



# GÖRSEL PROGRAMLAMA

Yrd. Doç. Dr. Durmuş ÖZDEMİR

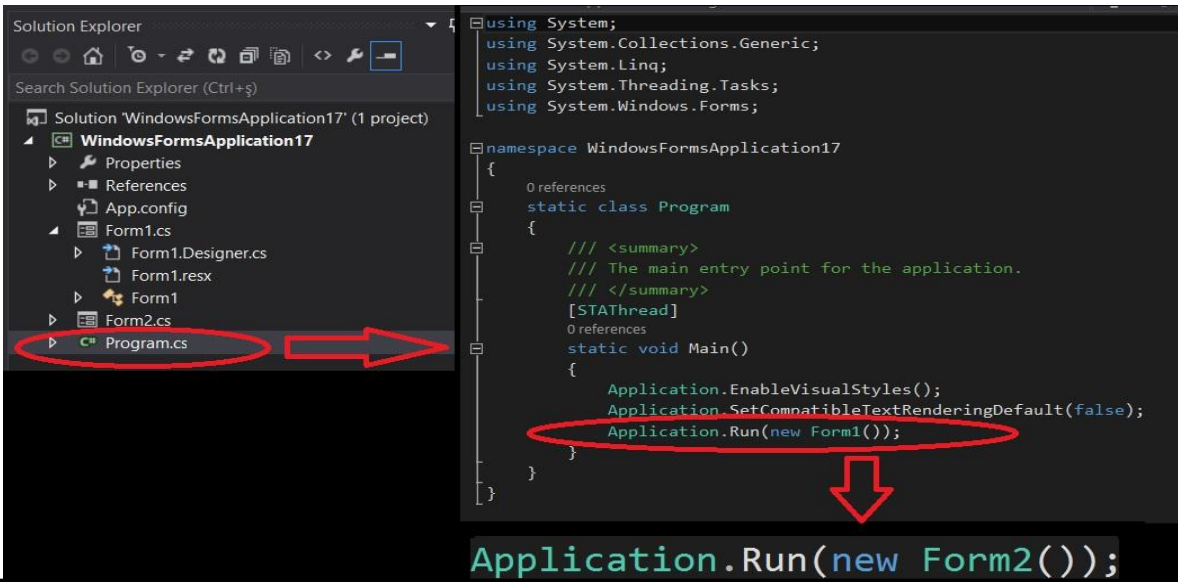
[durmus.ozdemir@dpu.edu.tr](mailto:durmus.ozdemir@dpu.edu.tr)

Hafta 7

## BU HAFTA

- Sonradan eklenen formu açılış formu olarak ayarlama
- Accept Button ve Cancel Button
- Backgroundimage
- CONTEXTMENUSTRIP
- DATETIMEPICKER
- DOMAINUPDOWN
- ERRORPROVIDER
- KEYPREVIEW

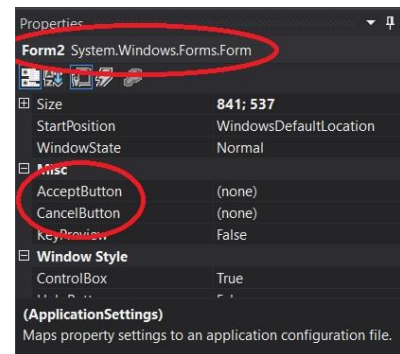
## Sonradan Eklenen Formu Açılış Formu Olarak Ayarlama



## ACCEPTBUTTON ve CANCELBUTTON

Enter ve Esc tuşuna basılınca, belirlenen ilgili butonun içindeki kodun aktif olmasını sağlar. Formun properties penceresinden yapılabileceği gibi kod satırından da yazılabilir.

```
1 reference
private void Form2_Load(object sender, EventArgs e)
{
    this.AcceptButton = button1;
    this.CancelButton = button2;
}
```

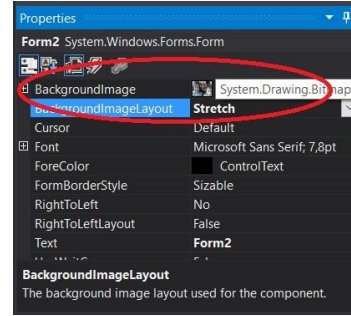


## BACKGROUNDIMAGE

**Tile:** Resim Küçük geliyorsa ekranı kaplaması için kullanılır.

**Center:** Resmin arka planda ortalı olarak durmasını sağlar.

**Stretch:** Küçük ise büyütür, büyük ise küçültür form üzerini kaplamasını sağlar.

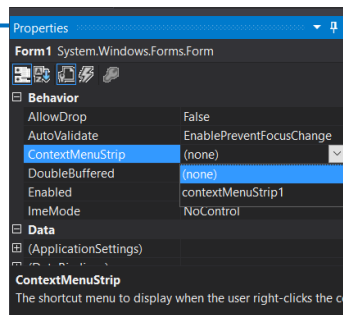


## CONTEXTMENUSTRIP

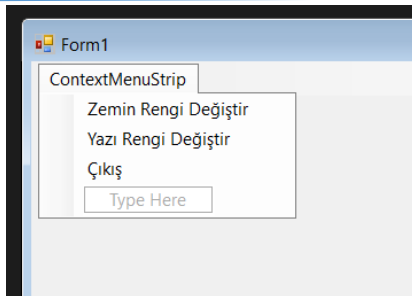
Forma sağ tıklanınca gelen menüyü hazırlamak için kullanılır.

Toolbox'tan  **ContextMenuStrip** kontrolü forma taşınarak içerik oluşturulur.

Ayrıca form ile bağlantı yapabilmek için properties penceresinden ContextMenuStrip özelliğini kullanarak hangi contextMenuStrip'i kullanacaksanız seçmeniz gerekir.



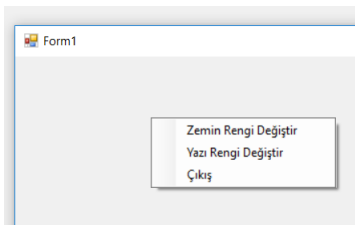
## Örnek: CONTEXTMENUSTRIP



```
1 reference
private void kesToolStripMenuItem_Click(object sender, EventArgs e)
{
    this.BackColor = Color.Red;
}

1 reference
private void kopyalaToolStripMenuItem_Click(object sender, EventArgs e)
{
    this.ForeColor = Color.Blue;
}

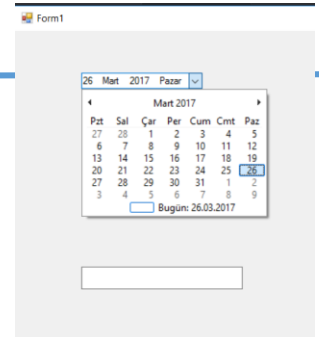
1 reference
private void çıkışToolStripMenuItem1_Click(object sender, EventArgs e)
{
    this.Close();
}
```



Not: Tasarım esnasında "-" çizgi işareti koyarsanız menüde bölüm çizgisi yapar.

## DATE TIME PICKER

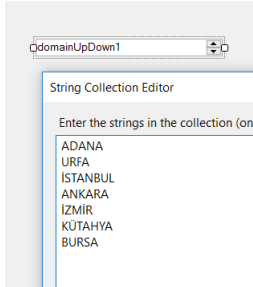
dateTimePicker1\_ValueChanged Event kullanılarak seçilen tarihin isteğimiz kontrolün içerisine aktarılmasını sağlayabiliriz..



```
private void dateTimePicker1_ValueChanged(object sender, EventArgs e)
{
    textBox1.Text = dateTimePicker1.Text;
}
```

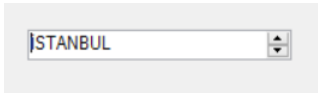
## DOMAINUPDOWN

ComboBox kontrolünün açılmayan şeklidir. Elemanları arasında gezinme işlemi yön ok tuşları ile yapılabilir. Yine Items özelliği ile liste eklenir.



Örneğin bir textBox'tan eleman eklenebilir.

```
domainUpDown1.Items.Add(textBox1.Text.Trim());
```

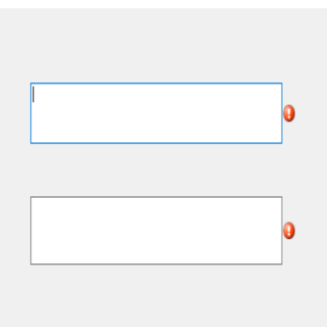


## ERRORPROVIDER:



ErrorProvider

Bu özellik herhangi bir hata durumunda ya da boş geçilmemesi istenilen durumlarda kontrolün yanında kırmızı ünlem (!) işaretinin çıkmasını sağlar. textBox'ların boş geçilmesini istemediğiniz durumlarda Validating ve Validated Event'ları kullanılır.



```
private void textBox1_Validating(object sender, CancelEventArgs e)
```

```
{
    if (textBox1.Text.Trim() == "")
        errorProvider1.SetError(textBox1, "Metin Kutusu 1'e değer girmeden ayrılamazsınız");
    else
        errorProvider1.SetError(textBox1, "");
}
```

```
private void textBox2_Validated(object sender, EventArgs e)
```

```
{
    if (textBox2.Text.Trim() == "")
        errorProvider2.SetError(textBox2, "Metin Kutusu 2'ye değer girmeden ayrılamazsınız");
    else
        errorProvider2.SetError(textBox2, "");
}
```

## KEYPREVIEW:

Program çalışırken imleç TextBox'da iken, klavyeden herhangi bir tuşa bastığınız anda, TextBox'ın **KeyDown** Event'ı işleyecektir. Yani basılan tuştan ilk olarak, **aktif olan kontrolün KeyDown'ı** haberdar olur.

Fakat bazı durumlarda **tüm kontrolleri ilgilendiren tuş vuruşları için**, genel bir kod yazmak gerekebilir.

İşte bu durumlarda **KEYPREVIEW özelliğine TRUE** değeri aktarılır.

```
private void textBox1_KeyDown(object sender, KeyEventArgs e)
{
    if (e.KeyCode == Keys.Left)
        textBox1.Left = textBox1.Left - 50;
    else if (e.KeyCode == Keys.Right)
        textBox1.Left = textBox1.Left + 50;
    else if (e.KeyCode == Keys.Up)
        textBox1.Top = textBox1.Top - 50;
    else if (e.KeyCode == Keys.Down)
        textBox1.Top = textBox1.Top + 50;
}
```

Not: Formun KeyPreview özelliği true olmalıdır.

## Modal (ShowDialog()) / Modeless(Show()) Form

Birden fazla Form ile çalıştığımız zaman, bir Form içinden diğer Form'u iki şekilde açabiliriz.

**Show ile açılan ikinci Form Modeless** Form'dur. Yani, öndeki formu kapatmadan arkadaki bir formu kapatabilirsiniz.

```
Form2 f = new Form2();
f.Show();
```

**ShowDialog kullandığımız zaman** açılan Form'u kapatmadan, arkadaki Formları kapatabiliriz.

```
Form2 f = new Form2();
f.ShowDialog();
```

## Controls.Add - Remove

Tasarım anında sürükle-bırak ile oluşturmuş olduğumuz her şeyin program içerisinde bir Kod karşılığı vardır. Controls.Add formun üzerine dinamik olarak kontrol eklemek için kullanılan metottur. Controls.Remove ise kontrolü silmek için kullanılır.

Yeni bir form açarak:

```
Button b = new Button ();
this.Controls.Add(b);
```

Programı çalıştırdığınızda şekildeki gibi,  
Formun sol üst köşesinde sabit ve  
özellikli olmayan bir buton oluşur.

```
// this. Controls.Remove (b);
```

```
Button b = new Button ();
b.SetBounds(50,100,200,30);
b.Text = "Manuel Butonum";
b.BackColor = Color. Yellow;
this.Controls.Add(b);
```

## KeyDown-KeyUp Kullanımı ve KeyCode

Klavyeden bir tuşun basılması veya tuşun basılı kalması durumunda devamlı olarak işleyen yordam(Event) KeyDown, tuşun bırakılması ile ise KeyUp çalışır. Basılan tuşun değeri KeyCode parametresine aktarılarak, istenilen işlem yapılır.

EN ÇOK KULLANILAN KEYCODE DEĞERLERİ			
A-Z	Keys.A-Key.Z	0-9	Keys.NumPad0-Keys.NumPad9
F1-F24	Keys.F1-Key.F24	Enter	Keys.Return
Insert	Keys.Insert	Del	Keys.Delete
Home	Keys.Home	End	Keys.End
*	Keys.Multiply	/	Keys.Divide
-	Keys.Subtract	+	Keys.Add

## KeyDown-KeyUp Kullanımı ve KeyValue

KeyValue ile basılan tuşun ASCII karşılığındaki değer öğrenilebilir.

```
1 reference
private void Form1_KeyDown(object sender, KeyEventArgs e)
{
    if (e.KeyCode == Keys.A)
    {
        MessageBox.Show("a harfine bastınız");
    }
    else if (e.KeyCode == Keys.Escape)
    {
        this.Close();
    }
    this.Text = e.KeyValue.ToString();
}
```

Not: KeyDown ve KeyUp kullanıldığı zaman mutlaka, Form özelliklerinden KeyPreview özelliği True yapılmalıdır.

## e.KeyData, e.Alt, e.Shift, e.Ctrl

e.KeyData Parametresi sayesinde, basılan tuşun karakter karşılığı öğrenilebilir. Mesela A tuşuna basıp ve A değerini bir değişkene almak veya yazdırmak amacıyla kullanılır.

e.Alt parametresi ile: Alt tuşu basılı iken bir başka tuşun basılması ile kullanıcıyı uyarabilirsiniz. e.Shift, e.Ctrl kullanımında aynıdır. Örneğin:

```
if(e.Alt)
{
    if((e.KeyCode == Keys.Right)) {
        MessageBox.Show(Şuanda Alt+Right tuşuna bastınız);}
}
```



## KEYPRESS ile Enter Tuşuna Basma Örneği

Bu event'ta klavyeden herhangi bir tuşa basılması ile otomatik olarak ve tek bir kez işlem yapar. İşleme zamanı KeyDown'dan sonra KeyUp'tan öncedir.

Burada basılan tuşun Ascii değerini e.KeyChar parametresi ile kıyaslayarak, istediğiniz işlemi yaptırabilirsiniz.

e.KeyChar Parametresi ile 13 numarası Ascii'de enter demektir.

```
if (e.KeyChar == 13) // if(e.KeyChar == (char)13)
{
    MessageBox.Show("Enter'a Bastın");
}
```

## DRAG-DROP (SÜRÜKLE-BIRAK)

Mouse yardımı ile bir kontrol içindeki değeri başka bir kontrolün içine taşıma ya da kopyalama imkanı sağlar.

Sürükle-Bırak olayının gerçekleşeceği kontrolün özelliği olan AllowDrop=True yapılmalıdır. Aksi halde işlem gerçekleşmez. Properties'ten ya da Form\_Load olayına yazabilirsiniz.

Genellikle DragOver Event'ı kullanılarak aşağıda verilen DragDropEffects parametreleri ile gerçekleşecek işlem belirlenir.

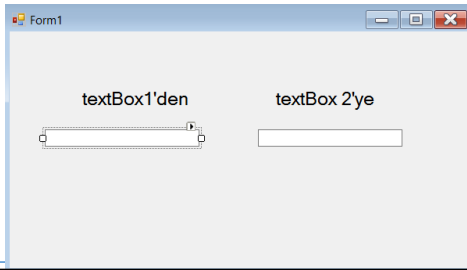
DragDropEffects'in alabileceği Değerler	
DragDropEffects.All	Bütün İşlemleri Yapar
DragDropEffects.Copy	Kopyalama Yapar
DragDropEffects.Link	Link oluşturur
DragDropEffects.Move	Taşıma yapar
DragDropEffects.None	Herhangi bir işlem olmaz

## DRAG-DROP (SÜRÜKLE-BIRAK)\_devamı

e.KeyState değeri ile Mouse'ın hangi tuşuna basılı olduğunda işlemin yapılacağı kontrol edilir (istenirse shift,ctrl vs.)

### e.KeyState alabileceği Değerler

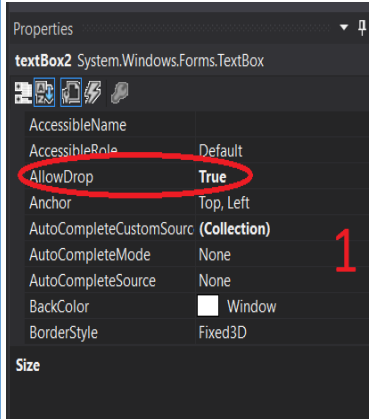
1	Mouse'un sol tuşu basılı
2	Mouse'un sağ tuşu basılı
5	Shift tuşu basılı
9	Ctrl tuşu basılı



textBox1 içindeki değeri sürükleyerek  
textBox2'ye taşıyalım

## DRAG-DROP (SÜRÜKLE-BIRAK)\_devamı

kaynak textBox1'de iken  
Mouse'ın sol tuşuna basıldığında DragDropEffects.All ile (tüm kopyala,Taşıma vs..)



```
private void textBox1_MouseDown(object sender, MouseEventArgs e)
{
    DragDropEffects d;
    if (e.Button == MouseButtons.Left)
    {
        d = textBox1.DoDragDrop(textBox1.Text, DragDropEffects.All);
        if (d == DragDropEffects.Move)
        {
            textBox1.Text = "";
        }
    }
}
```

A red circle highlights the method signature `private void textBox1_MouseDown(object sender, MouseEventArgs e)`. A red number '2' is next to the code block.

## DRAG-DROP (SÜRÜKLE-BIRAK)\_devamı

```
1 reference
private void textBox2_DragOver(object sender, DragEventArgs e)
{
    switch (e.KeyState)
    {
        case 1:
            e.Effect = DragDropEffects.Move;
            break;
        case 9:
            e.Effect = DragDropEffects.Copy;
            break;
        default:
            e.Effect = DragDropEffects.None;
            break;
    }
}
```

Burada sürüklemenin nasıl geldiğini, (hangi tuşa basılarak ve hiç basılmadan) saptamak için KeyState özelliğini kullanıyoruz.

3

## DRAG-DROP (SÜRÜKLE-BIRAK)\_devamı

Son olarak Hedef yani textBox2'nin kendi üzerine gelen değeri DragDrop yardımı ile kabul etmesi için aşağıdaki kod oluşturulur.

```
1 reference
private void textBox2_DragDrop(object sender, DragEventArgs e)
{
    textBox2.Text = e.Data.GetData(DataFormats.Text).ToString();
}
```

4



## CheckBox

Checked özelliği

Form1

SABİT MAAŞ

YABANCI DİL PRİMİ

GENEL TOPLAM

Bildiği Yabancı Diller

- ☒ İngilizce
- ☐ Fransızca
- ☐ Almanca
- ☒ Arapça

HESAPLA

```
1 reference
private void button1_Click(object sender, EventArgs e)
{
    int a = 0; float maas = 0; float prim = 0;
    maas = Convert.ToSingle(textBox1.Text);

    if (checkBox1.Checked==true)
    {
        a += 15;
    }

    if (checkBox2.Checked==true)
    {
        a += 10;
    }

    if (checkBox3.Checked==true)
    {
        a += 5;
    }

    if (checkBox4.Checked==true)
    {
        a += 15;
    }

    prim = maas * a / 100;
}
```

ked özelliği