Show(b.ToString());
```

Atama Operatörleri

Operatör	Amaç
*=	Çarpma
/=	Bölme
+=	Toplama
-=	Çıkarma
&=	Birleştirme
^=	Üs
\=	Tamsayı
++	Bir Artır
--	Bir Azalt

Orijinal Yazım: a = a + 33;

Kısa Yazım : a+=33;

```
int i = 5;
```

```
i *= 7;
```

```
this.Text = i.ToString (); // sonuç 35
```

```
int i = 5;
```

```
i += 7;
```

```
this.Text = i.ToString (); // sonuç 12
```

Atama Operatörleri

Operatör	Amaç
<code>*</code>	Çarpma
<code>/</code>	Bölme
<code>+</code>	Toplama
<code>-</code>	Çıkarma
<code>&</code>	Birleştirme
<code>^</code>	Üs// XOR power
<code>\</code>	Tamsayı
<code>++</code>	Bir Artır
<code>--</code>	Bir Azalt

```
int a=0, b=0;
for (a=1; a<=5;a++)
{
    b+=a; → b=b+a;
}
MessageBox.Show(b.ToString());
```

```
int i = 5;
i ^= 3;
this.Text = i.ToString ();
```

Kıyaslama Operatörleri

Operatör	Amaç
<code><</code>	Küçük
<code><=</code>	Küçük Eşit
<code>></code>	Büyük
<code>>=</code>	BüyükEşit
<code>==</code>	Eşit
<code>!=</code>	Eşit Değil

Mantıksal Operatörler

Operatör	Amaç
<code>&&</code>	Koşulların Tamamı doğru ise True olarak döner
<code> </code>	Koşulların en az birisi doğru ise
<code>!</code>	Değil Operatörü, ters değer döndürür
<code>^</code>	İki koşulu kıyaslar birisi true birisi false ise TRUE değer dönderir XOR

Aritmatiksel Operatörler

Operatör	Amaç
^	Üs Alma
/	Bölme
*	Çarpma
+	Toplama
-	Çıkarma
\	Tam Sayı Böl
%	Mod Alma

İşlem sırası parantez önceliğini dikkat edildikten sonra:

\wedge , /, *, + ve -

şeklinde.

Kontrol Deyimleri-1

if (koşul)

Koşula göre **True** veya **False** değeri döndürür.

Kullanım Şekli:

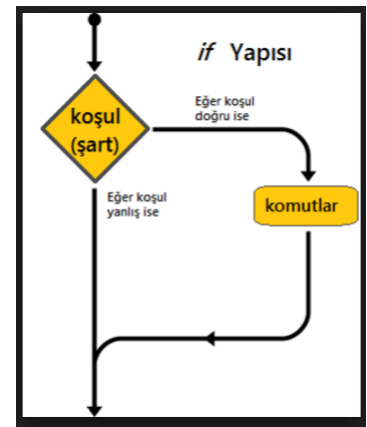
if (koşul)

{

//koşul sağlandığında yapılacak işlem

}

(Yarım Form)

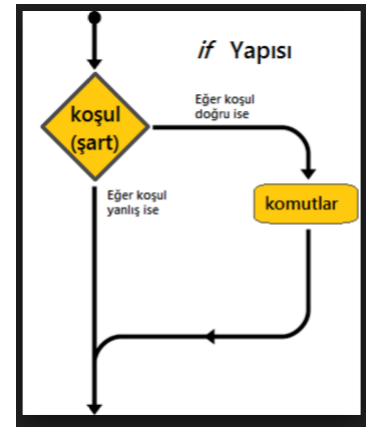


Kontrol Deyimleri-1

Örnek:

```
int a = 900;
if (a>25)
{
    a=a+35;
}
MessageBox.Show(a.ToString());
```

(Yarım Form)



Kontrol Deyimleri-2

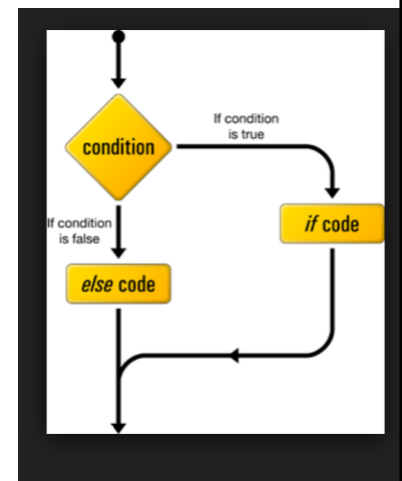
if – else yapısı

İki değeri kıyaslamak için kullanılır. Koşul doğru ise if bloğundaki komutlar, yanlış ise else bloğu çalışır

Kullanım Şekli:

```
if (koşul)
{
    //koşul sağlandığında yapılacak işlem
}
else
{
    // koşul sağlanmadığında yapılacak işlem
}
```

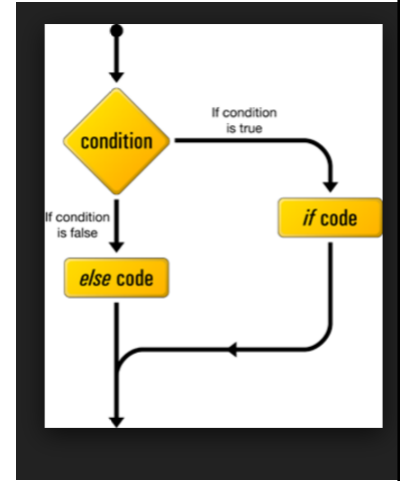
(Tam Form)



Kontrol Deyimleri-2

```
private void button3_Click(object sender, EventArgs e)
{
    int a = 300;
    if (a > 255)
    {
        a = a + 35;
    }
    else
    {
        a = a + 450;
    }
    MessageBox.Show(a.ToString());
}
```

(Tam Form)



Kontrol Deyimleri-3

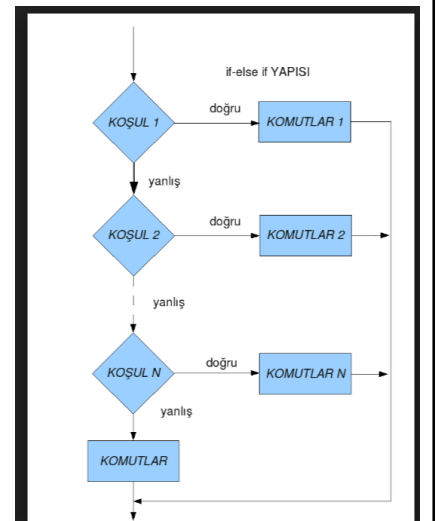
if -- else if – else yapısı

İkiden fazla koşul için kullanılır ve bu yapıda ilk True değerini döndüren koşul sağlandığında sona erer.

Kullanım Şekli:

```
if (koşul1)
{
    //koşul1 sağlandığında yapılacak işlem
}
else if (koşul 2)
{
    // koşul2 sağlandığında yapılacak işlem
}
else
{
    // belirtilen N adet koşulda uymaz ise burası çalışır
}
```

(Çok Koşullu Form)

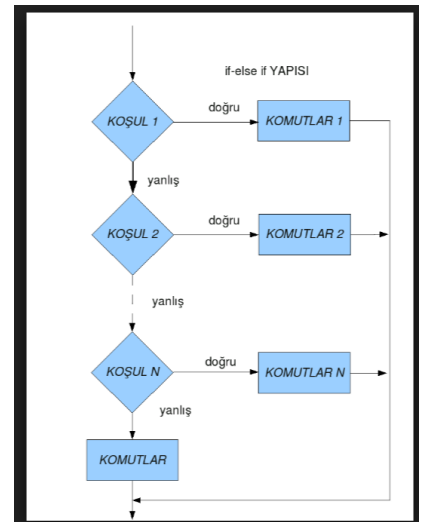


Kontrol Deyimleri-3

if -- else if -- else yapısı

```
double a = 10;
if (a>10)
{
    a = a+50;
    a=a/4;
    a=a*1.5;
}
else if (a<10)
{
    a=a+5;
}
else
{
    a=a-70;
}
MessageBox.Show(a.ToString());
}
```

(Çok Koşullu Form)



Kontrol Deyimleri-4

switch...case

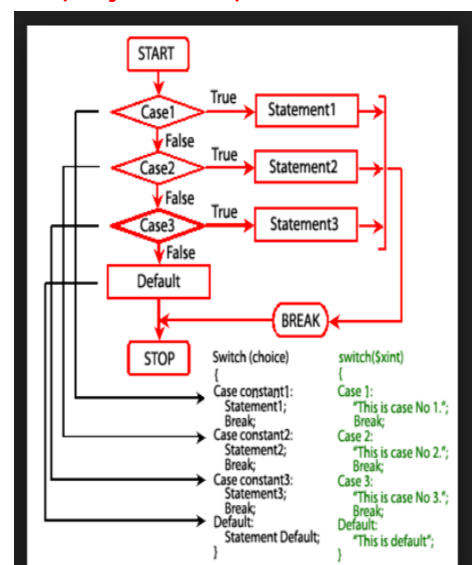
switch (değişken)

```
{
    case deger1 :
        islem1;
        break;
    case deger2 :
        islem2;
        break;
    .....
    .....
    case degerN :
        islemN;
        break;
    default :
        varsayılan işlem;
        break;
}
```

```
int a = 75;
switch (a)
{
    case 50:
        MessageBox.Show("değer 50");
        break;
    case 75:
        MessageBox.Show("değer 75");
        break;
    default:
        MessageBox.Show("geçersiz");
        break;
}
```

kullanıcı
istediği
değeri
kendisi
girsin?

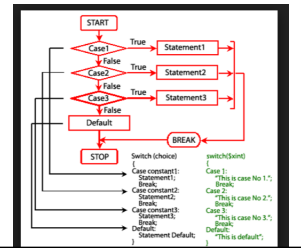
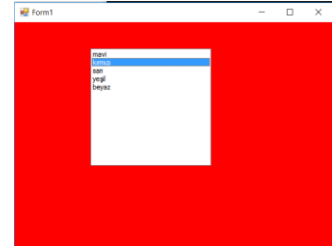
(Seçili Form)



Kontrol Deyimleri-4

```
private void listBox1_SelectedIndexChanged (object sender,
EventArgs e)
```

```
{
    switch (listBox1.Text)
    {
        case "mavi": this.BackColor = Color.Blue; break;
        case "sarı": this.BackColor = Color.Yellow; break;
        case "beyaz": this.BackColor = Color.White; break;
        case "kırmızı": this.BackColor = Color.Red; break;
        case "yeşil": this.BackColor = Color.Green; break;
    }
}
```



(Seçili Form)

Uygulama_1

0 İLE 100 arasında not girişine izin veren, Ara sınav ve Final notları kullanılarak geçme notu hesaplayan ve bu geçme notuna karşılık gelen harf notu belirleyen program kodunu yazınız.

NOT ARALIKLARI
HARF KARŞILIĞI
89.5 ile 100 AA
79.5 ile 89.4 BA
69.5 ile 79.4 BB
59.5 ile 69.4 CB
49.5 ile 59.4 CC
39.5 ile 49.4 DC
0 ile 39.4

Uygulama _2

Girilen ağırlık ve boy verilerine göre vücut-kitle indeksi belirleniyor ve kişinin girdiği ağırlık sınıfı sunuluyor. ($vct = y / (x * x)$);



Merak Ediniz!

- "Ternary" ve "İç içe Ternary" nedir? Nasıl kullanılır?
- TryParse Nedir? Niçin kullanılır? araştırınız..