

## Ders5:

### Menüler, Alt yordamlar, renk ve yazı tipi pencereleri (fontDialog ve colorDialog)

- ✓ Menüler (Menus and Toolbars → Menustrip)
- ✓ Menü kısa yol oluşturma
- ✓ Context MenuStrip – Kolay menü (Menus and Toolbars → contextMenuStrip)
- ✓ Alt yordamlar
- ✓ Fonksiyonlar
- ✓ Yazı Tipi Diyalog Kutusu (Font Dialog)
- ✓ Renk Diyalog Kutusu (Color Dialog)
- ✓ About Box

#### Menüler:

- Birçok uygulamada kullanılan benzer menü görünümü
- File, Edit
  - Dosya, metin düzeni işlemleri
- View, Window
  - Paneller, çalışma sayfaları görünümleri
- Project, Build, Debug
  - Proje, derleme ve hata ayıklama işlemleri
- Tools, Help
  - Yardımcı araçlar, yardım seçenekleri

Visual Studio menüleri birçok uygulamanın menülerine benzer niteliktedir. Menü isimlerinde, belirli bir harfinin altı çizilmiştir. Belirtilen harflere ALT tuşu ile birlikte basıldığında, o menülere kısayolla ulaşılır. Menü komutlarının bazılarında ise, sadece o komuta özel bir kısayol tanımlıdır. Bu kısayollar CTRL veya SHIFT gibi birkaç tuş kombinasyonu ile gerçekleşir.

#### ContextMenuStrip:

Açılan menülere alternatif olarak hazırlanan kısa yol menüleri çok kullanışlıdır. Genelde komutun hangi menüde olduğunu bilmediğimiz durumda işimize yaramaktadır. Visual Basic projelerindeki kısayol menülere ContextMenuStrip adı verilmektedir. Kısa yol menüsünü projeye eklemek için ToolBox' ta ContextMenuStrip seçilir ve forma eklenir. Eklemeyi yaptıktan sonra Text Here yazan yere komutları ekleyebiliriz. ContextMenu kontrolü sağ fare tuşu ile açılan menüler oluşturmak için kullanılır.

### Altyordam ve Fonsiyonlar:

Uygulama geliştirirken, bir işlemin birçok yerde kullanıldığı zamanlar olur. Bu gibi durumlarda bir kere yazılan kodlar, farklı yerlerde tekrar yazılır. Uygulama üzerinde bir değişiklik yapılmak istenirse, tekrar yazılan kodların tek tek bulunup değiştirilmesi gerekir. Böylece hem uygulamanın yazımı zorlaşır, hem de değişiklik yapmak giderek imkânsız hale gelir. Bu problemler, birçok yerde yapılması istenen işlemlerin fonksiyonlar ve yordamlar içinde yazılması ile çözülür. Sadece fonksiyon ve yordamların isimleri kullanılarak, istenen yerlerde kodlar çalıştırılır.

Yapılan işlemin sonucunda oluşan değer isteniyorsa fonksiyonlar kullanılır. Eğer yapılan işlemlerin sonunda bir değer döndürülüyorsa yordamlar kullanılır.

## Sub

- Dönüş değeri olmayan kod bloklarıdır.
- Birçok yerde kullanılacak kodlar, yordamlar ile gruplanmalıdır.

```
Sub Temizle()  
    Label1.Text = ""  
    ListBox1.Items.Clear()  
End Sub
```

Sub yordamları dönüş değeri olmayan kod bloklarıdır. Bu kodlar Sub ve End Sub ifadeleri arasına yazılır.

### **Sub YordamIsmi([Giriş\_Parametreleri[As Tipi]]) End Sub**

Uygulama içinde birçok yerde çalışacak olan kodlar Sub yordamı içinde yazılır. Bu kodlar, içine yazıldıkları yordamın ismi ile çağırılarak, istenen yerde tekrar çalıştırılabilir. Örneğin, bir uygulama başlarken form üzerindeki kontrollerin temizlenmesi gerekiyorsa, bu kodları bir daha yazmamak için yordam kullanılabilir.

## Function

- İşlem yapıldıktan sonra değer döndürülür.
- Return ifadesinden sonraki kodlar çalıştırılmaz.

```
Function KontrolOk() As Boolean
    If TextBox1.Text.Length > 0
        And ComboBox1.SelectedIndex > -1 Then
            Return False
        End If
        Return True
    End Function
```

Fonksiyonlar bir işlem yaptıktan sonra geriye değer döndürürler. Örneğin, bir çarpma fonksiyonunun dönüş değeri, parametre olarak verilen iki sayının çarpımı olacaktır. Fonksiyonların tanımları değişkenler gibidir.

### Function Fonksiyon(ByVal Param1 As VeriTipi, ...) As DönüşVeriTipi End Function

Fonksiyonların geriye dönüş değerleri Return ifadesi ile ya da fonksiyonun ismi verilerek yapılır.

#### Yazı Tipi Diyalog Kutusu (FontDialog):

FontDialog kontrolü , geliştireceğimiz uygulamalarda yazı tipini belirlememizi sağlayan bir kontroldür. Toolbox üzerinden seçtiğimiz FontDialog kontrolünü, form üstüne bıraksak bile, formun altında görüntülenecektir. Form üzerine buton, textbox ve fontdialog kutusu ekledikten sonra kod penceresini aşağıdaki gibi düzenleyelim.

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal
    FontDialog1.ShowDialog()
    TextBox1.Font = FontDialog1.Font
End Sub
```

### Renk Diyalog Kutusu (ColorDialog):

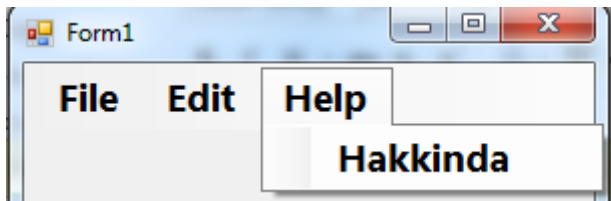
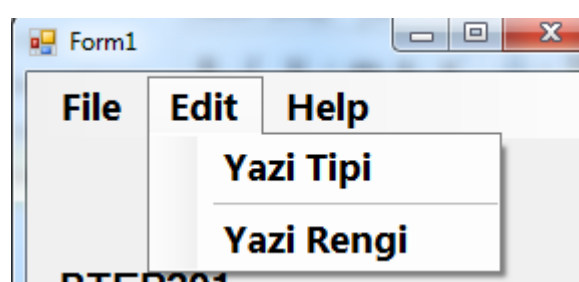
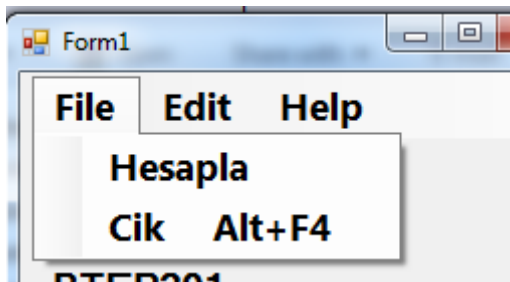
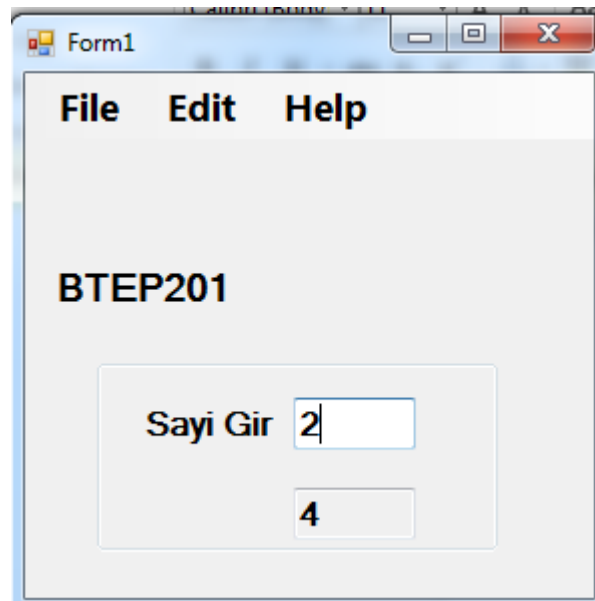
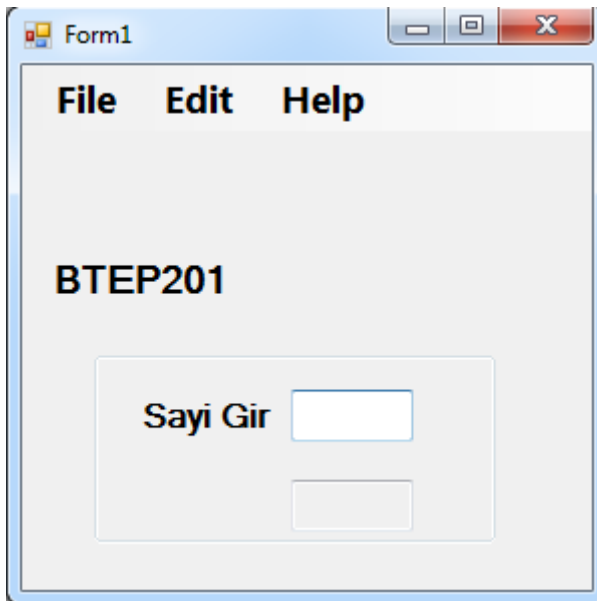
ColorDialog kontrolü , geliřtirdiđimiz uygulamalarda kullanabileceđimiz yazı rengini belirlememizi sađlayan bir kontroldür. Belirlediđimiz rengi kontrol özelliklerinde (buton rengi, formun zemin rengi, vb. gibi) kullanabileceđimiz gibi, grafik uygulamalarında da kullanabiliriz. Bu diyalog kontrolü de diđer kontroller gibi form altında görüntülenir.

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs)
    ColorDialog1.ShowDialog()
    TextBox1.ForeColor = ColorDialog1.Color
End Sub
```

### AboutBox:

Project menüsünden Add New Item seçeneđi ile eklenir.

Uygulama:



```
Public Class Form1
    Private Sub CikToolStripMenuItem_Click(sender As Object, e As EventArgs) Handles
CikToolStripMenuItem.Click
        Close()
    End Sub

    Private Sub HakkindaToolStripMenuItem_Click(sender As Object, e As EventArgs)
Handles HakkindaToolStripMenuItem.Click
        AboutBox1.Show()
    End Sub

    Private Sub YaziTipiToolStripMenuItem_Click(sender As Object, e As EventArgs)
Handles YaziTipiToolStripMenuItem.Click
        FontDialog1.ShowDialog()
        Label2.Font = FontDialog1.Font
    End Sub

    Private Sub YaziRengiToolStripMenuItem_Click(sender As Object, e As EventArgs)
Handles YaziRengiToolStripMenuItem.Click
```

```

        ColorDialog1.ShowDialog()
        Me.ForeColor = ColorDialog1.Color
    End Sub

    Private Sub HesaplaToolStripMenuItem_Click(sender As Object, e As EventArgs)
Handles HesaplaToolStripMenuItem.Click
        Dim sayi, cevap As Integer
        sayi = Integer.Parse(sayiTextBox.Text)
        cevap = Hesaplama(sayi)
        cevapTextBox.Text = cevap
        deneme()
    End Sub

    Private Sub deneme()
        'alt yordam
        MessageBox.Show("deneme")
    End Sub

    Private Function Hesaplama(ByVal sayi As Integer) As Integer
        'fonksiyon
        Return sayi * sayi
    End Function

    Private Sub YazıTipiToolStripMenuItem1_Click(sender As Object, e As EventArgs)
Handles YazıTipiToolStripMenuItem1.Click
        'contextmenustrip
        YazıTipiToolStripMenuItem_Click(sender, e)
    End Sub

    Private Sub HesaplamaToolStripMenuItem_Click(sender As Object, e As EventArgs)
Handles HesaplamaToolStripMenuItem.Click
        'contextmenustrip
        HesaplaToolStripMenuItem_Click(sender, e)
    End Sub
End Class

```