

PROGRAMLAMAYA GİRİŞ

Bir problemin bilgisayar ortamında bir programlama dili yardımıyla çözümlenmesine programlama denir. Bilgisayarda çalıştırdığımız bütün programlara yazılım denir ve bir programlama dili ile yapılmıştır. Yazılımları 3 gruba ayırabiliriz.

1)İşletim sistemleri: Mutlaka olması gereken yazılımlardır(Windows, Linux, Android, iOS, UNIX vb..) Bilgisayarların kullanılmasını sağlar, kaynak paylaşımı yaparlar, yazılım ve donanımların kullanılmasını sağlarlar.

2)Paket Programlar: Her hangi bir problemin çözümü için yazılmış programlardır. Örneğin AutoCad, oyunlar, muhasebe programları, ofis programları vs..

3)Programlama Dilleri: Program yazmaya yarayan özel yazılımlardır. Günümüzde 1500 e yakın programlama dilleri vardır. Hepsinin kendine özgü özellikleri vardır. Günümüzde C, C++,C#, Java, Visual Basic, ASP, PHP vb. gibi programlama dilleri kullanılmaktadır.

Programlamada kullanılan dilden ziyade algoritma önemlidir. **Algoritma:** Bir problemin çözüm adımlarıdır. Problemin çözümünü yapamadığınız zaman programını da yapamazsınız.

Hangi dili kullanırsanız kullanın, o dilde program yazabilmek için o dilin **Derleyici(Compiler)** denilen programına ihtiyacınız vardır. Derleyici programlar; yazılan programın o dilin kurallarına uyup uymadığını kontrol ederler, yazım hatalarını gösterirler.

Programlamada hatalar 2 ye ayrılır:

1)Yazım hataları(syntax error): Programın yazımında yapılan hatalardır. Derleyici bulabilir ve düzeltilmesini ister. Yazım hataları programlar çalışmaz.

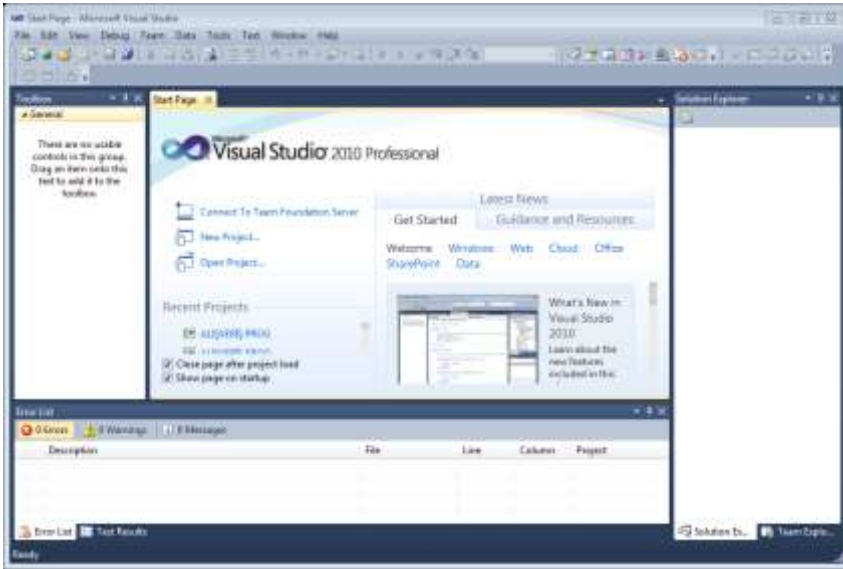
2)Mantık hataları(run- time error): Programın algoritmasında yapılan mantık hatalarıdır. Derleyici bulamaz. Mantık hatalı bir program çalışır ancak yanlış çalışır. Program çalışırken hatalı olduğu ortaya çıkar.

Görsel Programlama: Programlama yaparken görsel elemanlar (metin kutusu, etiket, açılır listeler, butonlar vb.)ve kodların kullanıldığı olay tabanlı programlamadır. Programın çalışması kullanıcının verdiği olaylar doğrultusunda devam eder. Hangi olay olduğunda ne yapılması isteniyorsa, kodlar o elemanın ilgili olayı içine yazılır.Örneğin, butona tıklanıldığında, girilen verilerin işlenmesi ve kaydedilmesi isteniyorsa, o işleri yapan kodlar butonun tıklama (button_click) olayı içine yazılmalıdır.

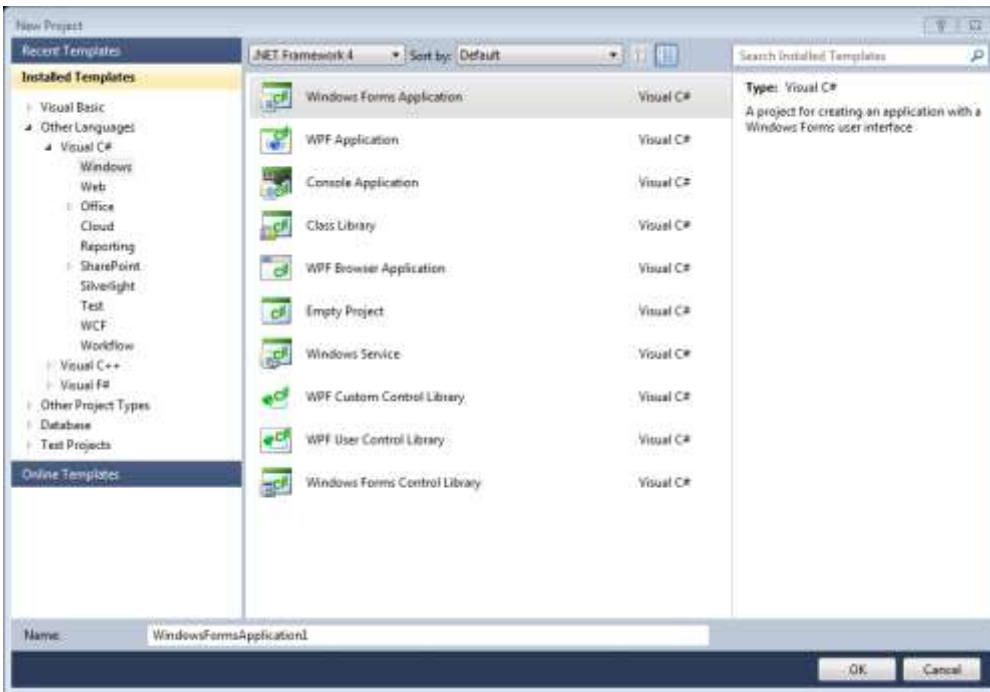
Kullanılan her elemanı kendine özgü özellikleri ve tanımlı olayları vardır. Örneğin button elemanının click olayı, textbox elemanının textchanged (metin değişme) olayı vardır. Bazı özellikler ortaktır, her elemenda bulunur.

Görsel programlama **dizayn ve kodlama** olarak 2 aşamada yapılır. Öncelikle programın arayüzü görsel elemanlarla tasarlanır. Daha sonra hangi elemanın hangi olayı olduğundan ne iş yapılması isteniyorsa, o olay içine kodlar yazılır.

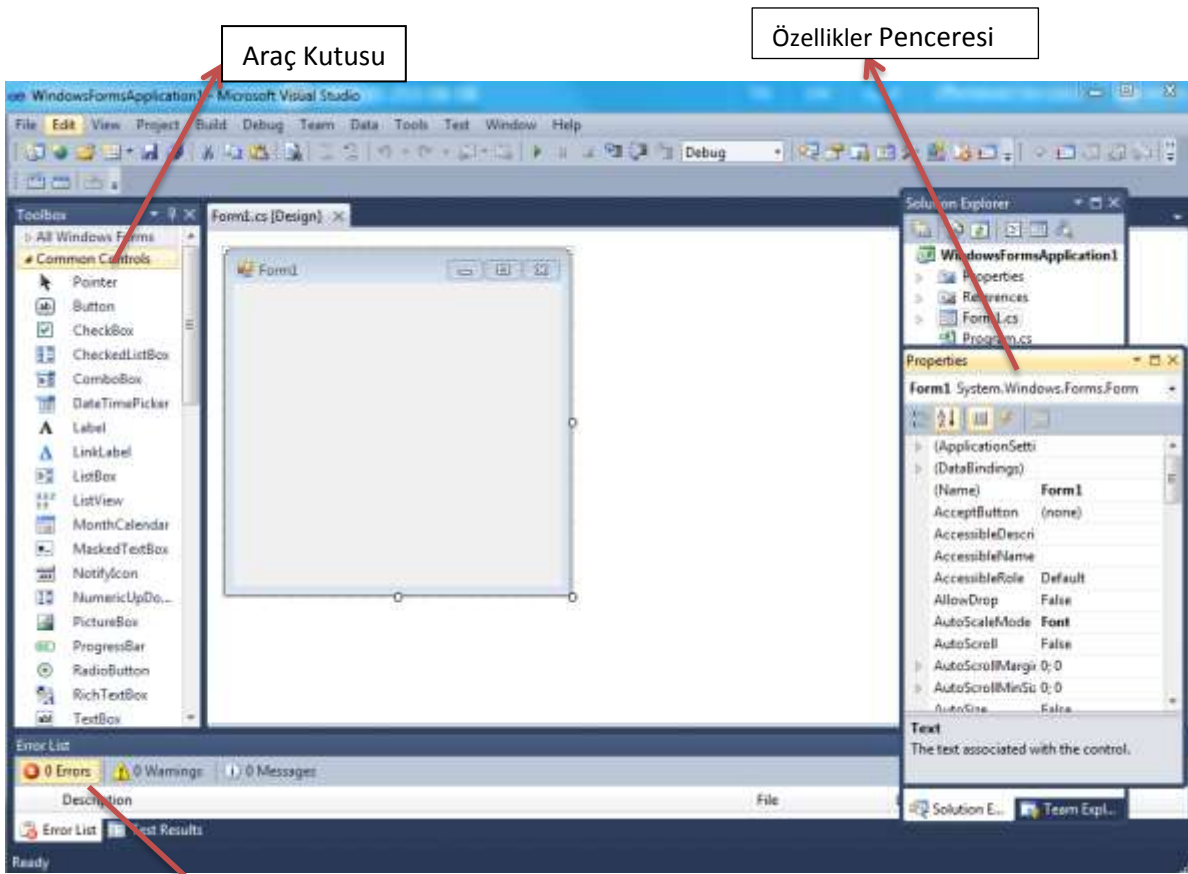
Görsel programlama yapılırken en sık kullanılan program geliştirme ortamı (IDE) Visual Studio nun değişik versiyonlarıdır. Biz derslerimizde **Visual Studio 2010** versiyonunu kullanacağız. Bu programı kullanarak **Visual C#** dili ile program geliştireceğiz.



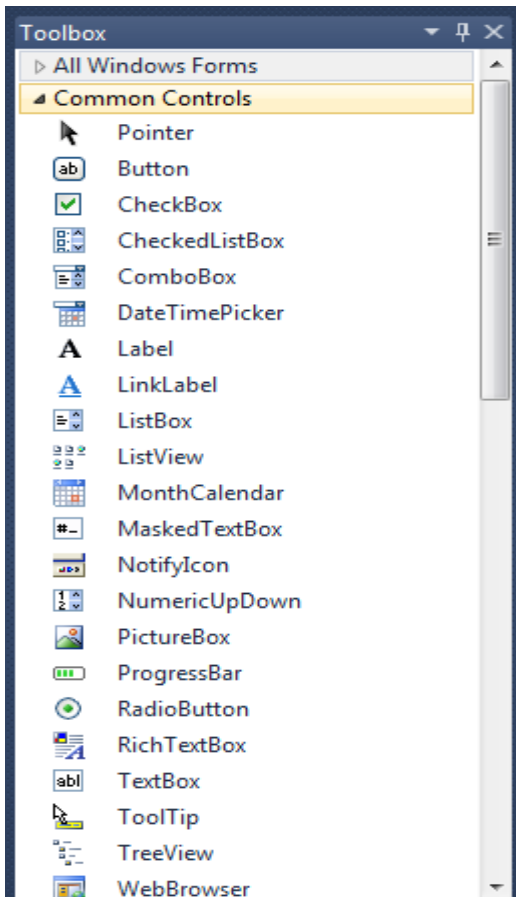
Programa Giriş Ekranı



Yeni projeye başlangıç ekranı (Burada Windows Form Application seçilerek OK butonuna tıklanır)



Hata listesi



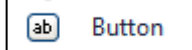
Program yazma ortamında genellikle sağ tarafta bulunan araç kutusu (toolbox), program yaparken kullanacağımız elemanların topluca bulunduğu penceredir. Sağ tarafta bulunan **Properties**(özellikler penceresi) seçili olan elemanın özelliklerini ve özellik değerlerini gösterir. Dizayn aşamasında bir çok elemanın özelliklerini buradan ayarlayabiliriz.

Altteki **Hata listesi (Error List)** penceresinde programı yazarken yapılan hatalar listelenir. Bu hatalar aynı zamanda kod kısmında da altı kırmızı ile çizilerek hata olduğu belirtilir.

Toolbox da ki elemanları fare ile tutup form üzerine sürükleyerek ve orada yeniden boyutlandırarak form üzerine yerleştirilir. Kendimiz isim verebildiğimiz gibi sistem otomatik olarak da isimlendirme yapar (Örneğin: button1, button2, button3 vs)

Kodlama yaparken elemanları ve özelliklerini yazarken otomatik açılan listeden yararlanıla bilinir. Böylelikle yazım hatası yapılmaz. Elemanın isminin birkaç harfini yazdığınızda açılan listeden seçerek kullanırsınız. Özelliklerini yazacağınız zaman . koyduktan sonra açılan listeden ilgili özellik seçilebilir.

ÇOK KULLANILAN ELEMANLAR:



1) Button Elemanı : Genellikler içine yazılan kodları çalıştırmak için (click olayı) kullanılan bir elemandır. Bir kod bloğunu çalıştırmak amacıyla form üzerinde kullanılır. Button elemanının bazı özellikleri dizayn ortamında sağ taraftaki properties penceresinden ayarlanabilir.

Bazı özellikleri:

a) **Text Özelliği:** Butonun üzerindeki yazıyı belirler. Dizaynda Text özelliğine istenilen metin yazılarak buton üzerinde çıkması sağlanır. Ayrıca kod ortamında da belirlenebilir. ÖR: `button1.Text = "GİRİŞ YAP"` ifadesi ile buton1 in üzerinde GİRİŞ YAP yazısının çıkması sağlanır.

b) **Name Özelliği:** Butonun adını belirler. Kod ortamında bu isim kullanılır. Siz de belirleyebilirsiniz, otomatik verilen ismi de kullanabilirsiniz.

c) **BackColor:** Buton un üstündeki zemin rengini belirler. Dizayn aşamasında belirleyebilirsiniz. Ya da kod ortamında da yapabilirsiniz.

`Button1.BackColor=Color.Red` (Renk vermek için Color nesnesi kullanılır. İstenilen renk seçilir.)

d) **Enabled:** Butonun aktif /pasif olmasını belirler. Pasif olunca tıklanmaz. True veya False değeri alabilir. True ise aktif, false ise pasif hale gelir. `Button1.Enabled= false` yazılırsa buton pasif hale gelir.

e) **Font:** Buton üzerindeki yazının tipini belirler.

f) **ForeColor:** Buton üzerindeki yazının rengini belirler. `Button1.ForeColor=Color. Green` (yazının rengini yeşil yapar)

g) **Visible Özelliği:** Butonun görünüp/ gizlenmesini sağlar. True yapılırsa Görünür. False yapılırsa Gizlenir.

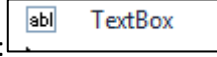
`Button1.visible=false` (buton gizlenir)

Olayları:

a) **click Olayı :** Buton için en çok kullanılan olaydır. Dizayn aşamasında buton üzerine çift tıkladığında click olayı içine kodları yazacağınız yer açılır.

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

Diğer olayları zamanla anlatılacaktır.



2) **Textbox Elemanı(metin kutusu)** : Veri girişi için kullanılan elemandır. Kullanıcıdan program içine bilgi alımında kullanılır. Tolbox dan çift tıklayarak ya da sürükleyip bırakarak form üzerine yerleştirilir.

Girilen veriler metin formatında programa alınır. Eğer sayısal veriler girilecekse, program içinde uygun sayı tipine çevrildikten sonra matematiksel işlemlerde kullanılır.

`X=Convert.ToInt16(textbox1.text);` (textbox1 içine girilen tam sayı bilgisini sayıya çevirip x içine koyar)

`Y=Covert.ToDouble(textbox2.text)` (textbox2 içine girilen ondalık sayı bilgisini ondalık sayıya çevirip y içine koyar)

`İsim=textbox3.text` (textbox3 e girilen metin bilgiyi isim değişkenine koydu. Metin olduğu için çevirmeye gerek yok)

`Textbox4.text=textbox1.text+ textbox2.text;` (text1 deki bilgiye text2 deki bilgi eklenip (sonuna yapıştırıp) text4 e koyar)

NOT : + işareti etrafındaki bilginin en az biri metin türünde ise toplama anlamı yoktur.Bilgiyi bir birine ekler.

NOT: C# da her komut satırı ; ile biter.

Textbox Özellikleri:

a) **Text**: metin kutusu içindeki yazı demektir. Program çalışırken içine yazılan yazıya program ortamında kullanmak için kullanılır. Ör. `Textbox1.text` text1 e girilen yazı demektir. Kod ortamında da içine bilgi yazılabilir.

ÖR: `text1box1.Text=" HARRAN ÜNİVERSİTESİ";`

ÖR: `textbox1.text=textbox2.text+textbox3.text;`

b) **Name** : metin kutusunun adıdır. Kodlarda metin kutusuna erişmek için adı kullanılır.

c) **BackColor**: Textbox ın zemin rengini belirler. Dizayn aşamasında belirleyebilirsiniz. Ya da kod ortamında da yapabilirsiniz.

ÖR: `textbox1.BackColor=Color.Pink ;`

d) **Enabled**: Metin kutusunu aktif /pasif olmasını belirler. Pasif olunca veri girişi yapılamaz. True veya False değeri alabilir. True ise aktif, false ise pasif hale gelir. `Textbox1.Enabled= false` yazılırsa metin kutusu pasif hale gelir.

e) **Font**: metin kutusundaki yazının tipini belirler.

f) **ForeColor**: Metin kutusu üzerindeki yazının rengini belirler. `Textbox1.ForeColor=Color. Green` (yazının rengini yeşil yapar)

g) **Visible Özelliği**: Textbox ın görünüp/ gizlenmesini sağlar. True yapılırsa Görünür. False yapılırsa Gizlenir.

`Textbox1.visible=false` (kutu gizlenir)

h) **passwordchar**: Metin kutusu şifre girişinde kullanılacaksa girilen harfin görünmemesi için bu özellikte belirlenen karakter görünür. Girilen bilgi kod ortamında normal olarak kullanılır.