

# Visual C#.NET 2005

## Ders Notları

**Doç. Dr. E. Şahin Çonkur**

**Erdem Yıldız**

Pamukkale Üniversitesi

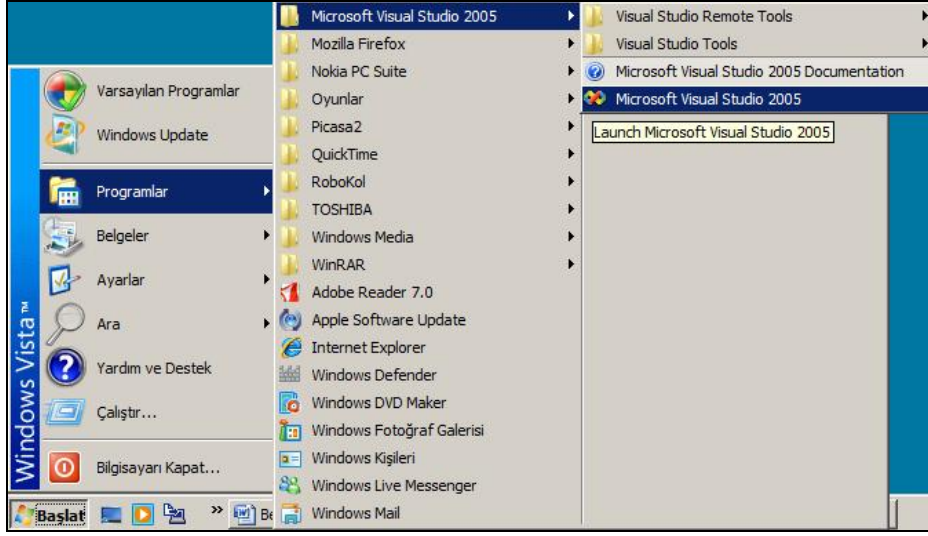
Mühendislik Fakültesi

Makine Mühendisliği Bölümü

Makine Teorisi ve Dinamiği Anabilim Dalı

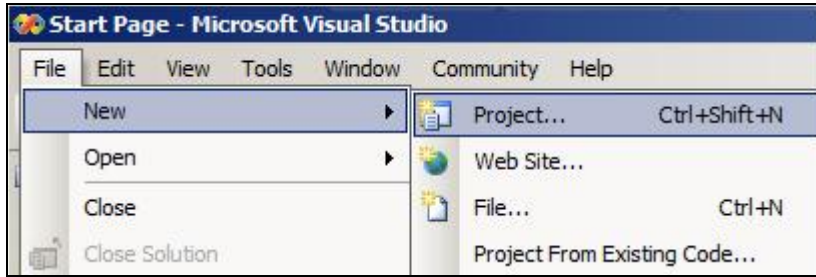
## Microsoft Visual Studio 2005 Programının Açılması

Şekil 1’de görüldüğü gibi **Başlat -> Programlar -> Microsoft Visual Studio 2005 -> Microsoft Visual Studio 2005** ile programı açabilirsiniz.



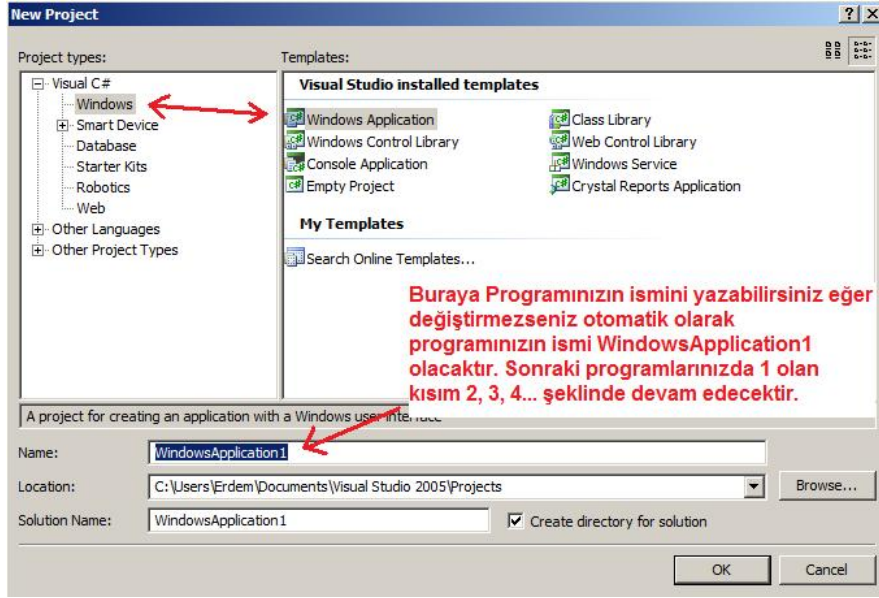
Şekil 1

Karşınıza **Microsoft Visual Studio 2005** programının **Start Page** ekranı gelecektir. Buradan Şekil 2’deki gibi **File -> New -> Project** ile **New Project** Penceresine ulaşacaksınız.



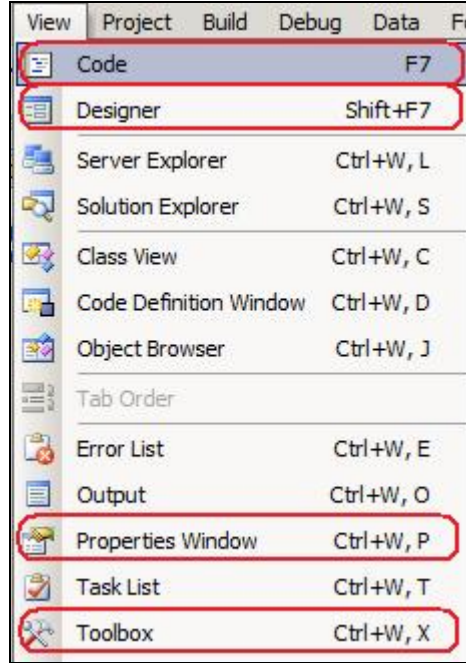
Şekil 2

**New Project** Penceresinde Şekil 3’teki gibi **Project Types**’ın altında **Visual C# -> Windows**’u seçtikten sonra **Templates**’in altında **Windows Application**’ı seçerek **OK** butonuna basarak programınızı açabilirsiniz.



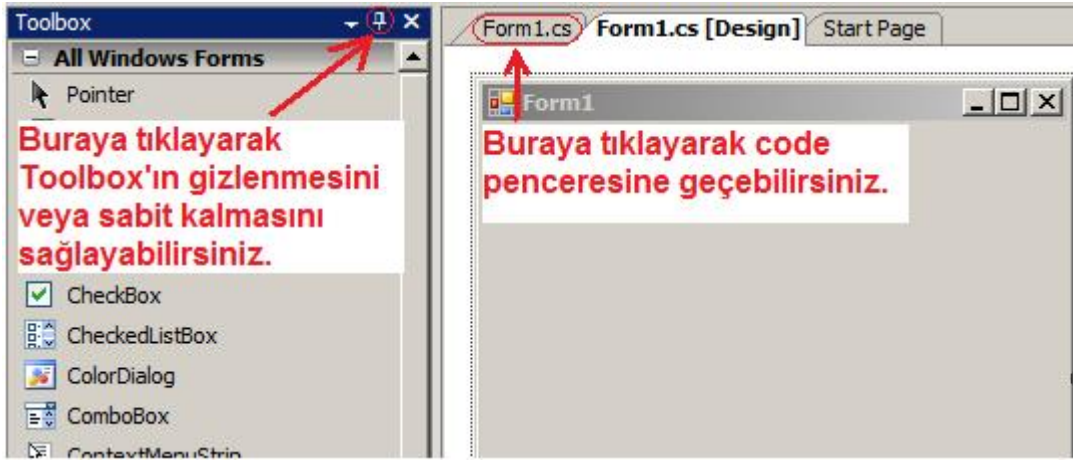
Şekil 3

Programınızın **Form1.cs [Design]** kısmına ulaşmış bulunmaktasınız. Programınızın kod penceresine Şekil 4'teki gibi **View->Code** ile ulaşabilirsiniz. Programınızın Design penceresine **View->Designer** ile ulaşabilirsiniz. Ayrıca programlarımızda kullanacağımız **ToolBox** ve **Properties Window**'a da **View**'den ulaşabilirsiniz.



Şekil 4

Programınızın **Design** penceresine **Toolbox**'tan istediğiniz kontrolleri sürükleyerek bırakabilirsiniz (Şekil 5). Burada dikkat edilecek bir nokta vardır. **Toolbox** penceresi **Form1.cs [Design]** penceresi aktif ise aktif olacaktır. Eğer aktif pencere **Form1.cs** veya başka bir pencere ise **Toolbox** penceresi silik olacak ve kullanılamayacaktır.



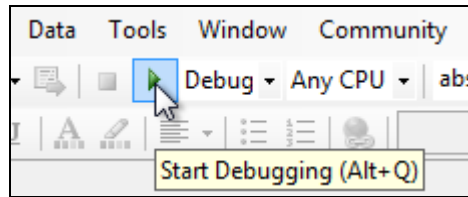
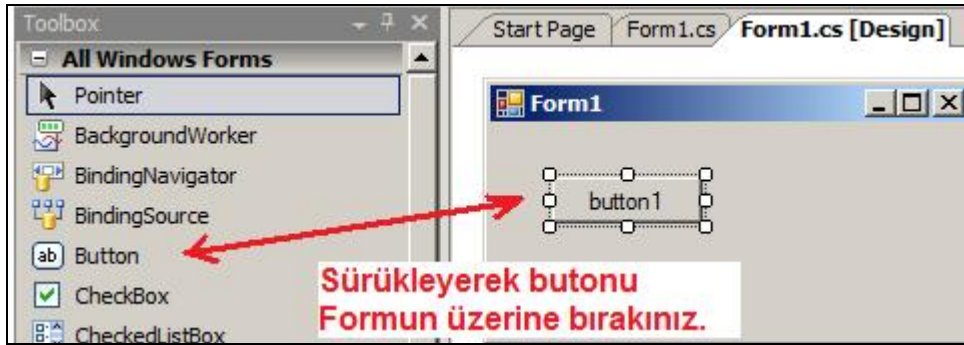
Şekil 5

## Örnek 1

Formunuzun **Form1.cs[Design]** penceresinin üzerine **Toolbox**'ta bulunan **Button**'u sürükleyerek bırakınız (Şekil 6). **button1**'e çift tıklayarak

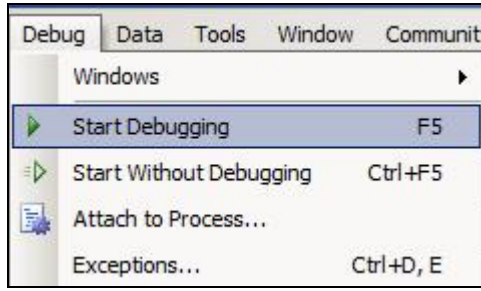
```
MessageBox.Show("Merhaba");
```

yazınız.



Şekil 6

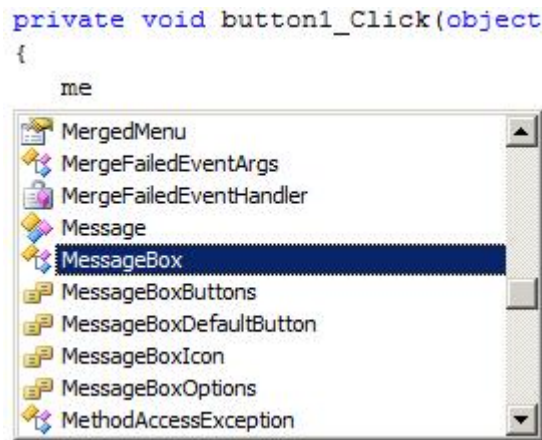
Programınızı **Debug -> Start Debugging** ile çalıştırınız (Şekil 7). Programı F5'e basarak ya da Şekil 6'nın alt kısmında görülen ikona tıklayarak da başlatabilirsiniz. Butona tıkladığınızda **Merhaba** yazan Mesaj kutusu açılacaktır. Program kodunuz Şekil 8'deki gibi olacaktır. Sadece **namespace** karşısında yazan yazı programı ilk açarken Şekil 3'de **Name** karşısına yazdığınız isim olacaktır.



Şekil 7

```
1 using System;
2 using System.Collections.Generic;
3 using System.ComponentModel;
4 using System.Data;
5 using System.Drawing;
6 using System.Text;
7 using System.Windows.Forms;
8
9 namespace makine1
10 {
11     public partial class Form1 : Form
12     {
13         public Form1 ()
14         {
15             InitializeComponent ();
16         }
17
18         private void button1_Click(object sender, EventArgs e)
19         {
20             MessageBox.Show ("Merhaba");
21         }
22     }
23 }
```

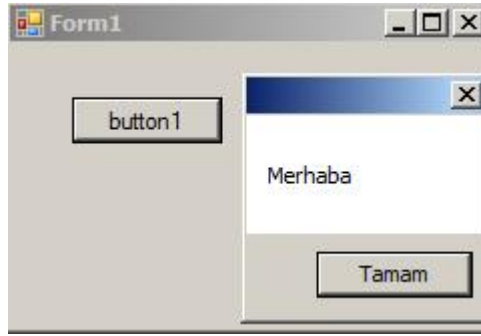
Şekil 8



Şekil 9

Visual C#.NET 2005 programının kod tamamlama özelliği vardır (Şekil 9). **MessageBox.Show("Merhaba")** yazmak istediğimizde yazının bir kısmını örneğin **me** yazarsak bir liste açılır ve yazdığımız kısma uygun olan fonksiyonlar belirir. İstedığımız fonksiyon seçili değilse **yukarı** ve **aşağı** oklarla fonksiyonu seçip **tab**

tuşuna basarak uzun uzun yazmaya gerek kalmadan **MessageBox** ifadesini seçebiliriz. Daha sonra nokta yazdığımızda yeni bir liste açılacaktır. Bu listeden benzer şekilde **Show** ifadesi seçilir. Kalan kısmı elle yazmak gerekmektedir. Program çalıştırıldığına ve düğmeye tıklandığında görüntü aşağıdaki gibi olacaktır.



Şekil - 55

## Örnek 2

Yeni bir **Form** açınız (Bkz. Şekil 2 ve Şekil 3). Formunuza iki tane **Buton** ekleyiniz. **Button1**'in üzerine çift tıklayarak aşağıdaki kodu ekleyiniz.

```
this.Text = "Merhaba";  
this.BackColor = Color.Blue;  
button2.Visible = true;  
button1.Visible = false;
```

Şekil 10

Button2'ye çift tıklayarak aşağıdaki kodu ekleyiniz.

```
this.Text = "Hoşgeldiniz";  
this.BackColor = Color.Red;  
button1.Visible = true;  
button2.Visible = false;
```

Şekil 11

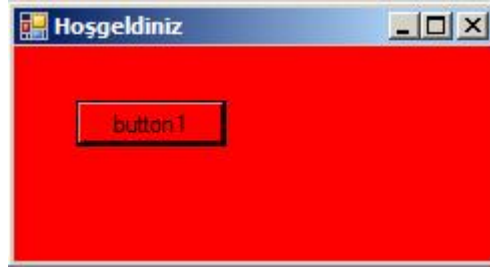
Burada

`this.Text = "Merhaba";` kodu Formun sol üst köşesine Merhaba yazdırmaktadır.

`this.BackColor = Color.Blue;` kodu Formun arka rengini mavi yapmaktadır.

`button2.Visible = true;` kodu button2'nin görünür olmasını sağlamaktadır.

Programınızı Debug -> Start Debugging ile çalıştırabilirsiniz(Şekil 7). Programdan bir görüntü (**button 2**'ye basıldıktan sonra) Şekil 12'deki gibi olacaktır.



Şekil 12

Ard arda iki tane bölme işareti `//` ile başlayan satırlar **açıklama** satırlarıdır. Bir satır açıklama satırı haline getirdiğinde rengi **yeşil** olur. **Açıklama satırları** özellikle daha sonra ne yazdığımızı anlamak veya başka biri bizim yazdığımız kodu değiştirmek istediğinde ona kodu açıklamak için kullanılır. Program **açıklama satırlarını** görmez. **Açıklama satırları** ayrıca bir kod grubunu geçici olarak devreden çıkarmak için kullanılır. **Form** kontrollerinin **BackColor**, **Visible** ve **Text** gibi özelliklerini kod ile değiştirebileceğiniz gibi **Properties Window**'dan da değiştirebilirsiniz.

### Örnek 3

**Formunuza** **ToolBox**'tan 2 tane **TextBox**, 1 tane **Button** ve 1 tane **Label** ekleyiniz. **Button**'un **Properties**'den **Text** özelliğine **Topla** yazınız (Şekil 13). **Button**'a çift tıklayarak Şekil 14'teki kodu yazınız. Programınızı **Debug->Start Debugging** ile çalıştırabilirsiniz (Şekil 7).

```
int a, b, c;
```

kodu 3 tane integer (tamsayı) tipinde değişken tanımlamanızı ( hafızadan bu değişkenler için yer ayırmamızı) sağlar. **TextBox**'ın içine yazılan yazı **string** tipinden olacağından (integer değil) **Convert.ToInt32** kodu ile **TextBox** içinde yazan yazıyı **integer** tipine dönüştürüp **a** veya **b** sayısına atama yapar.

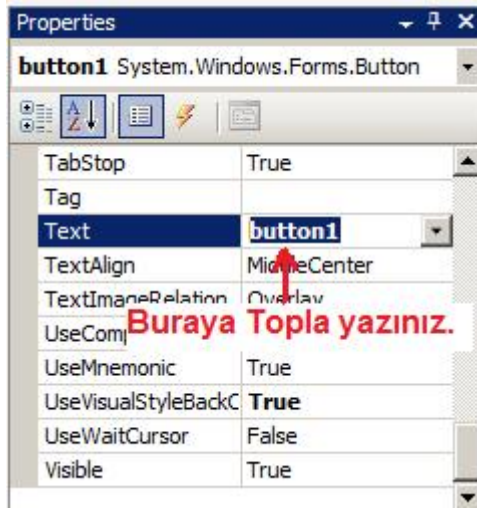
```
b = Convert.ToInt32(textBox2.Text);
```

Buradaki kod türkçe olarak ifade edilirse, "**textBox2.Text** içinde olan **string**'i **int**'e çevir ve **b**'nin değeri olarak ata" demektir. Buradaki **eşittir işareti** matematikteki **eşittir işaretiyle** aynı değildir. **Eşittir işareti**'nin buradaki anlamı **atama yapmak** demektir. Yani kod sağdan sola çalışır. Sağdaki değer soldaki değişkenin hafızada ayrılmış yerine o değişkenin değeri olarak atanır veya yazılır. **Convert.ToInt32(textBox2.Text)= b;** şeklinde bir ifade manasızdır.

Sonucu **label**'a yazdırırken **int** türündeki **c**'yi **string**'e çevirmeniz gerekiyor. Bunun kolay yolu **ToString()** metodunu kullanmaktır.

```
label1.Text = c.ToString();
```

Programınızı **TextBox**'ların içine tamsayı yazarak çalıştırırsanız hatasız bir şekilde programınız çalışır.



Şekil 13

```
int a, b, c;  
a = Convert.ToInt32(textBox1.Text);  
b = Convert.ToInt32(textBox2.Text);  
c = a + b;  
label1.Text = c.ToString();
```

Şekil 14

**TextBox**'ların içine ondalıklı sayı yazarsanız programınız hata verecektir, çünkü tanımladığınız değişkenler integer (tamsayı) tipindedir. **Button1\_Click** içindeki kodu (Şekil 14) silerek Şekil 15'teki kodu yazarsanız **TextBox**'ların içine tamsayı veya ondalıklı sayı girdiğinizde programınız hatasız çalışacaktır. **TextBox**'a yazdığınız ondalıklı sayı **virgül** ile ayrılmalıdır. Yani 3.1 değil 3,1 şeklinde olmalıdır. Fakat kod içine ondalıklı sayı yazdığımızda tam tersi geçerlidir. Yani 3,1 değil 3.1 şeklinde olmalıdır.

```
double a, b, c;  
a = Convert.ToDouble(textBox1.Text);  
b = Convert.ToDouble(textBox2.Text);  
c = a + b;  
label1.Text = c.ToString();
```

Şekil 15

**TextBox**'lara **integer** veya **double**'a çevirilemeyen **string** yazarsanız programınız hata verecektir. Örneğin **textBox1**'e a, **textBox2**'ye 10 yazınız ve çalıştırınız. Programınız hata verecektir. Eğer kod bloğunuzu **try-catch** içine yazarsanız bu hatayı yakalayıp kullanıcıya gösterebilirsiniz. **Button1\_Click** içindeki kodunuzu Şekil 16'daki gibi yapıp programınızı çalıştırınız. **Try** bloğu içinde bir hata oluştuğunda **catch** bloğunun içindeki kod çalışmaktadır. Aksi halde **catch** bloğunun içindeki kod çalışmamaktadır. Programınızdan birkaç görüntü Şekil 17'deki gibi olacaktır.

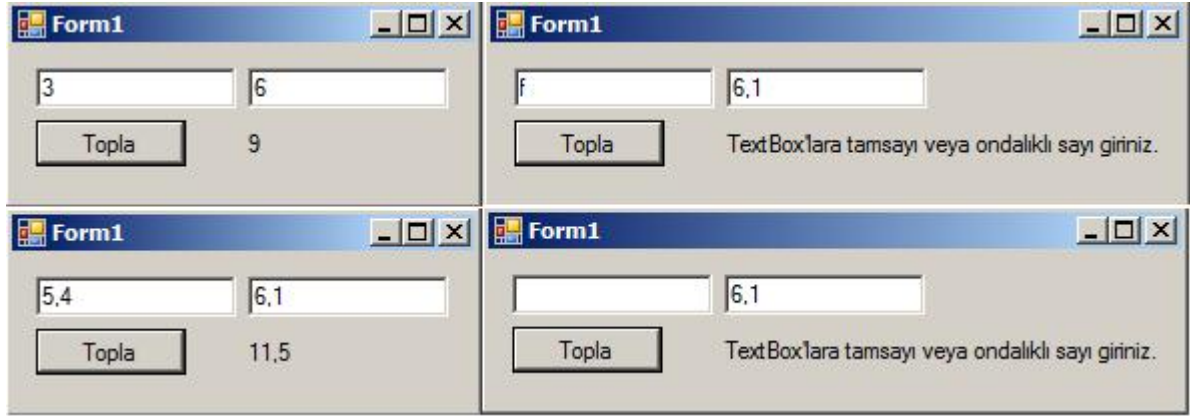


```

try
{
    double a, b, c;
    a = Convert.ToDouble(textBox1.Text);
    b = Convert.ToDouble(textBox2.Text);
    c = a + b;
    label1.Text = c.ToString();
}
catch
{
    label1.Text = "TextBox'lara tamsayı veya ondalıklı sayı giriniz.";
}

```

Şekil 16



Şekil 17

## Değişkenler ve Veri Tipleri

Değişkenler genellikle bir bilgiyi hafızada tutmak, değiştirmek ve gerektiğinde kullanmak için kullanılır.

### Değişken Tanımlama Kuralları

- Değişken ismi bir harf ile başlamalıdır.  
string Ad1; //Doğru  
string 1Ad; //Yanlış
- Değişken isminde boşluk bulunamaz.  
string Ad\_Soyad; //Doğru  
string Ad Soyad; //Yanlış
- Değişken isminde sadece harfler, rakamlar ve alt çizgi karakteri bulunabilir.  
string ali\_veli; //Doğru  
string ali+veli; //Yanlış
- Değişken ismi C# komutu olmamalıdır.  
string true; //Yanlış

## Veri Tipleri

Sayısal Tipler: **byte, sbyte, short, ushort, int, uint, long, ulong, float, double, decimal**

String Tipler: **string, char**

Mantıksal Tip: **bool**

Sayısal Tipler	Açıklama
<b>byte</b>	1 Baytlık işaretli tamsayı tipidir.
<b>sbyte</b>	1 Baytlık işaretli tamsayı tipidir.
<b>short</b>	2 Baytlık işaretli tamsayı tipidir.
<b>ushort</b>	2 Baytlık işaretli tamsayı tipidir.
<b>int</b>	4 Baytlık işaretli tamsayı tipidir.
<b>uint</b>	4 Baytlık işaretli tamsayı tipidir.
<b>long</b>	8 Baytlık işaretli tamsayı tipidir.
<b>ulong</b>	8 Baytlık işaretli tamsayı tipidir.
<b>float</b>	4 Baytlık ondalık sayı tipidir.
<b>double</b>	8 Baytlık ondalık sayı tipidir.
<b>decimal</b>	16 Baytlık bir veri tipidir.

Tablo 1

Sayısal Tipler	Açıklama
<b>int</b>	4 Baytlık işaretli tamsayı tipidir.
<b>float</b>	4 Baytlık ondalık sayı tipidir.
<b>double</b>	8 Baytlık ondalık sayı tipidir.

Tablo 2

Tablo1 genel olarak sayısal değişken tiplerini göstermektedir. Tablo 2 ise bizim en çok kullanacağımız tipleri göstermektedir. İşaretsiz tamsayı tipleri sadece pozitif sayıları tutabilmektedir. İşaretsiz tamsayı tipleri hem pozitif hem de negatif sayıları tutmaktadır. 1 Bayt'lık(8 Bit) bir tip  $2^8 = 256$  adet değer tutabilmektedir. 4 Bayt'lık(32 Bit) bir tip  $2^{32} = 4.294.967.296$  adet değer tutabilmektedir.

**bool** 2 Baytlık bir veri tipi olmasına rağmen sadece **true** veya **false** değerlerini alabilir.

**bool** cizim=**true**;

**char** iki baytlık bir karakter veri tipidir. Sadece 1 karakter tutabilir.

**char** harf='s';

String karakter sınırı verilmezse 2 milyar karaktere kadar atama yapılabilen sayısal olmayan veri tipidir.

**string** f= "Merhaba";

## Döngüler

### "for" Döngüsü

**for döngüsünün** yapısı aşağıdaki gibidir.

```
for(başlangıç işlemleri; devam şartları; ara işlemler)
{ komutlar; }
```

**for döngüsü** devam şartları sağlandığı sürece komutları işlemektedir. Başlangıç işlemleri sadece bir defa çalışmaktadır. Devam şartları **bool** (true veya false) türünden olmaktadır. Aşağıdaki örneği inceleyelim: 10

```
int N=20;

for(int i = 0; i <N; i++)
{
    this.Text=i.ToString();
}
```

Burada **int** tipinde bir değişken tanımlanmış ve bu değişkene 20 değeri atanmıştır. For içinde **int** tipinde **i** değişkeni tanımlanmış ve başlangıç değeri olarak 0 atanmıştır. Şu anda **i**'nin değeri 0'dır. **i**, **N** ile karşılaştırılır ve **i**, **N**'den küçük olduğu için döngü çalışmaya başlar ve bir sonraki döngü için **i++** ile **i**'nin değeri bir arttırılır. Döngü içine gelindiğinde **i**'nin değeri **Form**'un **Text** alanında yazdırılır. Sonuç olarak **i** **Form**'un **Text** alanında 0'dan 19'a kadar sayıp durur.

### “while” ve “do while” Döngüleri

```
while(şart)
{ komutlar; }

while (c <= a)
{
    b += c;
    c++;
}
```

**while** ve **do while**'da şart gerçekleştiği sürece(şart = true) komutlar işlenir.

```
do
{ komutlar;}
while(şart);
```

**do while**'da komutlar en az bir kere çalışır. **While**'da ise hiç çalışmayabilir.

```
do
{
    b += c;
    c++;
}
while (c <= a);
```

### “if” - “else” Yapısı;

```
if(şart)
{ komutlar;}
else { komutlar;}

if (a < 0)
{
    label1.Text = "Textbox'a pozitif bir sayı giriniz.";
}
else
{
    label1.Text = "";
}
```

**if - else** bloğunda else bloğu yazılmasa da **if** bloğu çalışmaktadır.

## Atama Operatörleri

<u>Operatör</u>	<u>Eşdeğeri</u>	<u>Anlamı</u>
x += y	x = x + y	x + y'nin değerini x'e ata
x -= y	x = x - y	
x *= y	x = x * y	
x++	x = x + 1	

<u>Operatör</u>	<u>Eşdeğeri</u>	
x--	x = x - 1	
x %= y	x = x % y	x'in y'ye bölümünden kalanı x'e ata

## Şart İfadeleri

==	Eşittir
!=	Eşit Değil
>	Büyük
<	Küçük
>=	Büyük Eşit
<=	Küçük Eşit
!	Değil
&&	Ve
	Veya

Örnek kullanım aşağıdaki gibidir.

```
if (a == 3)
{
    label1.Text = "TextBox'a 3 sayısını girdiniz.";
}
```

## Örnek 4

Bu örnekte  $a * x^2 + b * x + c = 0$  şeklindeki 2. Dereceden bir denklemin köklerini bulduran bir program yazacaksınız. Formunuza 5 adet TextBox ve 2 adet Button ekleyiniz. Button1'in Text özelliğini Kökleri bul ve Button2'nin Text özelliğini Temizle olarak değiştiriniz. Button1\_Click olayına aşağıdaki kodu yazınız. Button1\_Click olayına Button'a çift tıklayarak ulaşabilirsiniz.

```
try
{
    double a, b, c, delta, x1, x2, z;
    a = Convert.ToDouble(textBox1.Text); //textBox1'in içine yazılanı double'a çevirip a'ya atar.
    b = Convert.ToDouble(textBox2.Text);
    c = Convert.ToDouble(textBox3.Text);
    delta = b * b - 4 * a * c;
    if (delta < 0)
    {
        delta = Math.Abs(delta); //Math.Abs(delta) kodu, delta'nın mutlak değerini alır.
        z = Math.Sqrt(delta) / (2 * a); //Math.Sqrt(delta) kodu, delta'nın karekökünü alır.
    }
}
```

```

if (b == 0) //delta < 0 ve b == 0
{
    textBox4.Text = "+" + z.ToString() + "i";
    textBox5.Text = "-" + z.ToString() + "i";
}
else //delta < 0 ve b != 0
{
    textBox4.Text = Convert.ToString((-b / (2 * a)) + " + " + z.ToString() + "i";
    textBox5.Text = Convert.ToString((-b / (2 * a)) + " - " + z.ToString() + "i";
}
}
else //delta >= 0
{
    x1 = (-b + Math.Sqrt(delta)) / (2 * a);
    x2 = (-b - Math.Sqrt(delta)) / (2 * a);
    textBox4.Text = x1.ToString();
    textBox5.Text = x2.ToString();
}
}
catch
{
    MessageBox.Show("Hata oluştu.");
}

```

Form1\_Load olayına aşağıdaki kodu yazınız. Form1\_Load olayına Form'a çift tıklayarak ulaşabilirsiniz. Form1\_Load olayına yazılan kodlar program yüklenirken çalışacak kodlardır. "Load" kelimesinin Türkçesi "yüklemek" demektir.

```

textBox1.Text = "1";
textBox2.Text = "0";
textBox3.Text = "-4";
textBox4.Text = "birinci kök";
textBox5.Text = "ikinci kök";

```

Button2\_Click olayına aşağıdaki kodu yazınız.

```

textBox1.Text = "1";
textBox2.Text = "0";
textBox3.Text = "-4";
textBox4.Text = "birinci kök";
textBox5.Text = "ikinci kök";

```

## Math Sınıfı

Matematiksel işlemleri yapmamızda Math sınıfının özellikleri bize yardımcı olmaktadır. Örnek 4'te Math sınıfının 2 adet özelliğini görmüştünüz. Math sınıfının bazı özellikleri burada tanıtılacaktır.

`Math.E;` //e sayısını vermektedir.  $e \approx 2.71$

`Math.PI;` // $\pi$ (pi) sayısını vermektedir.  $\pi \approx 3.14$

`Math.Sin(b);` //b sayısının sinüsünü almaktadır, b sayısını radyan olarak kabul etmektedir. Radyandan dereceye geçmek isterseniz  $180 / \pi$  ile çarpmalısınız. Dereceden radyana geçmek için ise  $\pi / 180$  ile çarpmalısınız.

`Math.Cos(b);` //b sayısının kosinüsünü almaktadır.

`Math.Tan(b);` //b sayısının tanjantını almaktadır.

`Math.Exp(b);` //e<sup>b</sup> anlamındadır. b sayısını biz belirlemekteyiz.

`Math.Pow(b, c);` //b<sup>c</sup> anlamındadır. Örneğin b = 2, c = 3 ise sonuç  $2^3 = 8$ 'dir.

`Math.Sqrt(b);` //b sayısının karekökünü almaktadır.

`Math.Ceiling(b);` //b sayısını üst sayıya yuvarlar. Örneğin b = 10.3 ise sonuç 11'dir.

`Math.Floor(b);` //b sayısını alt sayıya yuvarlar. Örneğin b = 10.3 ise sonuç 10'dur.

`Math.Round(b);` //b sayısını yakın olan tamsayıya yuvarlar. Örneğin b = 10.3 ise sonuç 10'dur, b = 10.8 ise sonuç 11'dir.

`Math.Round(b, c);` //b sayısının virgülden sonra c sayısı kadar basamağı olacak şekilde yuvarlatılır. Örneğin b = 10.3867 ve c = 1 ise sonuç 10.4'tür.

`Math.Min(b, c);` //b ve c sayısından küçük olanını verir. Örneğin b = 3 ve c = 4 ise sonuç 3'tür.

`Math.Max(b, c);` //b ve c sayısından büyük olanını verir. Örneğin b = 3 ve c = 4 ise sonuç 4'tür.

`Math.Abs(b);` //b sayısının mutlak değerini almaktadır. Örneğin b = -3 ise sonuç 3'tür.

`Math.Log10(b);` //b sayısının 10 tabanında logaritmasını almaktadır. Örneğin b = 100 ise sonuç 2'dir.

`Math.Log(b);` //b sayısının ln'ini almaktadır.

`Math.Log(b, c);` //c tabanında b sayısının logaritmasını almaktadır. Örneğin b = 8 ve c = 2 ise sonuç 3'tür.

## Örnek 5

Bu örnekte 0'dan kullanıcının girdiği sayıya kadar olan sayıların toplamını hesaplayan ve kullanıcının girdiği sayının faktöriyelini hesaplayan bir program yazacağız. Bu örnekte for, while, do-while ve if-else yapılarını kullanacağız. Formunuza 1 adet TextBox, 1 adet Label ve 6 adet Button ekleyiniz. Button1\_Click olayına aşağıdaki kodu ekleyiniz.

```
try
{
    int a, b;
    b = 0;
    a = Convert.ToInt32(textBox1.Text);
    if (a < 0)
    {
        label1.Text = "Textbox'a pozitif bir sayı giriniz.";
    }
}
```

```

}
else //a >= 0
{
    for (int i = 0; i <= a; i++)
    {
        b += i;
    }
    label1.Text = b.ToString();
}
}
catch
{
    label1.Text = "Textbox'a tamsayı giriniz.";
}

```

Button1'in yaptığı işlemi farklı bir kodla yapalım. Button2'ye aşağıdaki kodu ekleyiniz.

```

try
{
    int a, b;
    b = 0;
    a = Convert.ToInt32(textBox1.Text);
    if (a < 0)
    {
        label1.Text = "Textbox'a pozitif bir sayı giriniz.";
    }
    else
    {
        for (int i = a; i >= 0; i--)
        {
            b += i;
        }
        label1.Text = b.ToString();
    }
}
catch
{
    label1.Text = "Textbox'a tamsayı giriniz.";
}

```

Button1 ve Button2'deki kodlar arasındaki farkı bularak nasıl çalıştıklarını gözlemleyiniz. Button3'e yazacağınız kod faktöriyel hesaplamayı sağlayacaktır. Button3'e aşağıdaki kodu yazınız.

```

try
{
    int a, b;
    b = 1;
    a = Convert.ToInt32(textBox1.Text);
    if (a < 0)
    {
        label1.Text = "Textbox'a pozitif bir sayı giriniz.";
    }
}

```

```

    }
    else
    {
        for (int i = 1; i <= a; i++)
        {
            b *= i;
        }
        label1.Text = b.ToString();
    }
}
catch
{
    label1.Text = "Textbox'a tamsayı giriniz.";
}

```

Button4'e yazacağınız kod, kullanıcının girdiği sayı eğer 0 ile 9 arasında ise sayının faktöriyelini alır. Eğer sayı 9'dan büyük ise 0'dan girilen sayıya kadar olan sayıların toplamını hesaplar.

```

try
{
    int a, b, c;
    b = 1;
    c = 0;
    a = Convert.ToInt32(textBox1.Text);
    if (a < 0)
    {
        label1.Text = "Textbox'a pozitif bir sayı giriniz.";
    }
    else if (a >= 0 && a < 10)
    {
        for (int i = 1; i <= a; i++)
        {
            b *= i;
        }
        label1.Text = b.ToString();
    }
    else
    {
        for (int i = a; i >= 0; i--)
        {
            c += i;
        }
        label1.Text = c.ToString();
    }
}
catch
{
    label1.Text = "Textbox'a tamsayı giriniz.";
}

```



Button1'e yazdığınız kodun yaptığı işi while döngüsü ile de yapabiliriz. Button1 ile Button5'in yaptığı iş aynı olacaktır. Button5'e aşağıdaki kodu yazınız.

```
try
{
    int a, b, c;
    b = 0;
    c = 0;
    a = Convert.ToInt32(textBox1.Text);
    if (a < 0)
    {
        label1.Text = "Textbox'a pozitif bir sayı giriniz.";
    }
    else
    {
        while (c <= a)
        {
            b += c;
            c++;
        }
        label1.Text = b.ToString();
    }
}
catch
{
    label1.Text = "Textbox'a tamsayı giriniz.";
}
```

Button1, button2, button5 ve button6 aynı işi yapmaktadır. Button6'da do-while döngüsü kullanılmaktadır. Button6'ya aşağıdaki kodu yazınız.

```
try
{
    int a, b, c;
    b = 0;
    c = 0;
    a = Convert.ToInt32(textBox1.Text);
    if (a < 0)
    {
        label1.Text = "Textbox'a pozitif bir sayı giriniz.";
    }
    else
    {
        do
        {
            b += c;
            c++;
        }
        while (c <= a);
        label1.Text = b.ToString();
    }
}
```

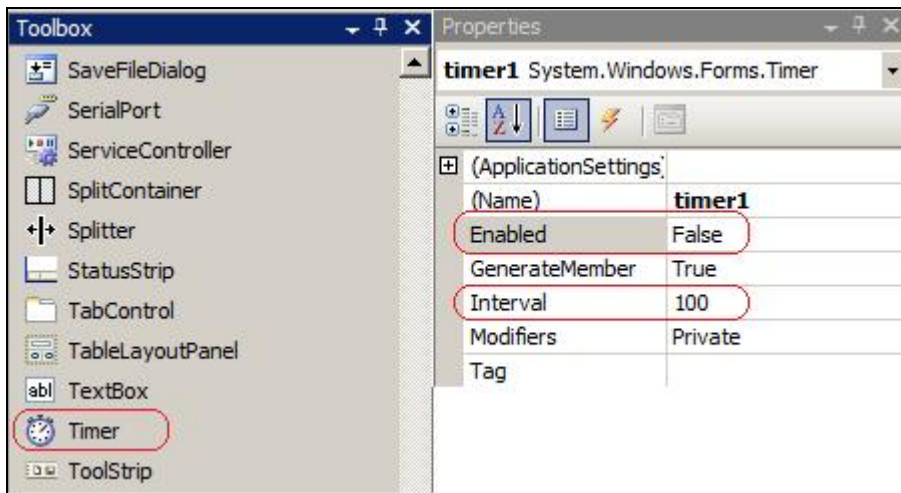
```

}
catch
{
    label1.Text = "Textbox'a tamsayı giriniz.";
}

```

## TIMER

Timer Windows Form kontrollerindedir. Formumuza timer yerleştirmek istediğimizde Toolbox'tan formumuzun design penceresine sürükleyerek bırakabiliriz. Timer'ın 2 adet önemli özelliği vardır. Timer'ın Enabled özelliği true veya false yapılabilir. False yapılırsa timer'ımızın çalışması durur, True yapılırsa timer'ımız çalışmaya başlar veya çalışıyorsa çalışmaya devam eder. Timer'ın Interval özelliği timer'ımızın ne kadar zamanda bir çalışacağını ayarlamaktadır. Interval özelliğine yazılan değer milisaniye cinsinden olmaktadır. Timer'ın özelliklerinde değişiklik yapmazsanız Enabled özelliği False ve Interval özelliği 100 olarak ayarlanmıştır.



Şekil 18

## Örnek 6

Formunuza 1 adet Timer ve 1 adet Button yerleştiriniz. Global(genel) değişken olarak k tanımlayınız ve ilk değerini 0(sıfır) olarak ayarlayınız.

```

public partial class Form1 : Form
{
    int k = 0; //global değişken

```

Timer'a çift tıklayarak aşağıdaki kodu yazınız.

```

k += 1; //k = k + 1 anlamındadır.
this.Text = String.Format("k = {0}", k); //{0} olarak belirtilen yere k'nın değeri gelmektedir. Formun sol üst köşesine k'nın değerini yazdırmaktadır.
if (k > 200)
{
    this.Width = k; //Formun genişliği = k
    this.Height = k; //Formun yüksekliği = k

```

```
}
```

Button'a çift tıklayarak aşağıdaki kodu yazınız.

timer1.Enabled = !timer1.Enabled; //timer1.Enabled özelliğinin deęilini alıyor. True ise False, False ise True yapıyor.

```
if (timer1.Enabled == true)
{
    button1.Text = "durdur.";
}
else
{
    button1.Text = "çalıştır.";
}
```

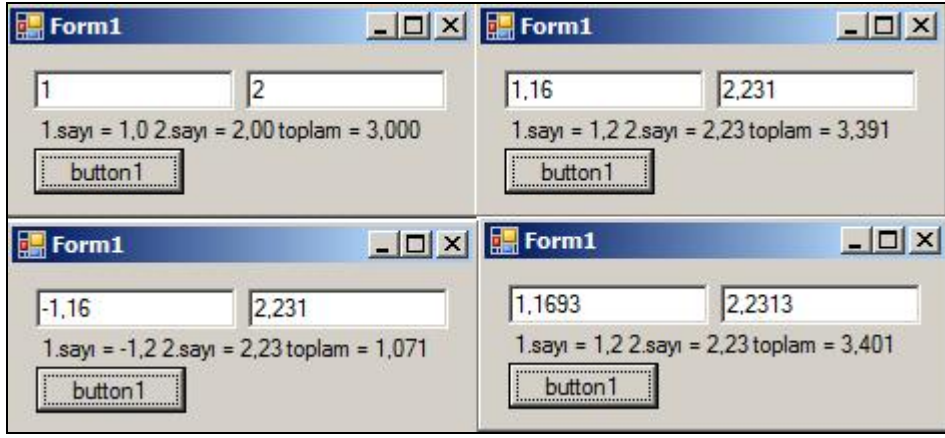
Timer'ın properties penceresinden Enabled özelliğini True ve Interval özelliğini 5 yapınız. Programınızı çalıştırınız.

## Örnek 7

Bu programımızın yaptığı işi Örnek 3'te yapmaktadır fakat burada farklı bir gösterim tarzı kullanılacaktır. Formunuza 1 adet Button, 1 adet label, 2 adet TextBox yerleştiriniz. Button'un Click olayına aşağıdaki kodu yazınız.

```
double m, n, toplam;
m = Convert.ToDouble(textBox1.Text);
n = Convert.ToDouble(textBox2.Text);
toplam = m + n;
label1.Text = String.Format("1.sayı = {0:f1} 2.sayı = {1:f2} toplam = {2:f3}",m, n, toplam);
```

Burada ("1.sayı = {0:f1} şeklinde devam eden kodda 0(sıfır) m'yi temsil etmektedir ve f1 virgülden sonra kaç basamağın gözükeceğini belirtmektedir. Aynı şekilde 2.sayı = {1:f2} şeklindeki kodda 1 n'yi temsil etmektedir ve f2 virgülden sonra 2 basamağın gözükeceğini belirtmektedir. Örneğin n'nin değeri 12,12323 ise kullanıcı 12,12 şeklinde görmektedir. N ile işlem yapılırken 12,12323 değeri alınmaktadır. Programınızdan birkaç görüntü aşağıdaki gibi olacaktır.



Şekil 19

## Diziler

Diziler de değişkenler gibi kullanılmaktadır. Değişkenler de sadece bir adet değer tutulurken dizilerde birden fazla değer tutulabilir. Örneğin 100 adet değeri tutmak için 100 adet değişken tanımlamak yerine 100 elemanlı bir dizi tanımlamak daha pratiktir.

Double türünden 1 boyutlu, 100 elemanlı ve sayi isimli bir dizi aşağıdaki gibi tanımlanır.

```
double[] sayi = new double[100];
```

Dizinin Türü      1 Boyutlu dizi olduğunu gösteriyor.

Dizinin 100 elemanlı olduğunu gösteriyor.

## Örnek 8

Formunuza 2 adet Button, 1 adet TextBox ve 1 adet ListBox ekleyiniz. Global(genel) değişkenleri aşağıdaki gibi tanımlayınız.

```
public partial class Form1 : Form
{
    double[] sayi = new double[1000]; //1000 adet dizi için hafızadan yer ayırır.
    int k = 0; //global(genel) değişken
```

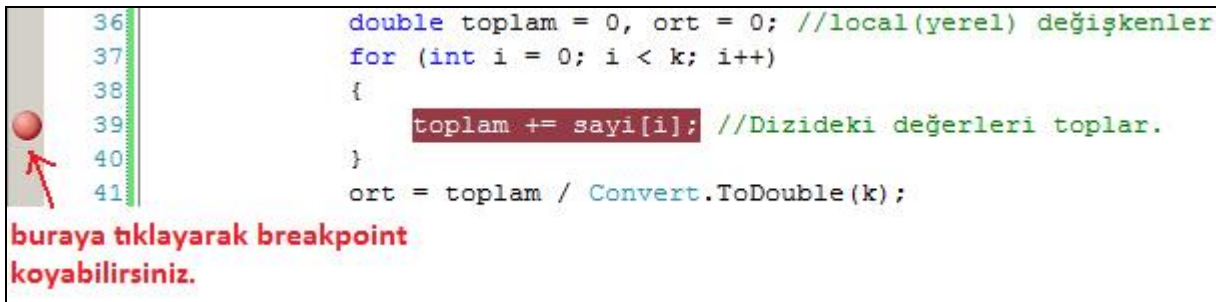
Button1'e tıklayarak aşağıdaki kodu yazınız.

```
if (textBox1.Text == "") //Eğer textBox1'in içi boş ise
{
    MessageBox.Show("bir sayı girmelisiniz!");
    return; //button1_Click olayından çık
}
sayi[k] = Convert.ToDouble(textBox1.Text); //textBox1'deki değeri sayi[k]'ya ata.
listBox1.Items.Add((k + 1).ToString() + ". sayı = " + sayi[k].ToString()); //listBox1'e sayi[k]'yı ve (k+1)'i ekle.
k += 1; //k'nın değerini 1 arttır.
textBox1.Text = ""; //textBox1'in içeriğini boşalt.
textBox1.Focus(); //İmleci textBox1'in içine getir.
```

Button2'ye tıklayarak aşağıdaki kodu yazınız.

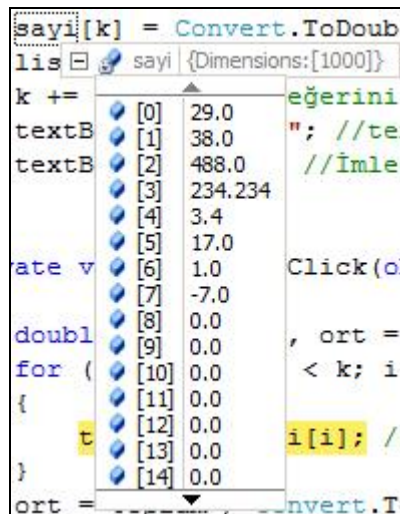
```
double toplam = 0, ort = 0; //local(yerel) değişkenler
for (int i = 0; i < k; i++)
{
    toplam += sayi[i]; //Dizideki değerleri toplar.
}
ort = toplam / Convert.ToDouble(k);
this.Text = toplam.ToString();
this.Text = String.Format("toplam = {0} ve ortalama = {1}", toplam, ort);
```

Programınıza Breakpoint ekleyerek değişkenlerinizin o anki değerlerini öğrenebilirsiniz. Programınız breakpoint'i koyduğunuz yere geldiğinde duracaktır. Mouse'unuz ile değişkenlerinizin üzerine gelerek o anki değerlerini öğrenebilirsiniz.



Şekil 20

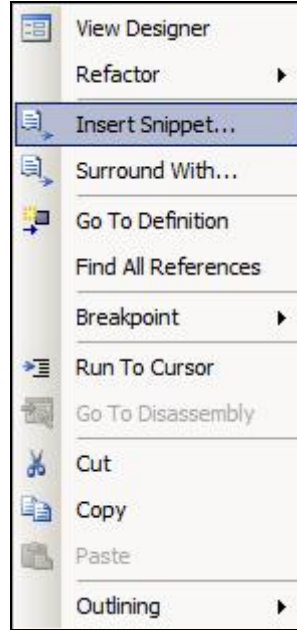
Şekil 20'deki yere Breakpoint koyarsanız Button2'ye tıkladığınız sürece program çalışır. TextBox'a birkaç değer girerek bu değerleri ListBox'ınıza atınız. Button2'ye tıklayarak programınızın durmasını sağlayınız ve mouse'unuz ile sayi isimli dizinizin üzerine gelerek dizinizde bulunan değerleri görebilirsiniz(Şekil 21).



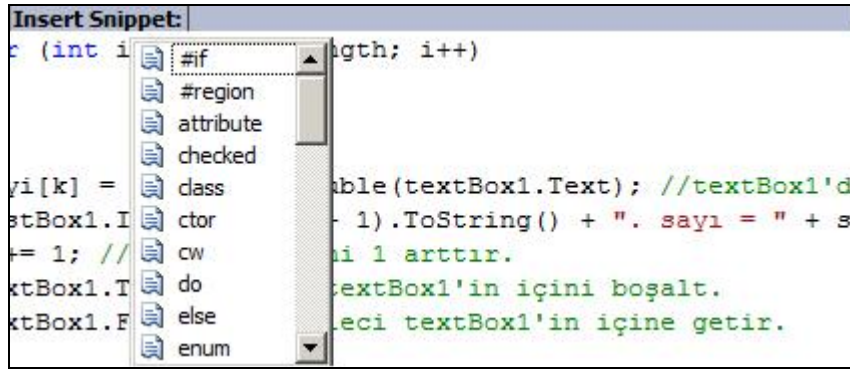
Şekil 21

Şekil 21'de gördüğümüz gibi dizinizdeki elemanların indis numaraları 0(sıfır) ile başlamaktadır.

Programınızda for, if, while gibi ifadelerin kullanım şekillerini kolay bir yolla öğrenebilirsiniz. Mouse'unuzun sağ tuşuna basarak Insert Snippet...'ı seçiniz buradan bir liste çıkacaktır. Bu listeden kullanmak istediğiniz yapıyı seçebilirsiniz(Şekil 22, 23).



Şekil 22



Şekil 23

## Fonksiyonlar

Fonksiyonlar programlarımızda daha az kod yazmamıza yardımcı olmaktadır. Bir işlem birden fazla yapılacaksa o işlemi fonksiyon olarak yazmamız daha uygundur. Örneğin 20 satırlık bir kodun aynısını, kodun başka yerlerinde 4 defa yazmak istersek toplamda 80 satırlık bir kod yazmamız gerekecektir. Ancak bu 20 satırlık kodu fonksiyon olarak yazarsak tek satır yazarak bu fonksiyonumuzu çağırabiliriz.

## Örnek 9

Formunuza 3 adet TextBox ve 3 adet Button yerleştiriniz. Global değişkenlerinizi aşağıdaki gibi tanımlayınız.

```
public partial class Form1 : Form
{
    double a, b, alan;//global(genel) değişken
```

3 adet Fonsiyonunuzu aşağıdaki gibi tanımlayınız.

```
private double dikdortgeninAlani(double kenar1, double kenar2)//kenar1 = a ve kenar2 = b
{
    return kenar1 * kenar2; //kenar1 * kenar2 değerini geriye döndür
}
private double daireninAlani(double yaricap) //yaricap = a
{
    return Math.PI * yaricap * yaricap; //Math.PI * yaricap * yaricap değerini geriye döndür
}
private void mb(String s) //s = "Merhaba"
{
    MessageBox.Show(s);
}
```

Button1'e çift tıklayarak aşağıdaki kodu yazınız.

```
a = Convert.ToDouble(textBox1.Text); //textBox1'deki değeri a'ya ata.
b = Convert.ToDouble(textBox2.Text); //textBox2'deki değeri b'ye ata.
alan = dikdortgeninAlani(a, b); //alan = kenar1 * kenar2
this.Text = alan.ToString();
```

Button2'ye çift tıklayarak aşağıdaki kodu yazınız.

```
a = Convert.ToDouble(textBox3.Text);
alan = daireninAlani(a); //alan = Mat.PI * yaricap * yaricap
this.Text = alan.ToString();
```

Button3'e çift tıklayarak aşağıdaki kodu yazınız.

```
mb("Merhaba");
```

Programınızı çalıştırabilirsiniz. Button1\_Click olayını siz yazmıyordunuz Button1'e çift tıklayınca kendiliğinden

```
private void button1_Click(object sender, EventArgs e)
{
}
kodu yazılıyordu. Şimdi ise Fonsiyon yazarken
private double daireninAlani(double yaricap)
{
}
```

şeklindeki kodu sizin yazmanız gerekmektedir. Yukarıdaki 3 adet Fonsiyondan 2'si benzer fakat Button3 ile çağırılan fonksiyon farklıdır. Button3 ile çağırılan fonksiyon geriye değer döndürmemektedir. Button1

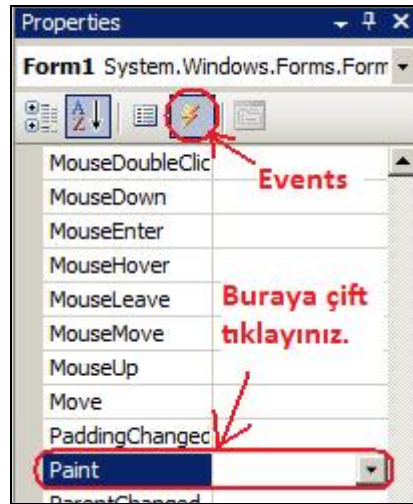
ve Button2 ile çağırılan fonksiyonlarda ise **return** komutu ile geriye değer döndürülmektedir. **private void** ile başlayan fonksiyonlar geriye değer döndürmemektedir. **private double** ile başlayan fonksiyonlar ise **return** kodu ile geriye değer döndürmektedir.

## Paint

Paint olayı çizimlerimizi yapmamız için bize yardımcı olacaktır. Daire, kare, dikdörtgen, doğru vb. çizimleri paint olayı içinde yapabiliriz. Form'unuzun Form1.cs[Design] penceresi aktif iken(Şekil 24) Properties penceresine gelip event ikonuna tıklarsanız(Şekil 25) Form1'e ait olayları görüntüleyebilirsiniz. Bu olaylardan Paint'e çift tıklarsanız Form1\_Paint olayına ulaşırsınız.



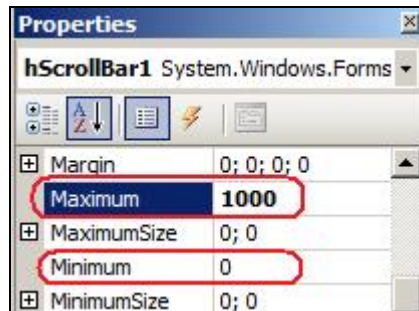
Şekil 24



Şekil 25

## Örnek 10

Formunuza 1 adet Timer, 1 adet Button, 2 adet Label ve 2 adet HScrollBar ekleyiniz. HScrollBar'ı Toolbox'tan yerleştirebilirsiniz. HScrollBar'larınızın minimum ve maximum değerlerini ayarlayabilirsiniz(Şekil 26). HScrollBar'larınızın maximum değerini 1000 olarak değiştirin ve minimum değerini 0 olarak bırakınız.



Şekil 26

HScrollBar'nızın değeri sağa doğru sürüklerken artmaktadır ve sola doğru sürüklerken azalmaktadır. Genel değişkenlerinizi aşağıdaki gibi tanımlayınız.



```
public partial class Form1 : Form
{
    int r = 50;
    int k = 1;
    int x, y;
```

Daire çizdirmek için daireCiz isimli fonksiyonunuzu aşağıdaki gibi tanımlayınız.

```
private void daireCiz(int r, PaintEventArgs e)
{
    Pen p = new Pen(Color.Green, k);
    e.Graphics.DrawEllipse(p, -r, -r, 2 * r, 2 * r); //Elipsimizi sınırlayan dikdörtgenimizin sol üst
köşesinin koordinatını(-r,-r) yapıyor. Bu dikdörtgenin genişliğini(Weight) 2*r, uzunluğunu(Height) 2*r
olarak ayarlıyor.
}
```

Form1\_Paint olayına aşağıdaki kodu yazınız.

```
e.Graphics.TranslateTransform(this.Width/2, this.Height/2); //Formun ortasını (0,0) noktası
yapmaktadır.
e.Graphics.DrawLine(Pens.Blue, 0, 0, 200, 0); //(0,0) noktası ile (200,0) noktası arasına mavi renkte
bir çizgi çizmektedir.
e.Graphics.DrawLine(Pens.Blue, 0, 0, 0, 200);
Pen p = new Pen(Color.Green, k); //Yeşil renkte, k kalınlığında bir kalem tanımlanmaktadır.
e.Graphics.DrawLine(Pens.Black, 0, 0, x, y);
daireCiz(r, e);
```

button1\_Click olayına aşağıdaki kodu yazınız.

```
timer1.Enabled = !timer1.Enabled; //Timer1 çalışıyorsa durduruyor, duruyorsa çalıştırıyor.
```

hScrollBar1'in üzerine çift tıklayarak aşağıdaki kodu yazınız.

```
label1.Text = hScrollBar1.Value.ToString(); //hScrollBar1'deki değeri label1'de gösteriyor.
r = hScrollBar1.Value; //hScrollBar1'in değerini r'ye atıyor.
Invalidate(); //Paint olayını çağırıyor.
```

hScrollBar2'nin üzerine çift tıklayarak aşağıdaki kodu yazınız.

```
label2.Text = hScrollBar2.Value.ToString();
k = hScrollBar2.Value;
Invalidate();
```

Form1\_Load olayına aşağıdaki kodu yazınız.

```
timer1.Interval = 1; //timer1_Tick içine yazılan kodlar 1ms aralıklarla tekrarlanıyor.
this.SetStyle(ControlStyles.AllPaintingInWmPaint, true); //Çizimdeki titremeyi önüyor.
this.SetStyle(ControlStyles.OptimizedDoubleBuffer, true); //Çizimdeki titremeyi önüyor.
```

Timer'a çift tıklayarak aşağıdaki kodu yazınız.

```

r += 1;
k += 1;
if (r >= 200) r = 0;
if (k >= 20) k = 0;
Invalidate();

```

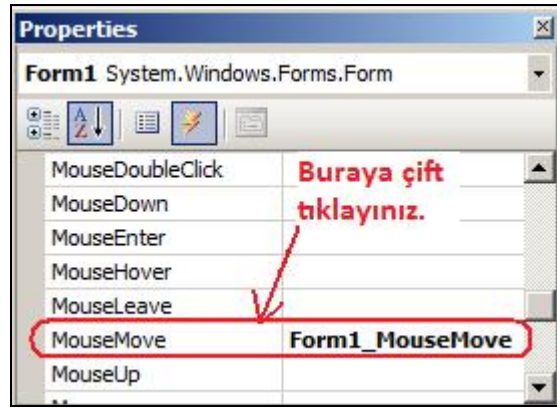
Form1\_MouseMove olayına aşağıdaki kodu yazınız.

```

x = e.X; //Mouse işaretçisinin x koordinatını x değişkenine atıyor.
y = e.Y; //Mouse işaretçisinin y koordinatını y değişkenine atıyor.
this.Text = String.Format("x = {0} y = {1}",x, y);
Invalidate();

```

Form1\_MouseMove olayını Form1\_Paint olayını bulduğunuz gibi çıkarabilirsiniz(Şekil 27).



Şekil 27

Formunuzun sol üst köşesi normalde (0,0) noktasıdır fakat

```
e.Graphics.TranslateTransform(a, b);
```

kodu ile yeni (0,0) noktası (a, b) noktası olmaktadır. Formunuzda soldan sağa doğru giderken x değeri artmaktadır ve yukarıdan aşağıya doğru giderken de y değeri artmaktadır. Doğru çizirken aşağıdaki kodu kullanmaktayız. Doğru çizmeyi iki noktayı birleştirerek yapmaktayız.

```
e.Graphics.DrawLine(Pens.Blue, 0, 0, 0, 200);
```

1. noktanın x koordinatı ← 2. noktanın y koordinatı  
1. noktanın y koordinatı ← 2. noktanın x koordinatı

Aşağıda sırasıyla dikdörtgen(içi boş), çember(içi boş), içi belirtilen renkle dolu bir dikdörtgen ve daire için gerekli olan kodlar yazılmıştır. 4 çizimde de siyah renk kullanılmıştır. DrawEllipse yazan kodda ilk değer elipsimizi sınırlayan dikdörtgenin sol üst köşesinin x koordinatını, ikinci değer elipsimizi sınırlayan dikdörtgenin sol üst köşesinin y koordinatını, üçüncü değer elipsimizi sınırlayan dikdörtgenin genişliğini, dördüncü değer elipsimizi sınırlayan dikdörtgenin uzunluğunu vermektedir.

```
e.Graphics.DrawRectangle(Pens.Black, 10, 20, 30, 40);
```

Sol üst köşenin x koordinatı ← yükseklik  
Sol üst köşenin y koordinatı ← genişlik

```
e.Graphics.DrawEllipse(Pens.Black, 10, 20, 30, 40);
```

```
e.Graphics.FillRectangle(Brushes.Black, 10, 20, 30, 40);  
e.Graphics.FillEllipse(Brushes.Black, 10, 20, 30, 40);
```

## DateTime

DateTime sınıfı tarih ve saat ile ilgili işlemlerinizi yapmanızı sağlar. DateTime sınıfı ile ilgili birkaç kod aşağıdaki gibidir.

```
DateTime t; //DateTime türünden bir t değişkeni tanımlanır.  
int a;  
a = DateTime.Now.Hour; //Şu andaki saat değerini a değişkenine atar.  
a = DateTime.Now.Minute; //Şu andaki dakika değerini a değişkenine atar.  
a = DateTime.Now.Second; //Şu andaki saniye değerini a değişkenine atar.  
t = DateTime.Now; //Şu andaki zamanı t değişkenine atar.  
t.ToLongTimeString(); //t değişkenindeki zamanı saat:dakika:saniye şekline çevirir.
```

## ArrayList

ArrayList sınıfı ile dizi tanımlaması yapabiliriz. ArrayList sınıfı ile tanımlanan dizilerin daha önce bahsettiğimiz dizilerden farkı boyutlarının belirtilmemesidir. ArrayList sınıfından dizi kullanmak istersek programımıza aşağıdaki satırı eklememiz gerekmektedir.

```
using System.Collections; //Diğer using ile başlayan satırların olduğu yere yazınız.
```

ArrayList sınıfından bir dizi tanımlaması aşağıdaki gibi yapılmaktadır.

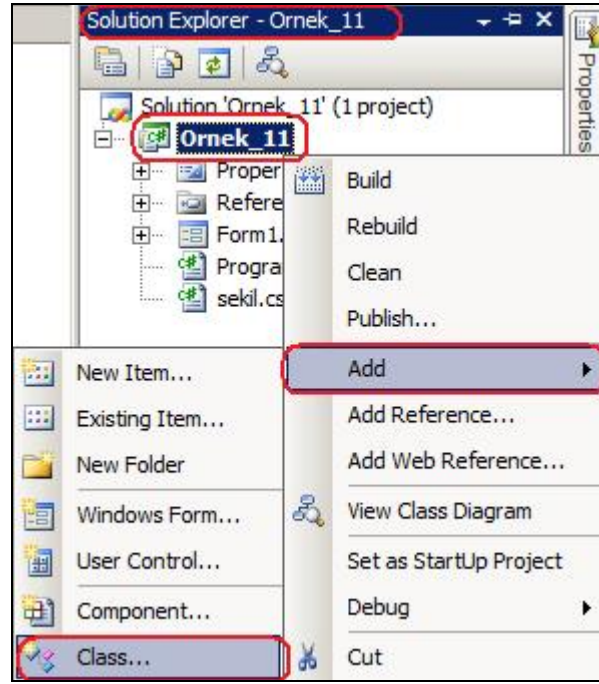
```
ArrayList ar1 = new ArrayList();  
└──────────┬───────────> Dizinin adı
```

ArrayList sınıfından tanımlanan dizilerin bazı özellikleri aşağıdaki gibidir.

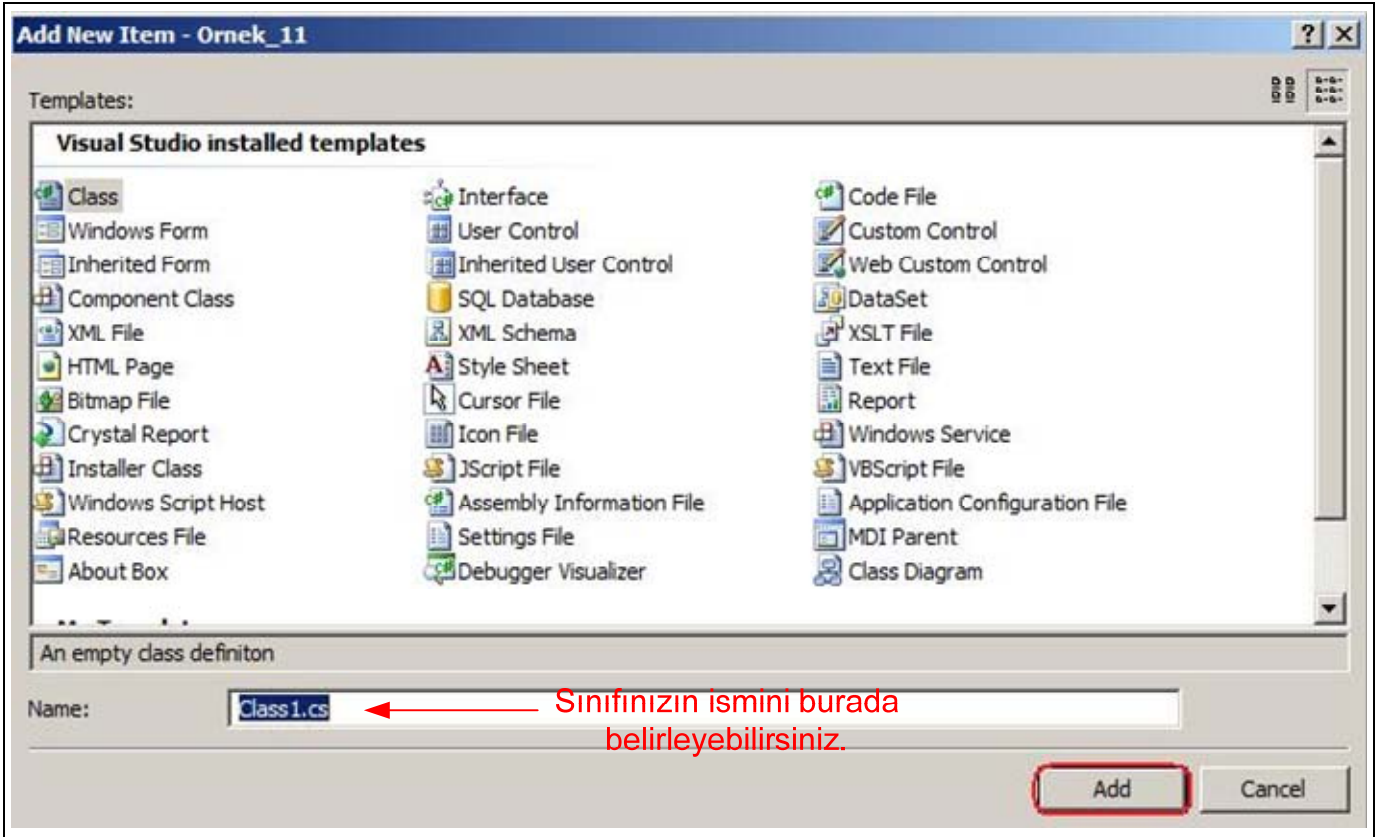
```
ar1.Clear(); //ar1 dizisindeki tüm elemanları siler.  
ar1.Count; //ar1 dizisindeki eleman sayısını verir.  
ar1.RemoveAt(3); //İndis numarası 3 olan elemanı ar1 dizisinden siler. Dizilerin indis numaraları 0'dan başlanmaktadır. 3 yerine silmek istediğiniz elemanın indis numarasını verebilirsiniz.
```

## Projenize yeni bir Form veya Sınıf eklemek

Projenize yeni bir sınıf eklemek için **View -> Solution Explorer** ile **Solution Explorer**'ı açtıktan sonra Projenizin isminin(Şekil 28'de Ornek\_11) üzerinde mouse'unuzun sağ tuşuna tıklayınız(Şekil 28). Açılan pencerede **Add -> Class...** seçeneklerine tıklarsanız karşınıza bir dialog penceresi gelecektir. Bu pencerede **Name** yazan kısma oluşturacağınız Sınıfın ismini yazabilirsiniz. Sınıfınızın ismini belirledikten sonra **Add** butonuna tıklayarak yeni bir sınıf ekleyebilirsiniz(Şekil 29).



Şekil 28



Şekil 29

Projenize yeni bir form eklemek için yapmanız gerekenler, projenize yeni bir sınıf eklemek için yaptıklarınıza çok benzemektedir. Projenize Sınıf eklemek için **Add -> Class...**'a tıklarken, Form eklemek için **Add -> Windows Form...**'a tıklamanız gerekmektedir.

## Örnek 11

Formunuza 3 adet Button ve 1 adet ColorDialog ekleyiniz. Button1'in Text özelliğine **Renk değiştir**, button2'nin Text özelliğine **Hepsini Sil** ve button3'ün Text özelliğine **Geri Al** yazınız. Programınıza **sekil** isimli bir sınıf ekleyiniz(Şekil 28-29). Sınıfınıza using ile başlayan satırların olduğu yere aşağıdaki kodu ekleyiniz.

```
using System.Drawing; //Color sınıfının gözükmesi için eklendi.
```

Sınıfınıza aşağıdaki değişken tanımlamalarını ekleyiniz.

```
class sekil
{
    //sekil sınıfının değişkenleri
    public int x1, y1, x2, y2;
    public Color k;
}
```

Form'unuzun kod sayfasında using ile başlayan satırların olduğu yere aşağıdaki kodu ekleyiniz.

```
using System.Collections; //ArrayList'in gözükmesi için eklendi.
```

Genel değişkenlerinizi aşağıdaki gibi tanımlayınız.

```
public partial class Form1 : Form
{
    int x1, y1, x2, y2;
    ArrayList ar = new ArrayList();
    Color k = Color.Green; //Color sınıfından bir değişken tanımlandı. Değişkenin ismi k ve ilk rengi yeşil'dir.
```

Form1\_MouseMove olayına aşağıdaki kodları yazınız.

```
//Mouse hareket ettikçe buradaki kodlar çalışır.
if (e.Button == MouseButton.Left) //Eğer sol mouse butonu kullanılıyorsa.
{
    x2 = e.X; //Mouse işaretçisinin x koordinatını x2'ye ata.
    y2 = e.Y;
    Invalidate();
}
```

Form1\_MouseUp olayına aşağıdaki kodları yazınız.

```
//Tıkladığınız mouse butonundan parmağınızı kaldırdığınızda buradaki kodlar çalışır.
if (e.Button == MouseButton.Left) //Eğer sol mouse butonu kullanılıyorsa.
{
    x2 = e.X;
    y2 = e.Y;
    Invalidate();
    sekil p = new sekil();
```

```
p.x1 = x1;
p.x2 = x2;
p.y1 = y1;
p.y2 = y2;
p.k = k;
ar.Add(p); //p'yi ar dizisine ekle.
}
```

Form1\_MouseDown olayına aşağıdaki kodları yazınız.

```
//Mouse butonlarından biri tıklandığında buradaki kodlar çalışacaktır.
if (e.Button == MouseButton.Left)
{
    x1 = e.X;
    y1 = e.Y;
}
```

Form1\_Paint olayına aşağıdaki kodları yazınız.

```
Pen p1 = new Pen(k, 3);
e.Graphics.DrawLine(p1, x1, y1, x2, y2);
for (int i = 0; i < ar.Count; i++)
{
    sekil s1 = (sekil)ar[i]; //ar isimli dizimizdeki elemanları kullanmak için yazıldı.
    Pen p2 = new Pen(s1.k, 3);
    e.Graphics.DrawLine(p2, s1.x1, s1.y1, s1.x2, s1.y2);
}
```

Button1\_Click olayına aşağıdaki kodu yazınız.

```
if (colorDialog1.ShowDialog() == DialogResult.OK)
{
    k = colorDialog1.Color; //colorDialog1'deki rengi k'ya ata.
    Invalidate();
}
```

Button2\_Click olayına aşağıdaki kodları yazınız.

```
ar.Clear(); //ar dizisinin içindeki elemanları siler.
Invalidate();
x1 = 0; y1 = 0; x2 = 0; y2 = 0;
```

Button3\_Click olayına aşağıdaki kodları yazınız.

```
if (ar.Count > 0) //ar dizisinin eleman sayısı sıfırdan büyükse.
{
    ar.RemoveAt(ar.Count - 1); //Sonuncu elemanı diziden çıkar.
    x1 = 0; y1 = 0; x2 = 0; y2 = 0;
```

```
    Invalidate();  
}
```

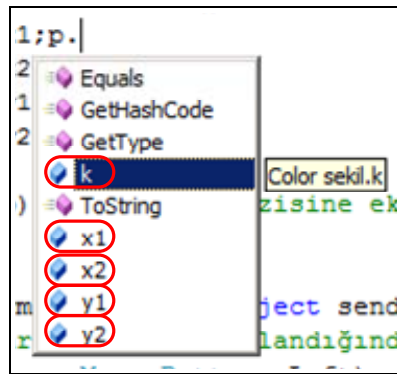
Form1\_Load olayına aşağıdaki kodu yazınız.

```
this.SetStyle(ControlStyles.AllPaintingInWmPaint, true);  
this.SetStyle(ControlStyles.OptimizedDoubleBuffer, true);
```

Programınızı çalıştırabilirsiniz. Ar isimli dizinizde çizilen çizgilerin renkleri, başlangıç ve bitiş koordinatları saklanmaktadır. Form1.cs içinde

```
sekil p = new sekil();
```

şekilde `sekil` sınıfından bir `p` değişkeni tanımladığınızda, `p` değişkeni yardımıyla `sekil` sınıfının içinde tanımladığınız `public` değişkenlere erişebilirsiniz(Şekil 30).



Şekil 30

## Örnek 12

Formunuza 2 adet GroupBox, 2 adet checkBox, 2 adet Label ve GroupBox1 ile GroupBox2'nin içine 2'şer adet RadioButton'u ekleyiniz. Radiobutton1'in üzerine çift tıklayarak aşağıdaki kodu yazınız.

```
if (radioButton1.Checked == true) this.Text = "1. radyo butonu seçildi.";
```

Daha önceki if kullanımlarında kıvrık parantez açıp kapadıktan sonra içerisine kodu yazıyorduk. Buradaki kullanımda ise kıvrık parantez kullanmadık. Bunun nedeni if koşulundan sonra uygulanacak kodlar tek satır(ifade) ise kodların kıvrık parantezler içerisine alınması gerekmemektedir. Benzer durum else içinde geçerlidir. Radiobutton2'nin üzerine çift tıklayarak aşağıdaki kodu yazınız.

```
if (radioButton2.Checked == true) this.Text = "2. radyo butonu seçildi.";
```

Radiobutton3'ün üzerine çift tıklayarak aşağıdaki kodu yazınız.

```
if (radioButton3.Checked == true) this.BackColor = Color.Blue;
```

Radiobutton4'ün üzerine çift tıklayarak aşağıdaki kodu yazınız.

```
if (radioButton4.Checked) this.BackColor = Color.Red;
```

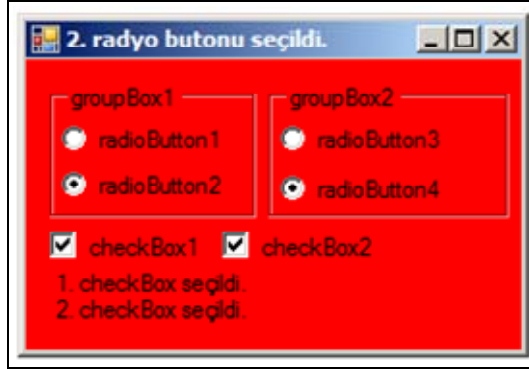
Radiobutton3 için `if (radioButton3.Checked == true)` kodunu yazarken Radiobutton4 için `if (radioButton4.Checked)` yazdık. Kodlar farklı olmasına rağmen yaptıkları iş aynıdır. CheckBox1'in üzerine çift tıklayarak aşağıdaki kodu yazınız.

```
if (checkBox1.Checked) label1.Text = "1. checkBox seçildi.";
else label1.Text = "";
```

CheckBox2'nin üzerine çift tıklayarak aşağıdaki kodu yazınız.

```
if (checkBox2.Checked) label2.Text = "2. checkBox seçildi.";
else label2.Text = "";
```

Programınızdan bir görüntü aşağıdaki gibidir.

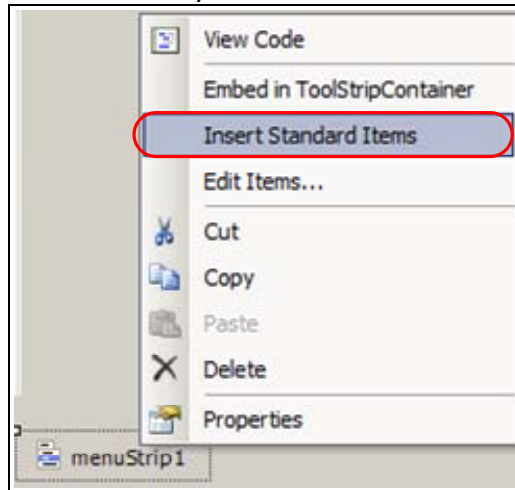


Şekil 31

CheckBox'lar da aynı anda birden fazla seçim yapılabilir. RadioButton'lar da ise aynı anda sadece bir tanesi seçilebilir. Eğer RadioButtonları gruplandırılmışsa gruplandırılmış olanlar arasından aynı anda sadece bir tanesi seçilebilir. Örneğin yukarıdaki örnekte aynı anda radioButton1 ve radioButton2 seçilemez.

## Örnek 13

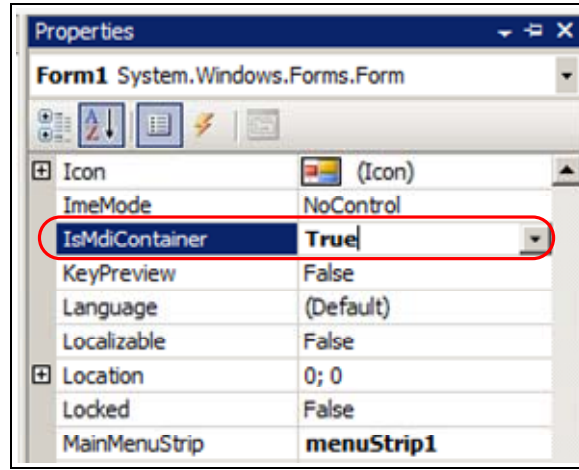
Formunuza 1 adet MenuStrip ekleyiniz. MenuStrip ile Formunuza menüler ekleyebilirsiniz. "Type Here" yazan yerlere istediğiniz menüleri ekleyebilirsiniz veya Formunuza attığınız MenuStrip'in üzerine sağ tıkladıktan sonra "Insert Standard Items"a tıklayarak standart elemanları ekleyebilirsiniz(Şekil 32).



Şekil 32

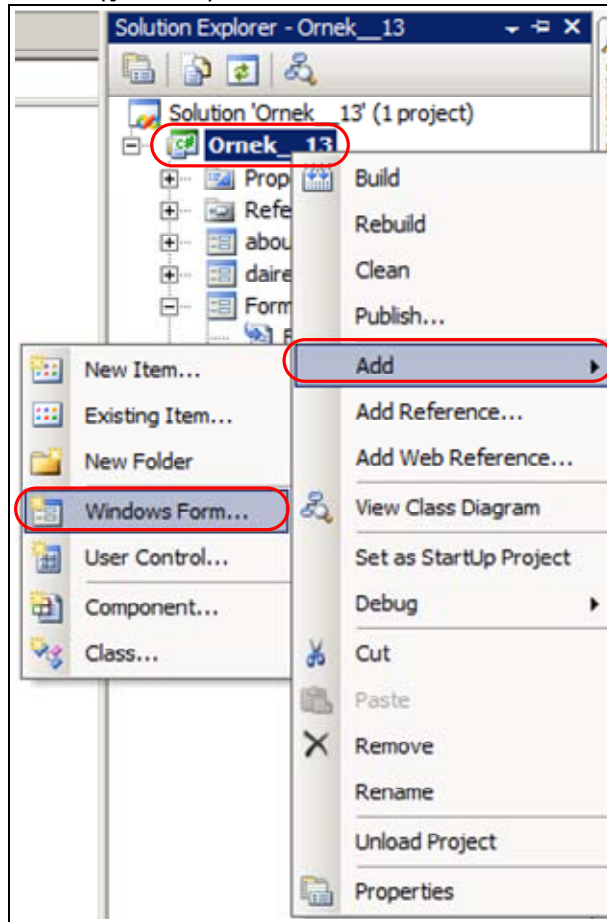
Bu örnek'te standart elemanlar eklenecektir. Form'unuzun Properties penceresinden "IsMdiContainer" özelliğini "True" yapınız(Şekil 33).



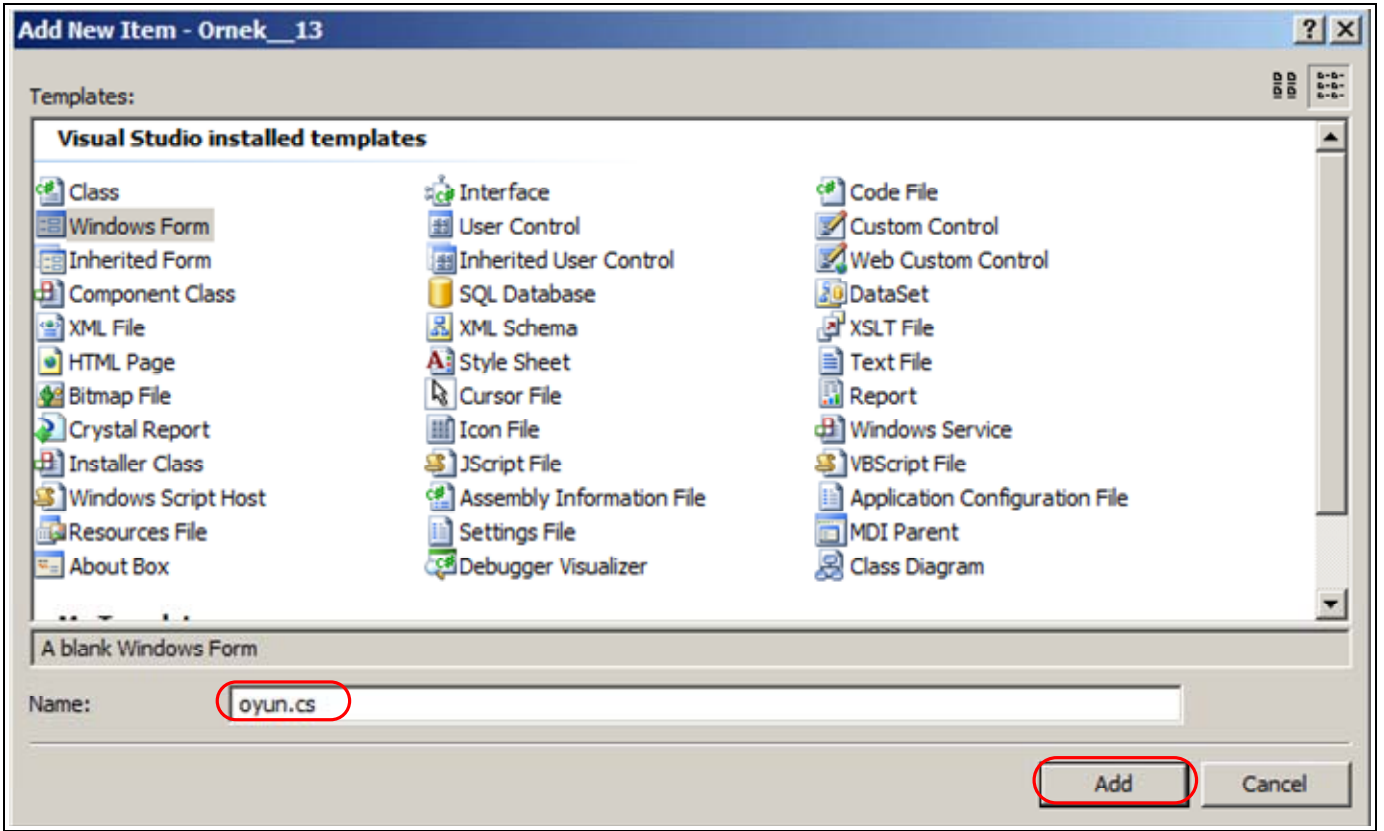


Şekil 33

Projenize “about”, “daire” ve “oyun” isimli 3 adet Form ekleyiniz. Projenize Form eklemek için Solution Explorer’da projenizin ismine sağ tıkladıktan sonra “Add” -> “Windows Form...”a tıklayarak “Add New Item” penceresine ulaşınız(Şekil 34). Bu pencerede “Name” yazan kısmın karşısına yeni Form’unuzun ismini yazıp “Add” butonuna basınız(Şekil 35).

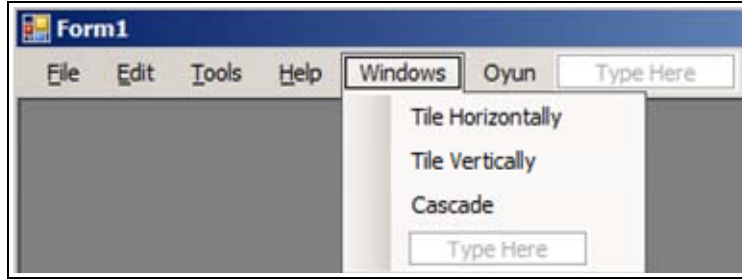


Şekil 34

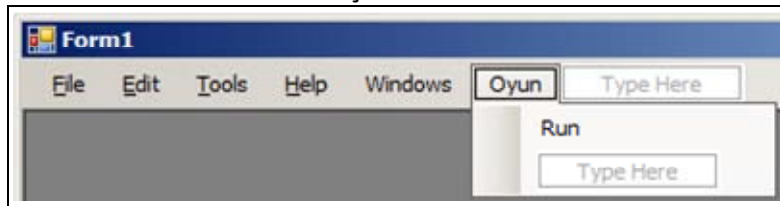


Şekil 35

Menünüz Şekil 36 ve Şekil 37'deki gibi olacak şekilde yeni elemanlar ekleyiniz.



Şekil 36



Şekil 37

“Tile Horizontally” menüsüne çift tıklayarak aşağıdaki kodu yazınız.

```
this.LayoutMdi(MdiLayout.TileHorizontal); //Pencereleri yatay olarak sıralar ve gösterir.
```

“Tile Vertically” menüsüne çift tıklayarak aşağıdaki kodu yazınız.

```
this.LayoutMdi(MdiLayout.TileVertical); //Pencereleri dikey olarak sıralar ve gösterir.
```

“Cascade” menüsüne çift tıklayarak aşağıdaki kodu yazınız.

```
this.LayoutMdi(MdiLayout.Cascade); //Pencereleri basamaklanmış şekilde gösterir.
```

“Run” menüsüne çift tıklayarak aşağıdaki kodu yazınız.

```
oyun t = new oyun();
t.MdiParent = this; //t Form'unun Parent'ını belirler.
t.WindowState = FormWindowState.Maximized; //Pencereyi maximum boyutlara getirir.
t.Show(); //Pencereyi gösterir.
```

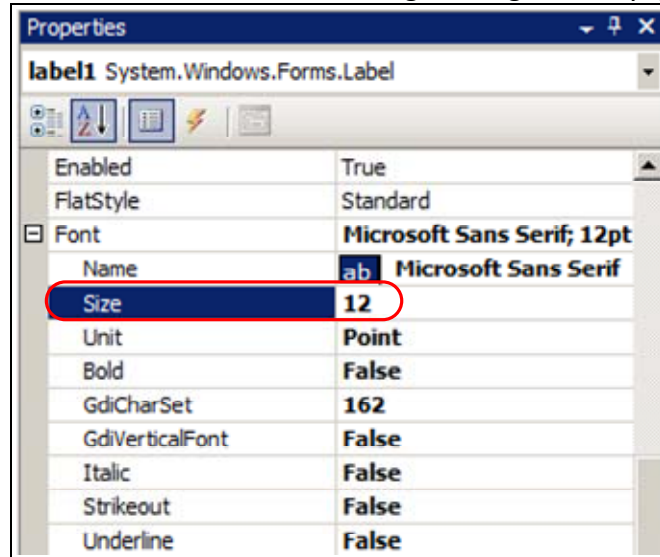
“File” menüsünün alt menüsü olan “New”e çift tıklayarak aşağıdaki kodu yazınız.

```
daire c = new daire();
c.MdiParent = this;
c.Show();
```

“Help” menüsünün alt menüsü olan “About”a çift tıklayarak aşağıdaki kodu yazınız.

```
about a=new about();
a.Show();
```

“About” isimli Form’unuza 1 adet “Label” ekleyiniz ve “Text” özelliğine “Bu Bizim Programdır.” yazınız. Label’ınızın “Font” özelliğinin altında bulunan “Size” özelliğinin değerini 12 yapınız(Şekil 38).



Şekil 38

Daire Form’unuzun daire\_Paint olayına aşağıdaki kodu yazınız.

```
e.Graphics.FillEllipse(Brushes.Blue, 20, 50, 100, 100);
```

Oyun Form’unuzun kod sayfasında genel değişkenlerinizi aşağıdaki gibi tanımlayınız.

```
public partial class oyun : Form
{
    int x, y;
    Random ra = new Random(); //Random sınıfından ra isimli bir değişken tanımlandı.
    int puan = 0;
```

Oyun\_Paint olayına aşağıdaki kodu yazınız.

```
x = ra.Next(0, 600); //0 ile 600 arasında rastgele bir sayı üretip x değişkenine atar.
y = ra.Next(0, 500);
e.Graphics.DrawEllipse(Pens.Blue, x, y, 50, 50);
```

Oyun Form'unuza 1 adet Timer ekleyiniz. Timer'ın Enabled özelliğini True ve Interval özelliğini 1000 yapınız. Timer1\_Tick olayına aşağıdaki kodu yazınız.

```
Invalidate();
```

Oyun\_MouseClick olayına aşağıdaki kodu yazınız.

```
if ((e.X > x && e.X < (x + 50)) && (e.Y > y && e.Y < (y + 50))) //Dairenin içine tıklanmış ise
{
    puan += 100;
    timer1.Interval -= 100;
    this.Text = puan.ToString();
}
```

Programınızı çalıştırabilirsiniz. "Daire" ile "Oyun" Form'ları ChildForm iken "About" Form'unda böyle bir özellik yoktur. "Daire" ile "Oyun" Form'ları "Form1"ın sınırladığı alanın dışına çıkamazlar.

## Örnek 14

Bu örnekte seri robot simülasyonu gerçekleştirilmektedir. Form'unuza 2 adet Buton ve 1 adet Timer ekleyiniz. Timer'ın Interval özelliğini 1 yapınız. Projenize "line" ve "link" isimli 2 adet sınıf ekleyiniz. Line sınıfınızın içine aşağıdaki kodu yazınız.

```
class line
{
    public double xee2,yee2;
}
```

Link sınıfınızın içine aşağıdaki kodu yazınız.

```
class link
{
    public double l, teta;
    public static double x1=0.0, y1=0.0; //Static değişkenler birden fazla değer saklayamaz. Örneğin ar dizisinde saklanan linklerin her birinin l veya teta değerleri farklı olabilir fakat x1 ve y1 1 adet değer saklayabilir.
}
```

Formunuzun genel değişkenlerini aşağıdaki gibi tanımlayınız.

```
public partial class Form1 : Form
{
    double x1, y1, x2, y2;
    ArrayList ar = new ArrayList();
    ArrayList ar1 = new ArrayList();
    int a;
```

Form1\_MouseDown olayına aşağıdaki kodu yazınız.

```
if (e.Button == MouseButton.Left)
{
    x1 = e.X;
```

```

        y1 = e.Y;
    }
    if (link.x1 == 0.0 && link.y1 == 0.0) //Eğer bu if ve içindeki kodlar yazılmazsa robotun başlangıç noktası sol üst köşe olmaktadır.
    {
        link.x1 = e.X;//link.x1 ve link.y1 robotun başlangıç noktasının koordinatlarıdır.
        link.y1 = e.Y;
    }

```

Form1\_MouseUp olayına aşağıdaki kodu yazınız.

```

if (e.Button == MouseButton.Left)
{
    x2 = e.X;
    y2 = e.Y;
    link k = new link();
    double y = y2 - y1;
    double x = x2 - x1;
    k.teta = Math.Atan2(y, x);
    k.l = Math.Sqrt(y * y + x * x);
    ar.Add(k); //ar dizisinde robotun linkleri saklanıyor. Her linkin açısı(teta) ve uzunluk(l) özelliği vardır.
    Invalidate();
}

```

Form1\_MouseMove olayına aşağıdaki kodu yazınız.

```

if (e.Button == MouseButton.Left)
{
    x2 = e.X;
    y2 = e.Y;
    a = ar.Count;
    Invalidate();
}
if (e.Button == MouseButton.Right)
{//Mouse'un sağ butonu ile robotun başlangıç noktası değiştirilebilir.
    link.x1 = e.X;
    link.y1 = e.Y;
    Invalidate();
}

```

Form1\_Paint olayına aşağıdaki kodu yazınız.

```

if (a == ar.Count) //Eğer bu if ve içindeki kodlar yazılmazsa son link'ten 2 adet oluyor. Biri robota ait biri ise robottan bağımsız oluyor.
{
    e.Graphics.DrawLine(Pens.Black, Convert.ToSingle(x1), Convert.ToSingle(y1),
Convert.ToSingle(x2), Convert.ToSingle(y2));
}
double xe1 = link.x1;
double ye1 = link.y1;

```

```

double xe2 = link.x1;
double ye2 = link.y1;
for (int i = 0; i < ar.Count; i++) //Robotun linklerini çizdirmeye yarıyor.
{
    link k = (link)ar[i];
    xe2 += k.l * Math.Cos(k.teta);
    ye2 += k.l * Math.Sin(k.teta);
    e.Graphics.DrawLine(Pens.Black, Convert.ToSingle(xe1), Convert.ToSingle(ye1),
Convert.ToSingle(xe2), Convert.ToSingle(ye2));
    xe1 = xe2;
    ye1 = ye2;
}
for (int i = 1; i < ar1.Count; i++) //Robotun uç noktasının geçtiği yolu çizdirmeye yarıyor.
{
    line k = (line)ar1[i-1];
    line k1 = (line)ar1[i];
    e.Graphics.DrawLine(Pens.Blue, Convert.ToSingle(k.xee2), Convert.ToSingle(k.yee2),
Convert.ToSingle(k1.xee2), Convert.ToSingle(k1.yee2));
}

```

Timer1\_Tick olayına aşağıdaki kodu yazınız.

```

double d=0;
double xee2 = link.x1, yee2=link.y1;
for (int i = 0; i < ar.Count; i++) //Robotun uç noktasının koordinatlarını buluyor ve her linkin açısını
bir önceki linkten 0.05 radyan fazla olmasını sağlıyor.
{
    d+=0.05;
    link k = (link)ar[i];
    xee2 += k.l * Math.Cos(k.teta);
    yee2 += k.l * Math.Sin(k.teta);
    k.teta += d;
}
line n = new line();
n.xee2 = xee2;
n.yee2 = yee2;
ar1.Add(n); //Robotun uç noktasının koordinatları ar1 dizisinde saklanmaktadır.
Invalidate();

```

Button1\_Click olayına aşağıdaki kodu yazınız.

```
timer1.Enabled = !timer1.Enabled;
```

Form1\_Load olayına aşağıdaki kodu yazınız.

```
this.SetStyle(ControlStyles.AllPaintingInWmPaint, true);
this.SetStyle(ControlStyles.OptimizedDoubleBuffer, true);
```

Button2\_Click olayına aşağıdaki kodu yazınız.

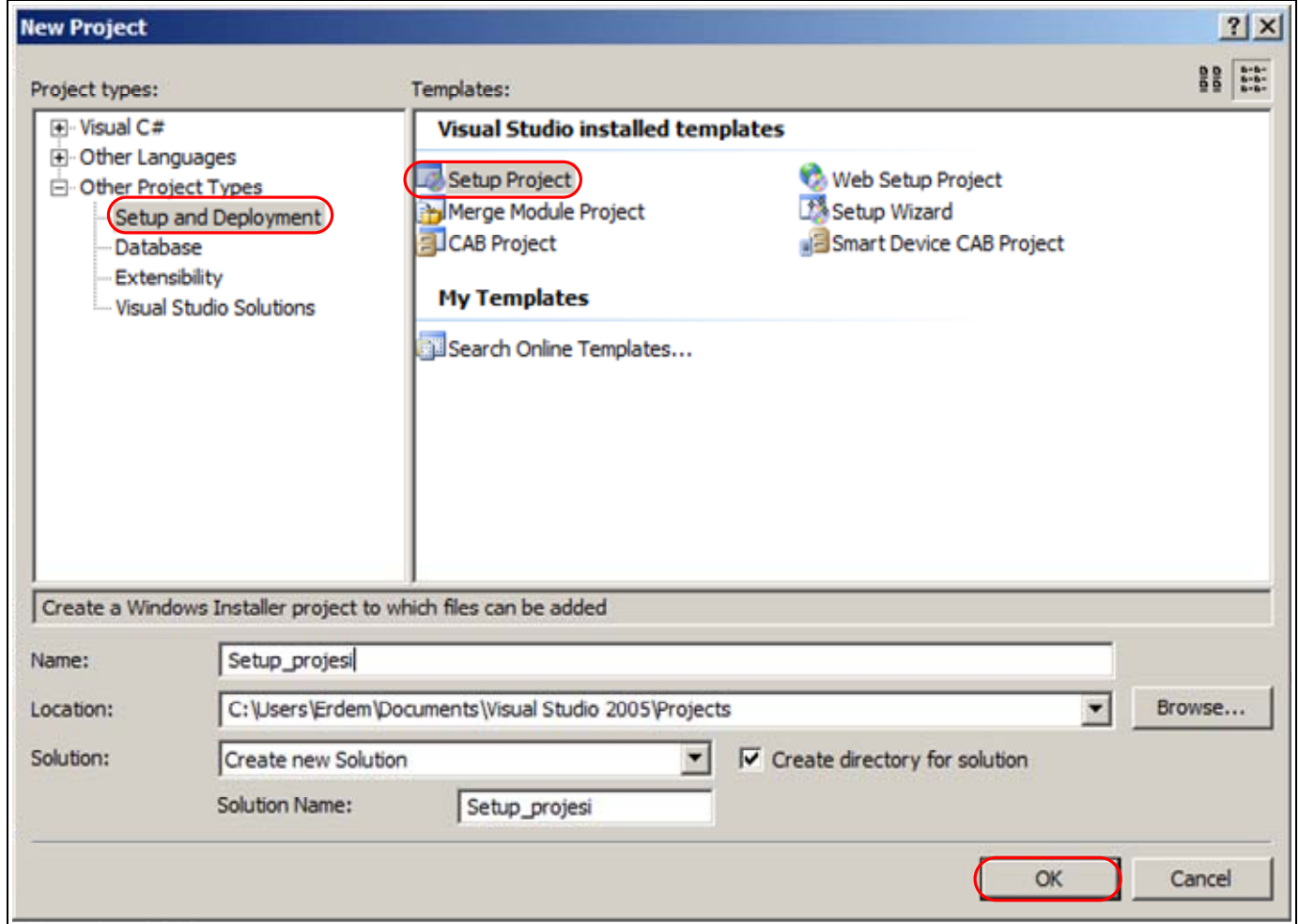
```
ar1.Clear(); //ar1 dizisinin elemanlarını siliyor.
```

Invalidate());

Programınızı çalıştırabilirsiniz. Programınıza ayrıca ColorDialog ekleyerek linklerin rengini değiştirebilirsiniz veya HScrollBar ekleyerek linklerin kalınlığını değiştirebilirsiniz.

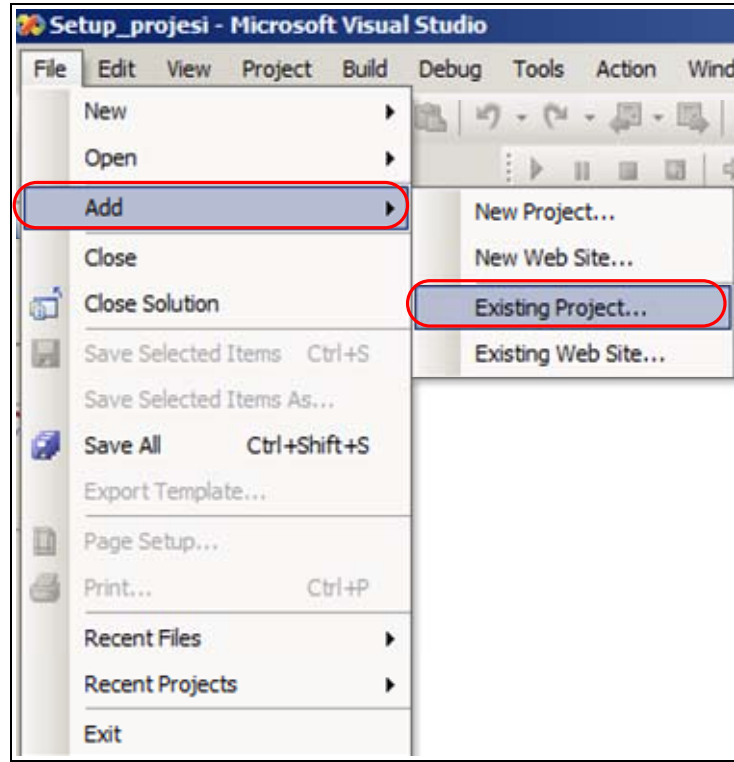
## Örnek 15

Bu örnekte daha önce hazırlamış olduğunuz bir programın Setup dosyasını hazırlamak anlatılacaktır. Visual Studio .NET programınız açıkken File -> New -> Project... ile "New Project" penceresini açınız. "Project Types" kısmında "Other Project Types"ın altındaki "Setup and Deployment" seçeneğini seçiniz ve "Templates"ın altındaki "Setup Project" seçeneğini seçerek OK butonuna basınız(Şekil 39).

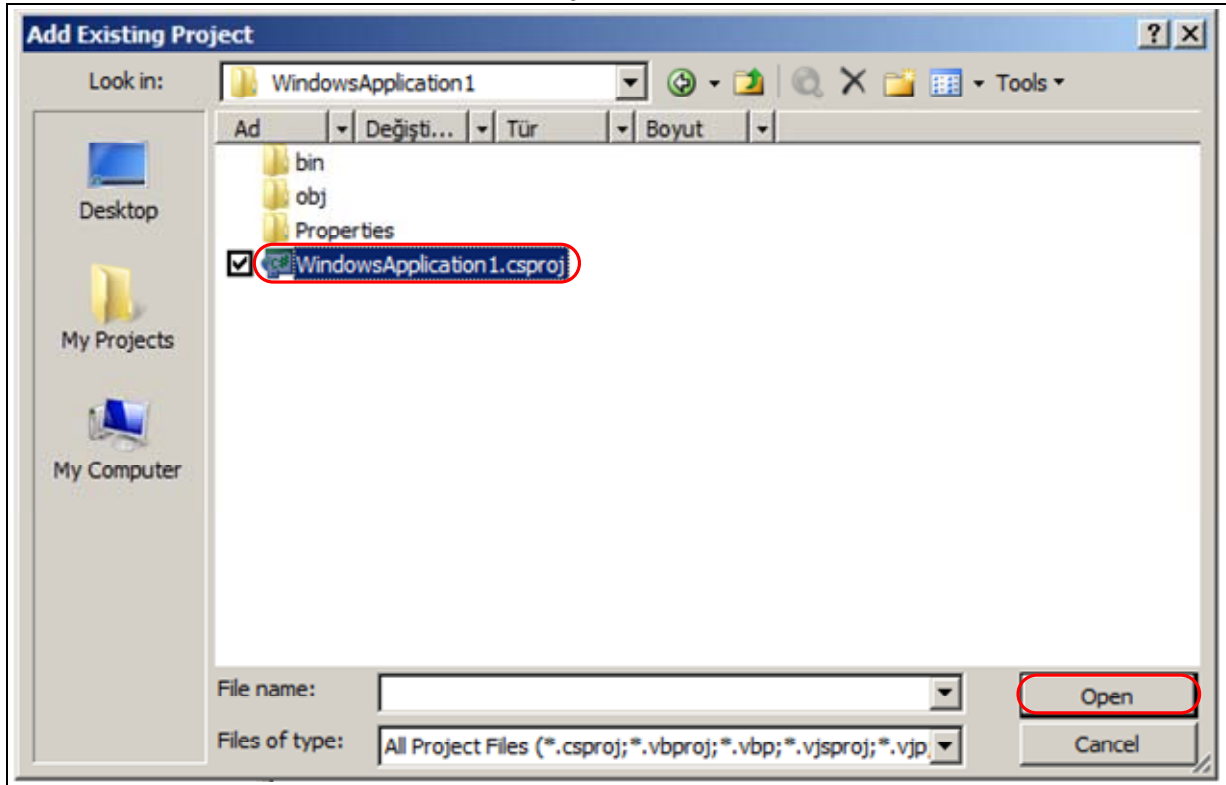


Şekil 39

Önceden hazırladığınız projeyi File -> Add -> Existing Project... ile Setup projenize ekleyiniz(Şekil 40 - 41).



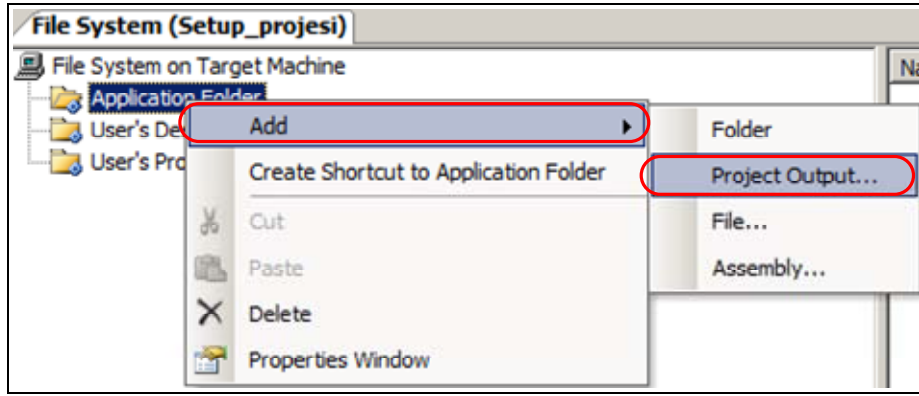
Şekil 40



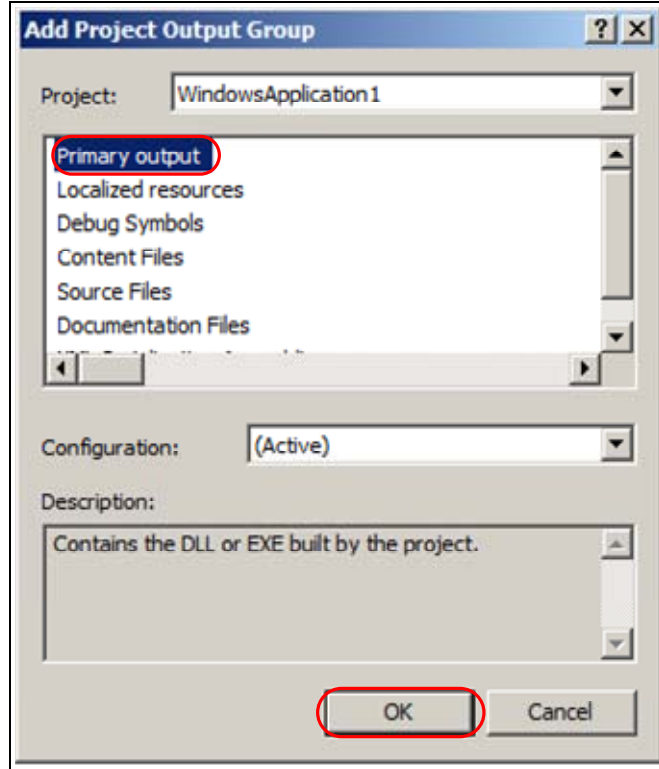
Şekil 41

“File System on Target Machine”in altında olan “Application Folder”a sağ tıklayıp Add -> Project Output...’a tıklayınız(Şekil 42). “Add Project Output Group” penceresinden “Primary output” seçeneğini seçtikten sonra “OK” butonuna basınız(Şekil 43).



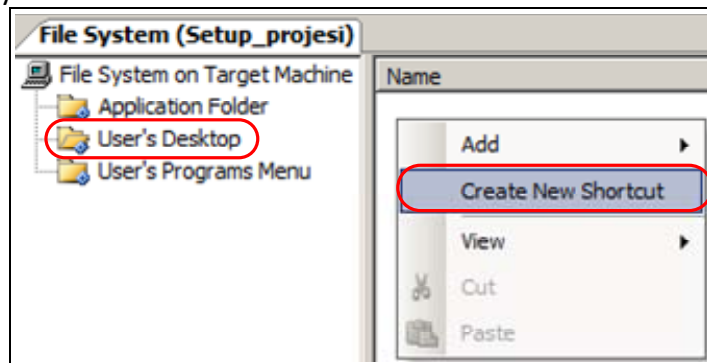


Şekil 42

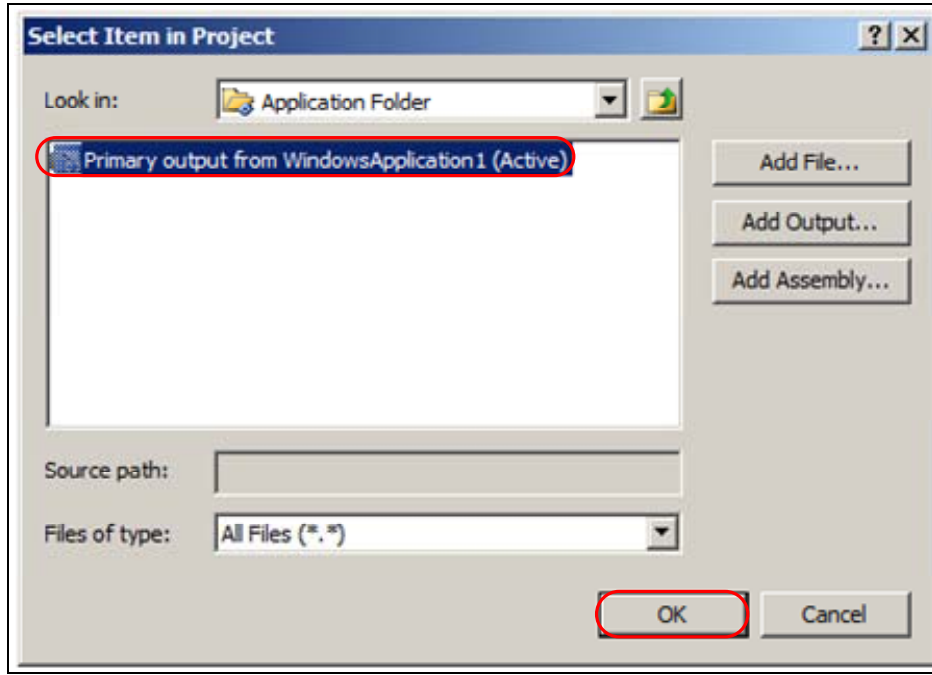


Şekil 43

“File System on Target Machine”in altında olan “User’s Desktop” seçili iken “Name”, “Type” yazan kısmın altında sağ tıkladıktan sonra “Create New Shortcut” a tıklayınız(Şekil 44). Karşınıza çıkan “Select Item in Project” penceresinden “Application Folder”ın içinde bulunan “Primary output from WindowsApplication1(Active)”i seçtikten sonra OK butonuna tıklayınız(Şekil 45). Sizin projenizde “WindowsApplication1” yazan kısım farklı olabilir.

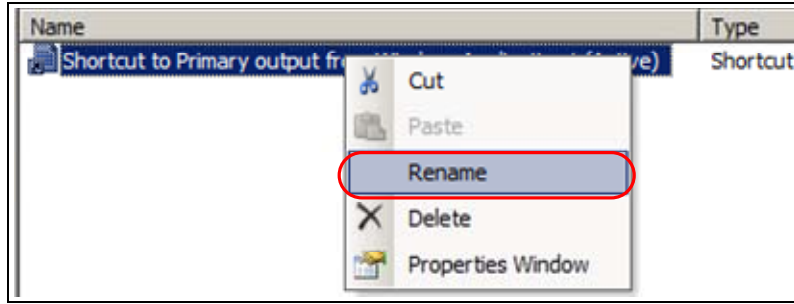


Şekil 44



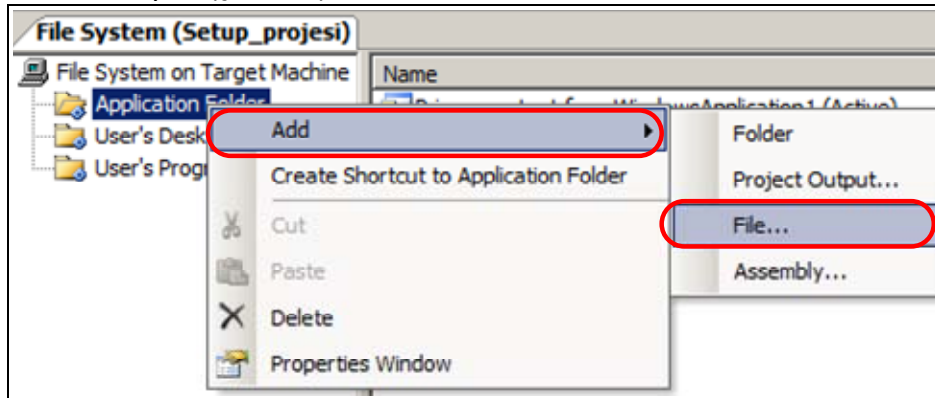
Şekil 45

Şekil 45'te yaptığınız ile Programınızın Setup'ını yaptığınızda Masaüstü'ne programın kısayolunu atmaktadır. Kısayolun ismini sağa tıkladıktan sonra "Rename"e tıklayıp değiştirebilirsiniz(Şekil 46).

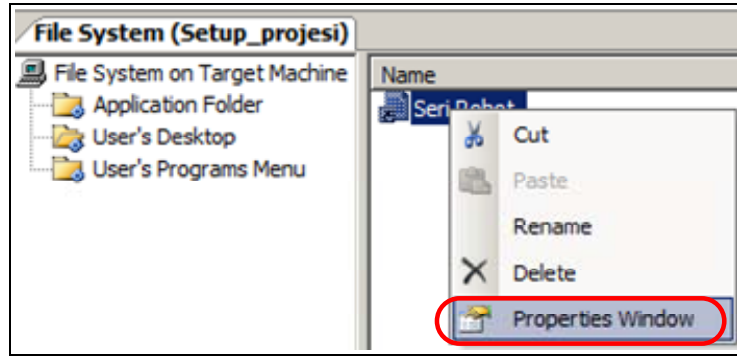


Şekil 46

"User's Desktop" seçili iken yaptığınız kısayol oluşturmayı "User's Program's Menu" seçili iken de yapınız. Application Folder'a sağ tıkladıktan sonra Add -> File... 'a tıklayınız(Şekil 47). Bilgisayarınızdan bir icon'u seçtikten sonra "Open" a tıklayınız. "User's Desktop"ta oluşturduğunuz kısayola sağ tıkladıktan sonra "Properties Window"a tıklayınız(Şekil 48).

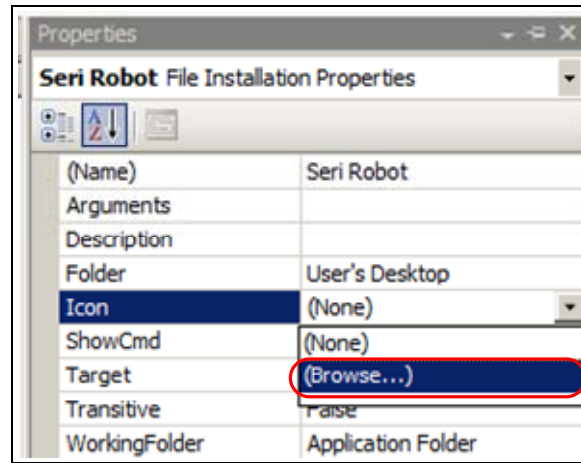


Şekil 47



Şekil 48

Properties penceresinde "Icon" özelliğinin yanında bulunan "(Browse...)"u seçiniz(Şekil 49). Karşınıza Icon penceresi çıkacaktır. Browse butonuna bastıktan sonra "Application Folder" içindeki iconunuzu seçiniz ve OK butonuna basınız.



Şekil 49

"User's Desktop"taki kısayola yaptığınız icon atama işlemlerinin aynısını "User's Program Menu"deki kısayol içinde yapınız. Programınızın hatasız çalıştığını kontrol ediniz(Aynı zamanda programınızı kaydetmiş oluyorsunuz). Bu örnekte proje dosyasının ismi "Setup\_projesi" verilmişti. Sizin proje dosyanızın ismi farklı olabilir. Programı bilgisayarınıza yüklemek için Setup\_projesi -> Setup\_projesi -> Debug -> setup.exe adımlarını takip etmelisiniz.

## Örnek 16

Formunuza 1 adet TextBox, 2 adet label ve 3 adet buton ekleyiniz. Using'lerin olduğu kısma aşağıdaki kodu ekleyiniz.

`using System.Threading; //Thread.Sleep kodunun kullanılabilmesi için yazılmaktadır.`

Button1'e çift tıklayarak aşağıdaki kodu yazınız.

```
double a;  
a = Convert.ToDouble(textBox1.Text);  
for (double i = 0; i < a; i++)  
{  
    Application.DoEvents(); //Form kontrollerinin cevap vermesini sağlamaktadır.  
    this.Text = i.ToString();  
}
```

Button2'ye çift tıklayarak aşağıdaki kodu yazınız.

```
label1.Text = "merhaba";
```

Button3'e çift tıklayarak aşağıdaki kodu yazınız.

```
label1.Text = "";  
double a;  
a = Convert.ToDouble(textBox1.Text);  
for (double i = 0; i < a; i++)  
{  
    Application.DoEvents();  
    Thread.Sleep(1000); //1000ms bekleme yaptırmaktadır.  
    label2.Text = i.ToString();  
}
```

Button4'e çift tıklayarak aşağıdaki kodu yazınız.

```
double a;  
a = Convert.ToDouble(textBox1.Text);  
for (double i = 0; i < a; i++)  
{  
    if (i > 8)  
    {  
        break;//for'dan çıkılmasını sağlamaktadır.  
    }  
    Application.DoEvents();  
    Thread.Sleep(1000);  
    label2.Text = i.ToString();  
}
```

## Örnek 17

Bu örnekte Krank-biyel mekanizmasının animasyonu yapılmaktadır. Formunuza 2 adet yatay kaydırma çubuğu(HScrollBar), 1 adet zamanlayıcı(Timer) ekleyiniz. Timer'ınızın Enabled özelliğini True ve Interval özelliğini 20 olarak ayarlayınız.

Genel(Global) değişkenlerinizi aşağıdaki gibi tanımlayınız.

```
public partial class Form1 : Form  
{  
    double x, y, teta, x1, L=100d, M=200d;
```

Form'unuza çift tıklayarak Form1\_Load olayına aşağıdaki kodu yazınız.

```
this.SetStyle(ControlStyles.AllPaintingInWmPaint, true);  
this.SetStyle(ControlStyles.OptimizedDoubleBuffer, true);  
hScrollBar1.Value = Convert.ToInt32(L);  
hScrollBar2.Value = Convert.ToInt32(M);
```

HScrollbar1'in üzerine çift tıklayarak hScrollBar1\_Scroll olayına aşağıdaki kodu yazınız.

```
L = Convert.ToDouble(hScrollBar1.Value);
```

HScrollbar2'nin üzerine çift tıklayarak hScrollBar2\_Scroll olayına aşağıdaki kodu yazınız.

```
M = Convert.ToDouble(hScrollBar2.Value);
```

Timer1'in üzerine çift tıklayarak timer1\_Tick olayına aşağıdaki kodu yazınız.

```
Invalidate();
```

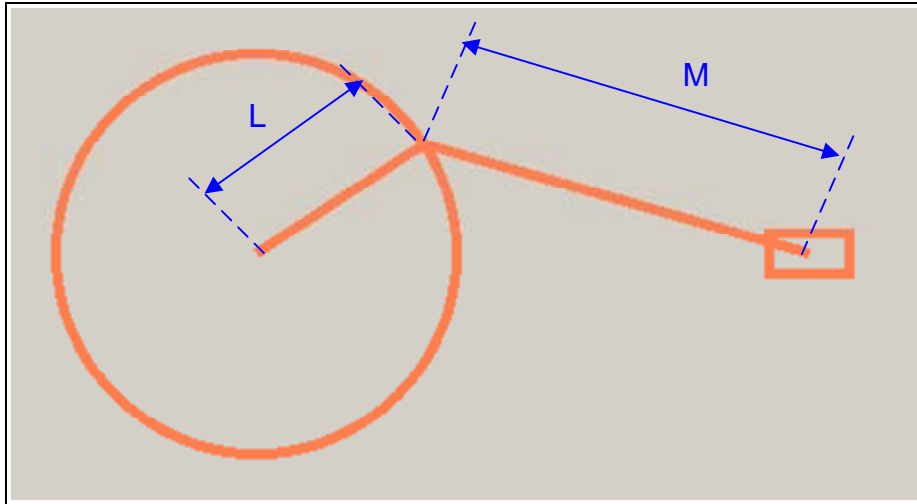
Convert.ToSingle kodunu fonksiyon olarak yazalım. Aşağıdaki fonksiyonu kodunuza ekleyiniz.

```
private float c(double a)
{
    return Convert.ToSingle(a);
}
```

Form1\_Paint olayına aşağıdaki kodu yazınız.

```
Pen p = new Pen(Color.Coral, 5);
e.Graphics.TranslateTransform(200, 200);
teta += 0.105;
x = L * Math.Cos(teta);
y = L * Math.Sin(teta);
e.Graphics.DrawLine(p, 0, 0, c(x), c(y));
x1 = x + Math.Sqrt(M * M - y * y);
e.Graphics.DrawLine(p, c(x), c(y), c(x1), 0);
e.Graphics.DrawEllipse(p, -100, -100, 200, 200);
e.Graphics.DrawRectangle(p, c(x1 - 20), -10, 40, 20);
```

L ve M uzunlukları Şekil 50'de gösterilmiştir.



Şekil 50