

Metodlar

Program yazarken bazı durumlarda bir işi gerçekleştiren kod yapılarını tekrar tekrar yazmak durumunda kaldığımız olur. Bu kodları yeniden yazmak yerine metodları kullanabiliriz.

C# içerisinde geriye değer döndüren ve değer döndürmeyen metodlar olmak üzere iki çeşit metod yapısı kullanılır.

Değer Döndürmeyen Metodlar

Kendi ismi ile çağrılarak içerisinde yazılmış olan kod satırlarını çalıştıran ve geriye herhangi bir değer döndürmeyen kod bloğudur. Değer döndürmeyen metodlar void anahtar sözcüğü ile başlarlar.

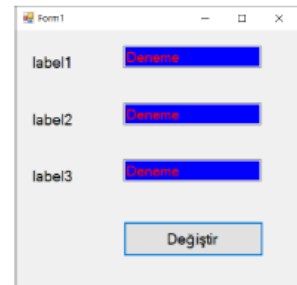
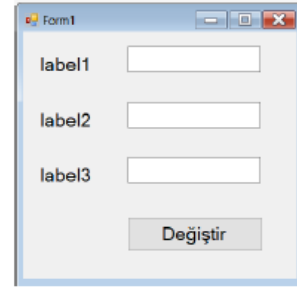
Kullanımı:

```
void metod_ismi(parametre)
{
    Metodun içerisinde çalışacak kodlar;
}
```

Metod içerisinde bir kez yazacağımız kod, bir daha yazılmasına gerek kalmadan sadece ismi ile çağrılarak defalarca çalıştırılabilir.

Örnek

```
public partial class Form1 : Form
{
    1 başvuru
    public Form1()
    {
        InitializeComponent();
    }
    1 başvuru
    void Renk_Degistir()
    {
        foreach(Control nesne in this.Controls)
        {
            if((nesne) is TextBox )
            {
                nesne.Text = "Deneme";
                nesne.BackColor = Color.Blue;
                nesne.ForeColor = Color.Red;
            }
        }
    }
    1 başvuru
    private void button1_Click(object sender, EventArgs e)
    {
        Renk_Degistir();
    }
}
```

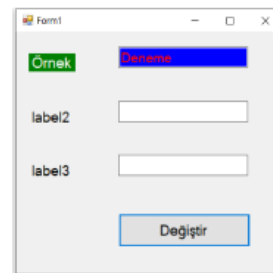


Metod İçerisinde Zorunlu Parametre Kullanımı

```
public partial class Form1 : Form
{
    1 başvuru
    public Form1()
    {
        InitializeComponent();
    }
    2 başvuru
    void Renk_Degistir(Control nesne)
    {
        if((nesne) is TextBox )
        {
            nesne.Text = "Deneme";
            nesne.BackColor = Color.Blue;
            nesne.ForeColor = Color.Red;
        }
        else if ((nesne) is Label)
        {
            nesne.Text = "Örnek";
            nesne.BackColor = Color.Green;
            nesne.ForeColor = Color.White;
        }
    }
    1 başvuru
    private void button1_Click(object sender, EventArgs e)
    {
        Renk_Degistir(textBox1);
        Renk_Degistir(label1);
    }
}
```

Metod içerisine, dışarıdan parametreler ile bilgi göndermek mümkündür.

Renk_Değiştir metoduna parametre olarak kontrol türünden olan nesne değerini gönderiyoruz. Gönderilen nesne parametresinin türüne göre metod içerisinde farklı işlemler yapılıyor.



Metod İçerisinde Opsiyonel Parametre Kullanımı

```
void Renk_Degistir(Control nesne, string yazı="Örnek Değer", Boolean egik=false)
{
    if (egik == true)
        nesne.Font = new Font(Font, FontStyle.Italic);
    if ((nesne) is TextBox )
    {
        nesne.BackColor = Color.Blue;
        nesne.ForeColor = Color.Red;
    }
    else if ((nesne) is Label)
    {
        nesne.BackColor = Color.Green;
        nesne.ForeColor = Color.White;
    }
    nesne.Text = yazı;
}
1 bagvuru
private void button1_Click(object sender, EventArgs e)
{
    Renk_Degistir(label1,"Adı Soyadı");
    Renk_Degistir(textBox1,"Özcan EKEN",true);
    Renk_Degistir(label2,egik:true);
    Renk_Degistir(textBox2, "LMYO");
}
```

Kullanımı zorunlu olmayan parametreleri yazmadan **Optional Parameters** yöntemi ile diğer parametrelere değer aktarımı yapabiliriz.

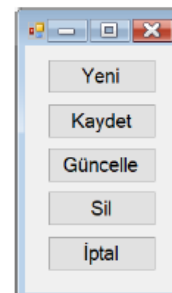
Kullanmadığımız parametreler metod tanımlanırken verilen **default (varsayılan)** değeri alır.

params Deyimi ile Parametre Kullanımı

Parametre sayısının belli olmadığı durumlarda **params** deyimi ile metod içerisinde bu parametreleri bir parametre dizisi şeklinde tanımlayabiliriz. **params** ile tanımladığımız dizi değişkeni metodun son parametresi olarak tanımlanmalıdır.

```
void pasif (bool onay, params Button[] btn)
{
    byte i=0;
    if(onay==true)
    {
        for(i=0;i<=btn.Length-1;i++)
        {
            btn[i].Enabled = false;
        }
    }
    else
    {
        for (i = 0; i <= btn.Length - 1; i++)
        {
            btn[i].Enabled = true;
        }
    }
}
```

```
private void Form1_Load(object sender, EventArgs e)
{
    pasif(true, btnGuncelle, btnIptal, btnKaydet, btnSil);
}
1 bagvuru
private void btnYeni_Click(object sender, EventArgs e)
{
    pasif(false, btnIptal, btnKaydet);
    pasif(true, btnYeni, btnGuncelle, btnSil);
}
```



Değer Döndüren Metodlar

Kendi ismi ile çağrılarak içerisinde yazılmış olan kod satırlarını çalıştıran ve geriye değer döndüren kod bloğudur.

Kullanımı:

GeriDönecek_DeğerinTipi **metod_ismi**(parametre)

{

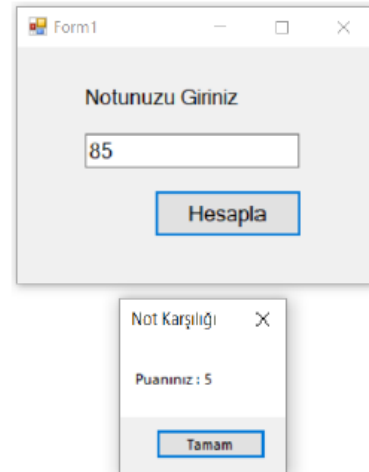
Metodun içerisinde çalışacak kodlar;

return GeriDönecekDeğer;

}

Örnek

```
byte NotHesapla(byte notu)
{
    byte puan = 0;
    if (notu < 25)
        puan = 0;
    else if (notu < 45)
        puan = 1;
    else if (notu < 55)
        puan = 2;
    else if (notu < 70)
        puan = 3;
    else if (notu < 85)
        puan = 4;
    else if (notu <= 100)
        puan = 5;
    else
        MessageBox.Show("Yanlış Not Girdiniz");
    return puan;
}
1 başvuru
private void button1_Click(object sender, EventArgs e)
{
    byte notdegeri, sonuc;
    notdegeri = Convert.ToByte(textBox1.Text);
    sonuc = NotHesapla(notdegeri);
    MessageBox.Show("Puanınız : " + sonuc, "Not Karşılığı");
}
```



Örnek

```
byte NotHesapla(byte yazılı, byte sozlu=50)
{
    byte puan = 0;
    int notu = 0;
    notu=(yazılı + sozlu)/2;
    if (notu < 25)
        puan = 0;
    else if (notu < 45)
        puan = 1;
    else if (notu < 55)
        puan = 2;
    else if (notu < 70)
        puan = 3;
    else if (notu < 85)
        puan = 4;
    else if (notu <= 100)
        puan = 5;
    else
        MessageBox.Show("Yanlış Not Girdiniz");
    return puan;
}
```

```
private void button1_Click(object sender, EventArgs e)
{
    byte yazilınotu,sozlunotu,sonuc;
    yazilınotu = Convert.ToByte(textBox1.Text);
    if (textBox2.Text == "")
        textBox2.Text = 0.ToString();
    sozlunotu = Convert.ToByte(textBox2.Text);
    if (Convert.ToByte(textBox2.Text) == 0)
        sonuc = NotHesapla(yazilınotu);
    else
        sonuc = NotHesapla(yazilınotu, sozlunotu);
    MessageBox.Show("Puanınız : " + sonuc, "Not Karşılığı");
}
```

