

Label(Etiket) Elemanı ve Özellikleri:

Form üzerinde bilgi göstermek için kullanılır. Bilgi çıkış elemanıdır. Diğer elemanlarda olan Name, Text, ForeColor, BackColor, Font, Visible gibi özellikler bunda da vardır. Üzerine yazılacak bilgi metin tipinde olmalıdır. Sayı tipinde bilgi yazılacaksa metin tipine çevrilmelidir.

ÖR: `x=10; ise label2.Text=x.ToString();` ya da `label2.text=c + " "`;

ÖR: Üzerindeki yazıyı değiştirmek için: `Label1.Text="MERHABA";`

Checkbox Elemanı: (İşaret kutusu) CheckBox

Birden fazla seçenekten çok sayıda seçmeyi sağlar. Programa veri girişi sağlar. Kullanıcının seçimlerini programa aktarmaya yarar.

Özellikler:

Text: Kutunun yanındaki yazı.

En çok kullanılan özelliği **checked** özelliğidir. Kutu seçili ise bu özelliği true, değilse False değeri alır. Program içinde if yapıları kullanarak kutunun seçili olup olmadığı kontrol edilir.

`Checkbo1.checked== true` ise kutu seçili anlamındadır.

Olayları : En çok kullanılan olay `checkedchanged` olayıdır. (Bir kutuya tıklandığında seçili durumu değiştiği zaman bu olay meydana gelir. Ayrıca bir butonun `click` olayında da kullanılabilir.)

RadioButton Elemanı (seçenek kutusu) RadioButton

Birden fazla seçenek içinden sadece bir tanesinin seçilmesini sağlar. Bir seçenek seçili iken başka bir seçenek seçilirse, önceki seçili olan seçili durumdan çıkar.

Özellikler: Daha önceki checkbox elemanında bahsedilen özellikler burada da vardır. Hangi radio butonun seçili olduğu kod ortamında if yapısı içinde `checked` özelliği kontrol edilerek bulunur. `Radiobutton1.checked== true` ise seçili demektir.

DEĞİŞKENLER:

Programa kullanıcının girdiği yada program içinde elde edilen bilgilerin saklandığı hafız yapılarına değişkenler denir. Bütün programlar dışarıdan bilgi alırlar ve yeni bilgiler üretirler. Bu bilgilerin program boyunca değişken denilen yapılarda tutulurlar. Gerekliğinde değerleri değişir, değerleri kullanılabilir (örneğin hesaplamalarda ya da form üzerinde yazdırma). Bilgi konulmadan önce değişkenler tipi ve ismi belirtilerek tanımlanmalıdır.

Değişkene isim verme: İsim verilirken bazı kurallara uymak lazım: Türkçe karakter kullanılamaz, Alt tire (_) haricinde özel karakter kullanılamaz. Boşluk içeremez. C# ait özel kelimeler olmaz (Ör: int, if, for, vb). Rakamlar başlayamaz. Bu özelliklere dikkat edilerek istenilen isimler verilebilir. Anlamlı

isimler olmak zorunda değildir. Ama genelde içine konulan bilgiyi hatırlatıcı isimler verilir. Ör: Sınav notunu koymak için vize, final, snotu, gibi isimler verilebilir.

Değişkene verilen isim program içinde değiştirilemez. Aynı şekilde kullanmak gerekir.

C# dili küçük harf /büyük harf duyarlıdır. Yani Vize ile vize farklı değişkenlerdir.

Tip Belirleme: Değişken içine konulacak bilginin türüne göre tip belirlenir. Bir değişkene;

Tam sayı bilgi konulacaksa int ; ondalık sayısal bilgi konulacaksa double veya float; metin bilgi konulacaksa string ; 1 karakterlik bilgi konulacaksa char tipinde tanımlanmalıdır.

ÖRNEKLER:

```
int x,y,z;
```

```
double ort, oran=12.45 ,a,sayi;
```

```
string okul,isim, adres;
```

```
isim="Murat";
```

NOT: Ondalık türünde tanımlanmış bir değişkene tam sayı bilgi konabilir.Ancak tam sayı türünde tanımlanmış bir değişkene ondalık bilgi konulamaz. Değişkenlere bilgi konurken bilginin türüne dikkat etmek gerekir. Değişkenin türüne uygun bilgi konulmalıdır. Eğer farklı bilgi konulacaksa, tip çevirme işlemi yapılmalıdır. Örneğin, textbox lardan okunan bütün bilgiler metin formatındadır. Okunan sayısal bilgi, sayısal olarak tanımlanmış bir değişkene konulmak için tip çevirimi yapmalıdır.

```
ÖR: int x=Convert.ToInt16(textbox2.text)
```

```
ÖR: double y=Convert.ToDouble(textbox1.text);
```

```
Ör: label1.text=x.ToString(); veya label1.text=x + " ";
```

KARAR YAPILARI

Belli bir şartın Doğru ya da yanlış olmasına göre program akışını değiştiren yapılardır. Şart ifadesi sonucu Dogru/Yanlış olan ifadelerdir. İçinde >,<, >=,<=, ==, != gibi karşılaştırma işaretleri olabilir. Şart ifadesi basit olabilir (x>10), veya birsen fazla şart ifadesi && (ve), || (veya) bağlaçları ile bağlanmış karmaşık ifadeler olabilir. Ör: ((x>3) && (y==3)) || (notu>=55)

Şart ifadesi ne kadar karışık olursa olsun, sonuçta değeri tek bir değerdir (Doğru veya yanlış). İşte bu şartın değerine göre program akışını yönlendiren yapılara Karar yapıları denir.

En çok kullanılan karar yapısı if () yapısıdır.

Genellikle 3 türü vardır:

<p>1)Basit if yapısı: if(şart) { } . . Şart doğru ise if altındaki blok çalışır. Şart yanlış ise bu blok çalışmadan atlanır.</p>	<p>2) if(şart) { } Else { } Şart doğru ise hemen altındaki blok çalışır. Yanlış ise else altındaki blok çalışır.</p> <p>İf= Eğer Else= Değilse</p>	<pre>if(şart1) { } else if (şart 2) { } else { } Şart 1 doğru ise hemen altındaki blok çalışır, diğerleri atlanır.şart1 yanlış ise şart2 kontrol edilir. Şart2 doğru ise hemen altındaki blok çalışır, o da yanlış ise en sonraki else kısmı çalışır. bu tür ifler birbirine bağlı birden fazla şart olduğu zaman kullanılır.</pre>
---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

NOT: Bir problemin içinde –se, -sa koşul ifadeleri varsa ihtimaller varsa, if karar yapıları kullanılacak demektir. Problemden bulunan bütün ihtimaller kodlanır, program çalışırken, şartların durumuna göre bazı yerler çalışır, bazı yerler atlanır.

İf yapıları birbiri yerine kullanılabilir. Şartlar ve if sayıları iyi ayarlanmalıdır.

ÖRNEK PROGRAM: textboxlara girilen vize ve final notlarını alarak, not ortalamasını bulup $ort \geq 55$ ise label3 e GEÇTİ, değilse KALDI yazdıran program.

The screenshot shows a Windows application window titled 'Form1'. Inside the window, there are two text input fields. The first is labeled 'VİZE' and the second is labeled 'FİNAL'. Below these fields, there is a label 'label3' and a button labeled 'HESAPLA'. The window has standard Windows window controls (minimize, maximize, close) in the top right corner.

```

private void button1_Click(object sender, EventArgs e)
{
    double v, f, ort;
    v = Convert.ToDouble(textBox1.Text);
    f = Convert.ToDouble(textBox2.Text);
    ort = v * 40 / 100 + f * 60 / 100;
    if (ort >= 55)
        label3.Text = "GEÇTİ";
    else
        label3.Text = "KALDI";
}

```

ÖRNEK: Malzeme adeti ve birim fiyatı textboxlardan alınarak, %18 KDV dahil seçili ise KDV dahil fiyatını bulacak, seçili değilse Normal fiyatı bulacak program.

```

private void button1_Click(object sender, EventArgs e)
{
    double adet, bf, tutar=0;
    adet = Convert.ToDouble(textBox1.Text);
    bf= Convert.ToDouble(textBox2.Text);

    if (checkBox1.Checked == true)
        tutar = bf * adet + bf * adet * 18 / 100;
    else
    { tutar = bf * adet; }
    label3.Text = tutar + " TL";
}

```

ÖRNEK: İşçi normal maaşı yanı sıra ürettiği parça sayısına göre ek ücret alıyor. Parça sayısı > 0 ve <=3 ise, parça başına 5 tl,; parça sayısı >3 ve <=10 ise, parça başına 10 tl, parça sayısı >10 ise parça başına 15 tl alıyor. Buna göre normal ücret ve parça sayısını alarak toplam ücreti bulan program.

```
private void button1_Click(object sender, EventArgs e)
{
    double normal, ps, ek=0, toplam;
    normal = Convert.ToDouble(textBox1.Text);
    ps= Convert.ToDouble(textBox2.Text);
    if (ps <= 3)
        ek = ps * 5;
    if((ps>3)&& (ps<=10))
        ek= ps*10;

    if(ps>10)
        ek=ps*15;
    toplam=normal+ek;
    label3.Text=toplam+" TL";
}
```

ÖRNEK: Restaurant menüsünden isteğe göre seçerek , hesabı bulan program.

```
private void button1_Click(object sender, EventArgs e)
{
    double tutar = 0;

    if (checkBox1.Checked == true)
```

```

        tutar = tutar + 15;
    if (checkBox2.Checked == true)
        tutar = tutar + 10;
    if (checkBox3.Checked == true)
        tutar = tutar + 20;
    if (checkBox1.Checked == true)
        tutar = tutar + 5;

    label3.Text = tutar + " TL";

}

```

ÖRNEK: Girilen 3 sayıdan en Büyükünü bulan program



```

private void button1_Click(object sender, EventArgs e)
{
    double s1,s2,s3,eb=0;
    s1 = Convert.ToDouble(textBox1.Text);
    s2 = Convert.ToDouble(textBox2.Text);
    s3 = Convert.ToDouble(textBox3.Text);

    if (s1 > s2 && s1 > s3)
        eb = s1;
    if (s2 > s1 && s2 > s3)
        eb = s2;
    if (s3 > s2 && s3 > s1)
        eb = s3;
    label3.Text = eb.ToString();
}

```

ÖRNEK: $ax^2+bx+c=0$ türündeki bir denklemin katsayılarını alarak, köklerini bulan program.

```
private void button1_Click(object sender, EventArgs e)
{
    double a,b,c,x1,x2,d;
    a = Convert.ToDouble(textBox1.Text);
    b = Convert.ToDouble(textBox2.Text);
    c = Convert.ToDouble(textBox3.Text);
    d = b * b - 4 * a * c;

    if (d >= 0)
    {
        x1 = (-b - Math.Sqrt(d)) / 2 * a;
        x2 = (-b + Math.Sqrt(d)) / 2 * a;
        label3.Text = x1.ToString();
        label6.Text = x2.ToString();
    }
    else
    {
        label3.Text = "KÖK YOK";
        label6.Text = " ";
    }
}
```