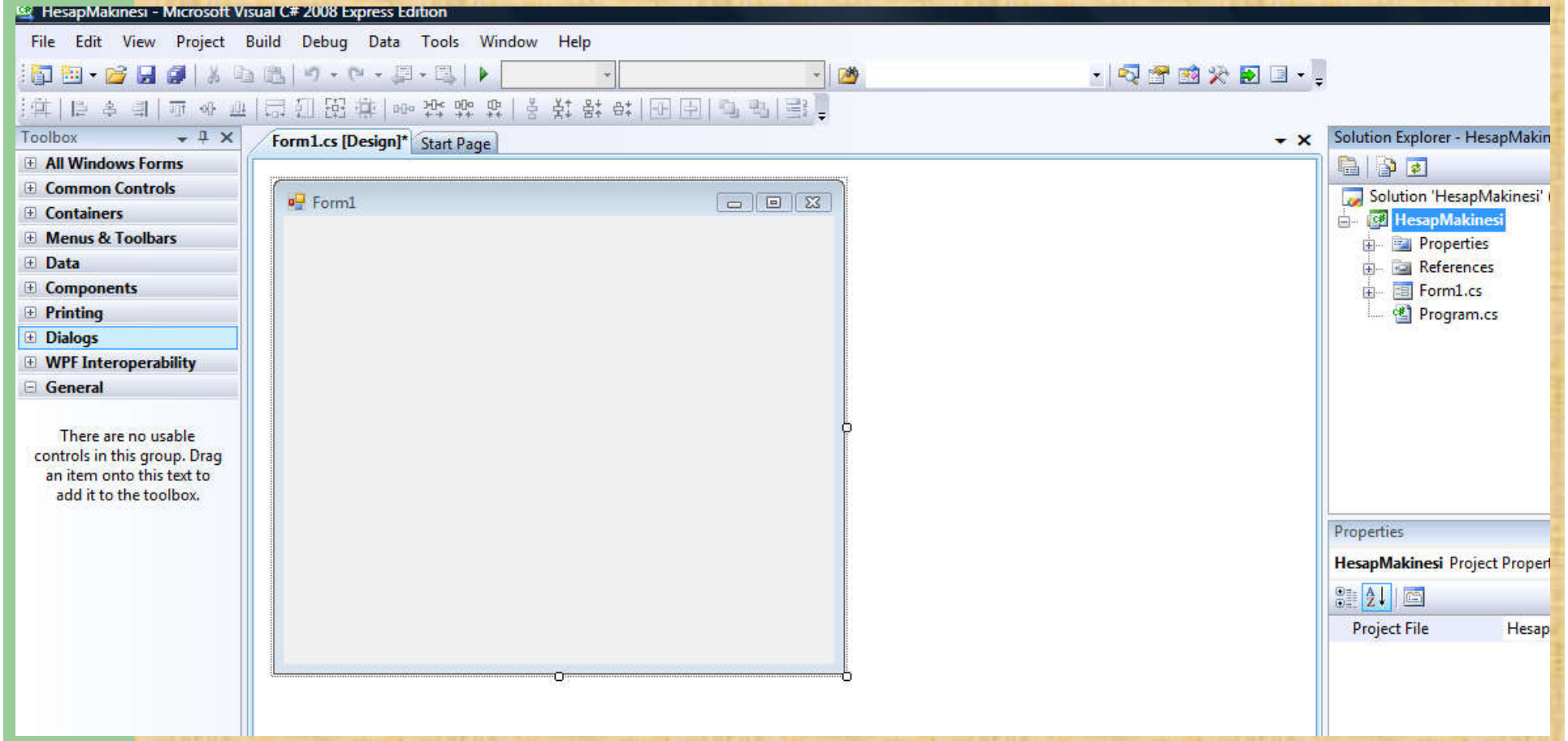
C# Windows Bileşenleri Giriş

- *Editör ekranı*
- *Form tasarımı*
- *Form özellikleri ve olayları*
- *Kontroller/Bileşenler; eklenmesi, özellikleri ve değiştirilmesi*
- *MDI formlarla çalışma, veri transferi*
- *DataGridview nesnesi*
- *Basit Hesap makinesi,Notepad uygulaması*

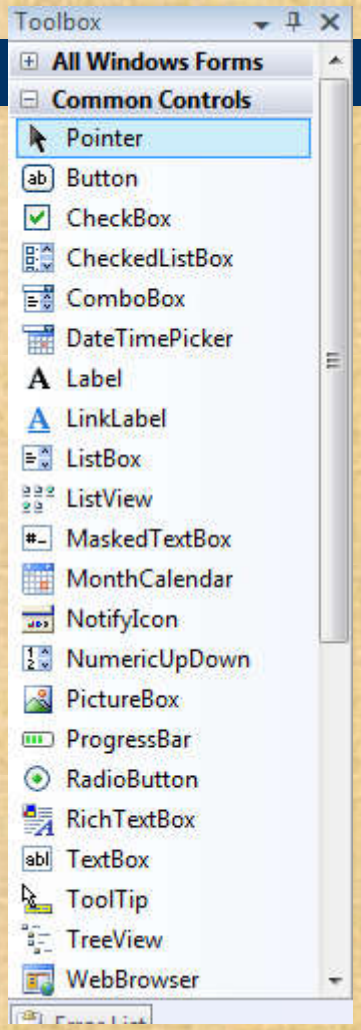
C# Windows Bileşenleri Giriş



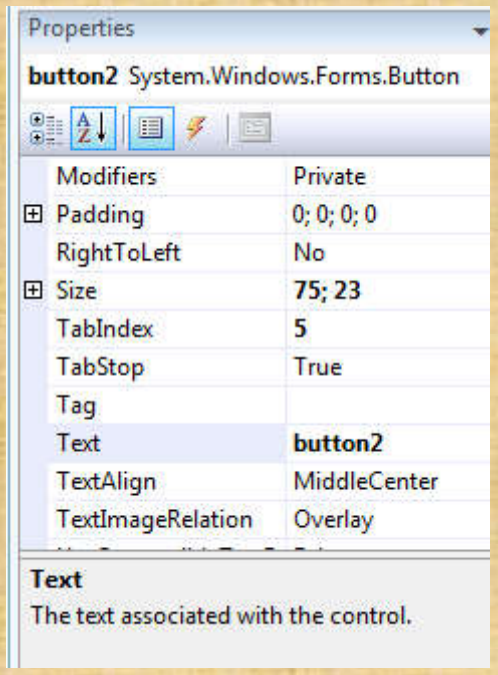
Visual Studio Ekranı

C# Windows Bileşenleri Giriş

- ***Form tasarımı***



Toolbox: Windows bileşenlerini sürükle bırak yada çift tıklama ile forma aktarırız



Properties(F4): Seçili bileşen ile ilgili özellikler ve olayları(event) belirlediğimiz penceredir.

C# Windows Bileşenleri Giriş

- ***Properties(Özellikler) ve Events(Olaylar)***

Bileşenlerin bazı genel özellikleri(properties)

Name - Bir bileşenin değişken ismi

Text - Bir bileşenin içindeki bilgiyi/etiketi tutan özellik

Enabled - Bir bileşeni aktif/pasif yapar

Visible - Bir bileşeni gösterir/gizler

BackColor - bileşenin arkaplan rengi

ForeColor- bileşenin önyüz(yazı) rengi

Font - bileşenin yazıtipi ayarlarını yapar

C# Windows Bileşenleri Giriş

- ***Properties(Özellikler)***

Multiline- textbox bileşenini çoklu satır yapar

PasswordChar-textbox için şifre karakteri belirler

Image- bileşen üzerinde resim görüntüler.

ImageAlign- Image ile görüntülenen resmi bileşen üzerinde hizalar

Opacity - Form nesnesini şeffaf kılar.(Kod ile 0-1 arası değer alır)

ControlBox- Form için minimize,maximize,close butonlarını iptal eder

Location - bileşenin konumunu x,y koordinatı olarak belirler.

Dock - Bileşenin bulunduğu parent içindeki yanaşmasını/yayılmasını belirler

TabIndex - Tab tuşu ile gezinti sırasını belirler.

C# Windows Bileşenleri Giriş

- ***Olaylar (Events)***

Kontrollerler için sık kullanılan bazı olaylar (Events)

Click - bileşenin tıklanması (çalışması)

MouseHover- mouse ile bileşen üzerine gelme

MouseLeave- mouse ile bileşen üzerinden ayrılma

Enter - Bir bileşenin aktif olması(cursor üzerinde)

Keypress- klavyeden bir tuşa basıldığında çalışan olay

C# Windows Bileşenleri Giriş

- ***Olaylar (Events)***

Kontrollerler için sık kullanılan bazı olaylar (Events)

Load - Form yüklendiğinde ilk çalışan olay

FormClosed - form kapandığında çalışan olay

FormClosing- form kapanıyorken çalışan olay

Form ile ilgili fonksiyonlar;

Close()-> olayi formu kapatır

Show()-> formu gösterir

Hide() -> formu gizler

C# Windows Bileşenleri Giriş

- ***Olaylar (Events)***

Keypress için metin kutusuna sadece sayı kabul eden örnek; 47-57 arası rakamları temsil eder.Diğer durumlarda hata mesajı çıkar. *e.handled* özelliği ile tuşun basılması engellenir.

```
if ((int)e.KeyChar >= 47 && (int)e.KeyChar <= 57)
            e.Handled = false;
              else
              {
          MessageBox.Show("sadece sayısal değer");
                  e.Handled = true;
              }
```

C# Windows Bileşenleri Giriş

- ***Olaylar (Events)***

Keypress için form için çıkış tuşu belirleme.

NOT: form keypreview özelliği =true olmalı

```
private void Form1_KeyDown(object sender, KeyEventArgs e)
        {
            if (e.KeyCode == Keys.F10)
                Close();
        }
```

Keydown için form arka planını değiştirme.

```
 private void Form1_KeyDown(object sender, KeyEventArgs e)
         { Random r=new Random();
  if (e.Control && e.Alt && e.KeyCode == Keys.R)
             {
 this.BackColor = Color.FromArgb(r.Next(0, 255), r.Next(0, 255), r.Next(0,
255));
           } }
```

C# Windows Bileşenleri Giriş

- ***Olaylar (Events)***

Keypress için 2 ayrı örnek kullanım***;***

```
if (e.KeyCode == Keys.Enter)
{
textBox2.Focus();
}
-------------------------------------------------
private void textBox1_KeyPress(object sender,
KeyPressEventArgs e)
{if ((int)e.KeyChar >= 47 && (int)e.KeyChar <= 57)
{
e.Handled = false;//eğer 47 -58 arasındaysa tuşu yazdır.
} }
```

C# Windows Bileşenleri Giriş

- ***Olaylar (Events)***

MouseDown/MouseUp için örnek kullanım;

```
private void MainForm_MouseDown(object sender,
MouseEventArgs e)
{
if (e.Button == MouseButtons.Left)
m_left = true;
if (e.Button == MouseButtons.Right) m_right = true;
}
```

C# Windows Bileşenleri Giriş

- *Olaylar (Events)*

Ctrl+Mouse click için bir örnek;

```
private void button1_MouseDown(object sender, MouseEventArgs e)
  {
Random r = new Random();
if(e.Button==MouseButtons.Left && (ModifierKeys &
Keys.Control)==Keys.Control)
button1.BackColor = Color.FromArgb(r.Next(0, 255), r.Next(0,
255), r.Next(0, 255));
  }
```

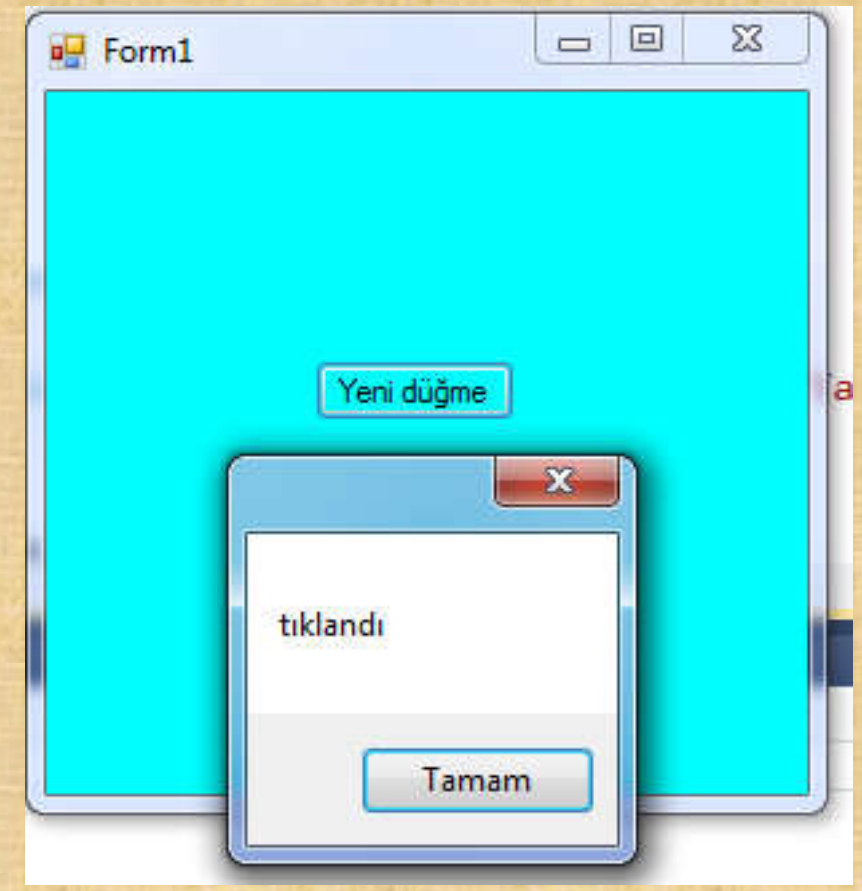
C# Windows Bileşenleri Giriş

- ***Runtime button ve Event oluşturma***

```
 public Form1()
        {
InitializeComponent();
pnl.Location = new Point(10, 10);
pnl.Dock = DockStyle.Fill;
pnl.BackColor = System.Drawing.Color.Aqua;
this.Controls.Add(pnl);
Button b1 = new Button();//Buton sınıfından «b1» butonu oluş.
b1.Location = new Point(100, 100);//butonu 100,100'e yerleştirdir
b1.Text = "Yeni düğme";//butona etiket ekledik
pnl.Controls.Add(b1); //butonu panel'e ekledik
b1.Click+=new EventHandler(b1_Click);//buton tıklandığında
çağrılacak eventhandler belirlendi.
        }
P       rivate void b1_Click(object sender,System.EventArgs e)
     {
      MessageBox.Show("tıklandı");
     }    } //Form1 sonu
```

C# Windows Bileşenleri Giriş

- *Runtime button ve Event oluşturma (devam)*

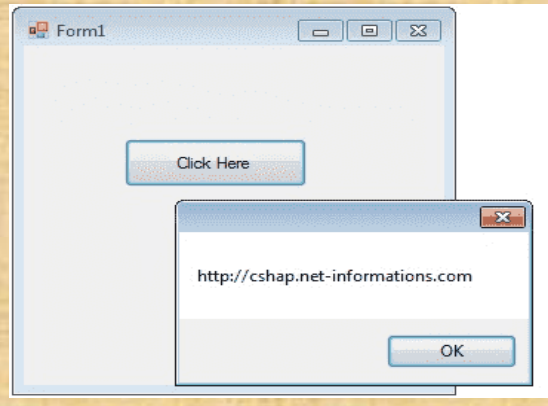


C# Windows Bileşenleri Giriş

- ***Windows Bileşenleri (Components/Controllers)***

C# Button Control

Windows Forms controls are reusable components that encapsulate user interface functionality and are used in client side Windows applications. A button is a control, which is an interactive component that enables users to communicate with an application. The Button class inherits directly from the ButtonBase class. A Button can be clicked by using the mouse, ENTER key, or SPACEBAR if the button has focus.



C# Windows Bileşenleri Giriş

- ***Windows Bileşenleri (Components/Controllers)***

When you want to change display text of the Button , you can change the Text property of the button.

button1.Text = "Click Here";

Similarly if you want to load an Image to a Button control , you can code like this

button1.Image = Image.FromFile("C:\\testimage.jpg");

The following C# source code shows how to change the button Text property while Form loading event and to display a message box when pressing a Button Control.

http://csharp.net-informations.com/gui/button.htm

C# Windows Bileşenleri Giriş

- ***Windows Bileşenleri (Components/Controllers)***

C# Label Control

Label controls can also be used to add descriptive text to a Form to provide the user with helpful information. The Label class is defined in the System.Windows.Forms namespace.

Add a Label control to the form - Click Label in the Toolbox and drag it over the forms Designer and drop it in the desired location.

If you want to change the display text of the Label, you have to set a new text to the Text property of Label.

```
label1.Text = "This is my first Label";
```

In addition to displaying text, the Label control can also display an image using the Image property, or a combination of the ImageIndex and ImageList properties.

```
label1.Image = Image.FromFile("C:\\testimage.jpg");
```

C# Windows Bileşenleri Giriş

- ***Windows Bileşenleri (Components/Controllers)***

C# TextBox Control

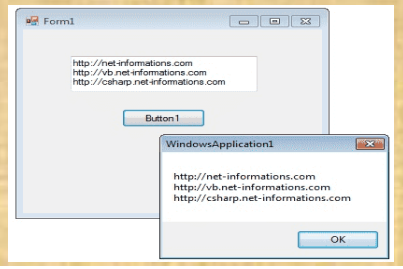
A TextBox control is used to display, or accept as input, a single line of text. This control has additional functionality that is not found in the standard Windows text box control, including multiline editing and password character masking.

For displaying a text in a TextBox control , you can code like this

textBox1.Text = "http://csharp.net-informations.com";

You can also collect the input value from a TextBox control to a variable like this way

string var; var = textBox1.Text;

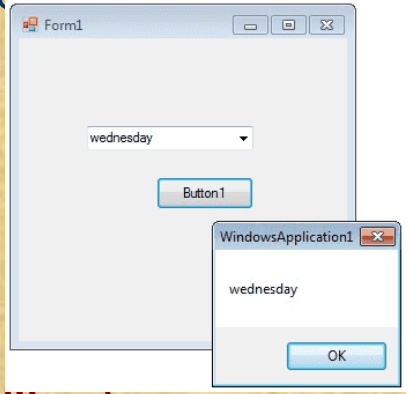


C# Windows Bileşenleri Giriş

- *Windows Bileşenleri (Components/Controllers)*

C# ComboBox Control

You can add individual objects with the Add method. You can delete items with the Remove method or clear the entire list with the Clear method.



Add item to combo box

```
comboBox1.Items.Add("Sunday");
```

How to retrieve value from ComboBox

If you want to retrieve the displayed item to a string

```
string var;  var = comboBox1.Text;
```

Or

```
var item = this.comboBox1.GetItemText(this.comboBox1.SelectedItem);
MessageBox.Show(item);
```

How to remove an item from ComboBox

You can remove items from a combobox in two ways. You can remove item at a the specified index or giving a specified item by name.

```
comboBox1.Items.RemoveAt(1);
```

The above code will remove the second item from the combobox.

```
comboBox1.Items.Remove("Friday");
```

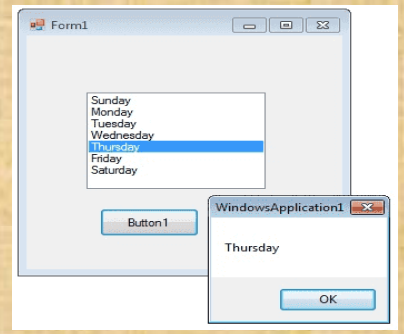
C# Windows Bileşenleri Giriş

- ***Windows Bileşenleri (Components/Controllers)***

C# ListBox Control

The ListBox control enables you to display a list of items to the user that the user can select by clicking.

In addition to display and selection functionality, the ListBox also provides features that enable you to efficiently add items to the ListBox and to find text within the items of the list. You can use the Add or Insert method to add items to a list box. The Add method adds new items at the end of an unsorted list box.




```
listBox1.Items.Add("Sunday");
```

If you want to retrieve a single selected item to a variable , you can code like this

```
string var;
var = listBox1.Text;
```

The SelectionMode property determines how many items in the list can be selected at a time.

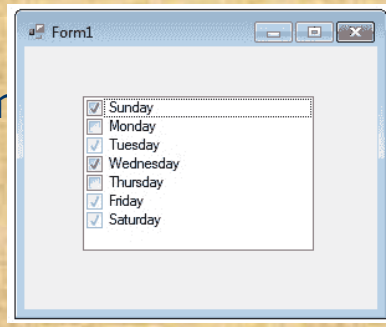
```
listBox1.SelectionMode = SelectionMode.MultiSimple;
```

C# Windows Bileşenleri Giriş

- ***Windows Bileşenleri (Components/Controllers)***

C# Checked ListBox Control

The user can place a check mark by one or more items and the checked items can be navigated with the CheckedListBox.CheckedItemCollection and CheckedListBox.CheckedIndexCollection. To add objects to the time, assign an array of object references with the AddRange m then displays the default string value for each object.




```
String Days= {"Sunday", "Monday", "Tuesday"};
checkedListBox1.Items.AddRange(days);
```

You can add individual items to the list with the Add method. The CheckedListBox object supports three states through the CheckState enumeration: Checked, Indeterminate, and Unchecked.

```
checkedListBox1.Items.Add("Sunday", CheckState.Checked); checkedListBox1.Items.Add("Monday",
CheckState.Unchecked);
checkedListBox1.Items.Add("Tuesday", CheckState.Indeterminate);
```

C# Windows Bileşenleri Giriş

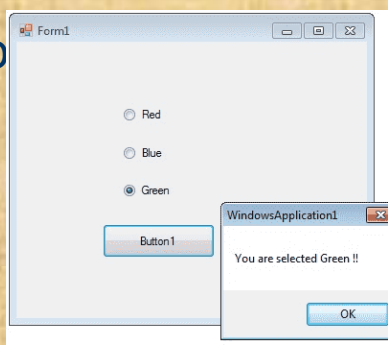
- ***Windows Bileşenleri (Components/Controllers)***

C# RadioButton Control

A radio button or option button enables the user to select a single option from a group of choices when paired with other RadioButton controls.

When a user clicks on a radio button, it becomes checked, and all other radio buttons with same group become uncheckedThe RadioButton control can display text, an Image, or both. Use the Checked property to get or set the state of a RadioButto

radioButton1.Checked = true;



C# Windows Bileşenleri Giriş

- ***Windows Bileşenleri (Components/Controllers)***

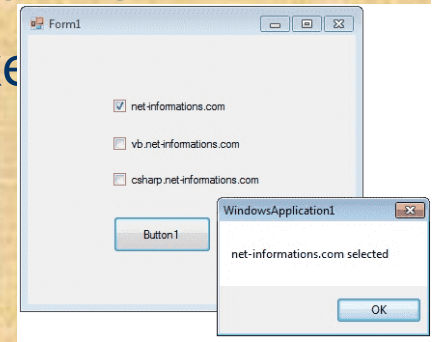
C# CheckBox Control

CheckBox to give the user an option, such as true/false or yes/no. You can click a check box to select/deselect it.

```
if (checkBox1.Checked == true) {
msg = "net-informations.com";
}
```

You can use the CheckBox control ThreeState property to direct the control to return the Checked, Unchecke Indeterminate values.

```
checkBox1.ThreeState = true;
```



C# Windows Bileşenleri Giriş

- ***Windows Bileşenleri (Components/Controllers)***

C# PictureBox Control

The Windows Forms PictureBox control is used to display images in bitmap, GIF , icon , or JPEG formats.

You can set the Image property.

```
pictureBox1.Image = Image.FromFile("c:\\testImage.jpg");
```

The SizeMode property, which is set to values in the PictureBoxSizeMode enumeration, controls the clipping and positioning of the image in the display area.

```
pictureBox1.SizeMode = PictureBoxSizeMode.StretchImage;
```

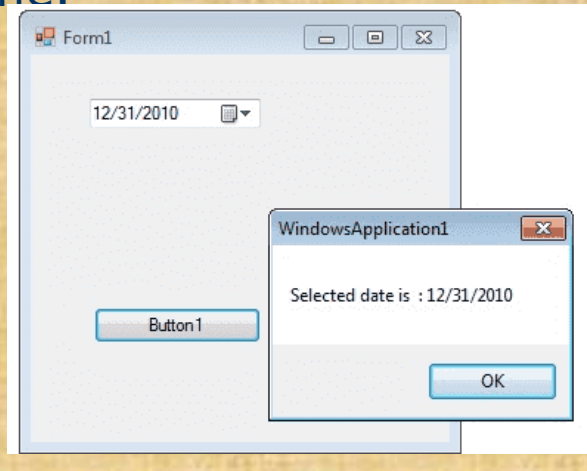
C# Windows Bileşenleri Giriş

- ***Windows Bileşenleri (Components/Controllers)***

C# DateTimePicker Control

The DateTimePicker control has two parts, a label that displays the selected date and a popup calendar that allows users to select a new date. The most important property of the DateTimePicker is the ***Value*** property, which holds the selected date and time.

```
dateTimePicker1.Value = DateTime.Today;
```



C# Windows Bileşenleri Giriş

- ***Windows Bileşenleri (Components/Controllers)***

C# DateTimePicker Control

The Value property contains the current date and time the control is set to. You can use the Text property or the appropriate member of Value to get the date and time value.

```
DateTime iDate;
iDate = dateTimePicker1.Value;
```

The control can display one of several styles, depending on its property values. The values can be displayed in four formats, which are set by the Format property: Long, Short, Time, or Custom.

```
dateTimePicker1.Format = DateTimePickerFormat.Short;
```

C# Windows Bileşenleri Giriş

- *Windows Bileşenleri (Components/Controllers)*

C# Timer Control

Timer kontroller.

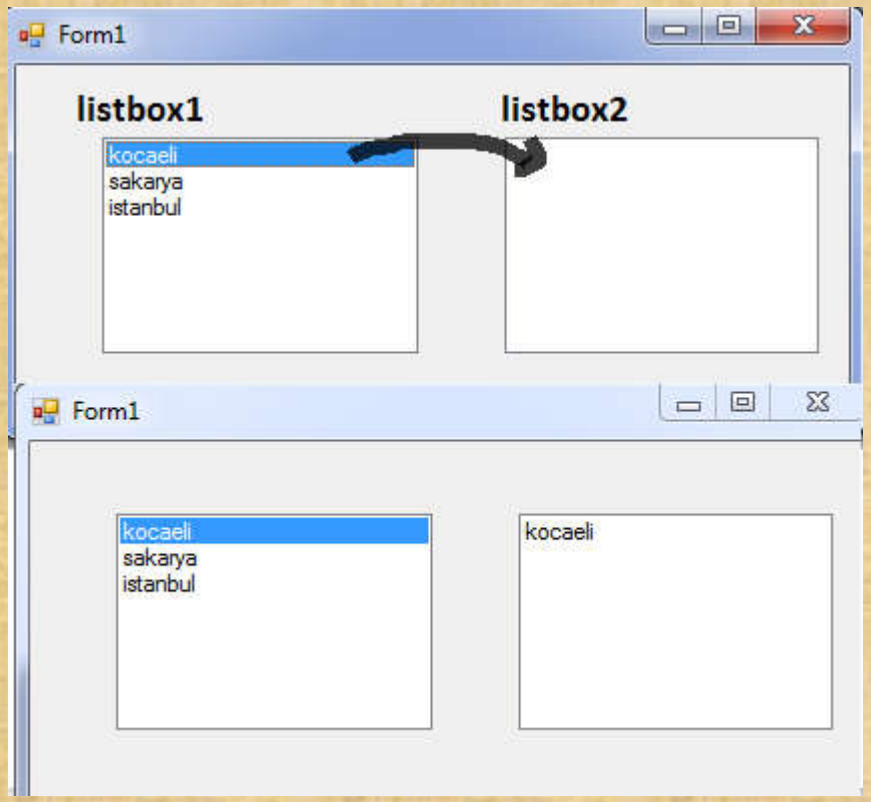
Interval – çalışma zaman aralığı

Enabled- timer'ı aktif/pasif ediyor.

Tick (Event)- timer her interval değerinde tick olayına geliyor

C# Windows Bileşenleri Giriş

- *DRAG and DROP olayı*



C# Windows Bileşenleri Giriş

- *DRAG and DROP olayı*

Listbox'lar arasında Drag Drop işlemi

1- İlk önce listbox1'de bileşen var ise **mouse sol tuşuna** basılınca (listbox1 mousedown içinde) seçilen elemanı buluyoruz. Bunun için Point sınıfından ve listbox'un IndexFromPoint özelliğinden faydalanıp koordinat olarak seçilen elemanın indeksini çekiyoruz.

```
if (listBox1.Items.Count == 0) { return; }//boşsa birşey yapma
Point n= new Point(e.X, e.Y);
int i= listBox1.IndexFromPoint(n);
listBox1.DoDragDrop(listBox1.Items[i], DragDropEffects.All);
```

C# Windows Bileşenleri Giriş

- ***DRAG and DROP olayı***

Listbox'lar arasında Drag Drop işlemi

2- listbox2'nin **AllowDrop** özelliği true olacak şekilde; **DragOver** eventine yani taşınacak hedef bileşene, **mouse sol butonu** tıklandığı kontrol edilip (keystate) taşıma efektini uygularız.

```
if (e.KeyState == 1)
                    e.Effect = DragDropEffects.All;
```

3- Son olarak sürükle bırak işlemi tamamlanınca gerçekleşen **DragDrop** eventine gerçekleşecek olayı yazıp uygularız..

```
listBox2.Items.Add(e.Data.GetData(DataFormats.Text.ToString())
);
```

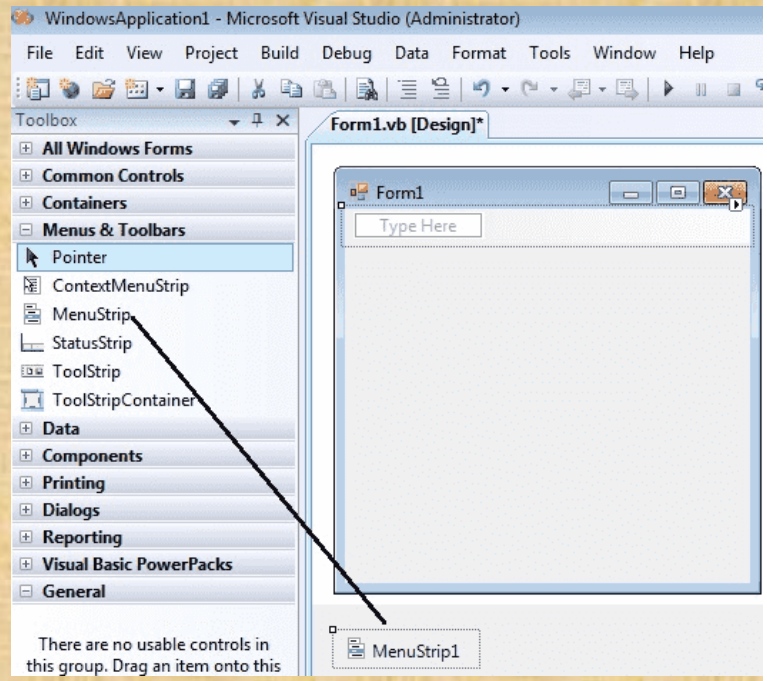
GetData metodu sürüklenen bilgiyi getiren metotdur. Sürüklenen bilginin formatını DataFormats belirler.

C# Windows Bileşenleri Giriş

- ***Windows Bileşenleri (Components/Controllers)***

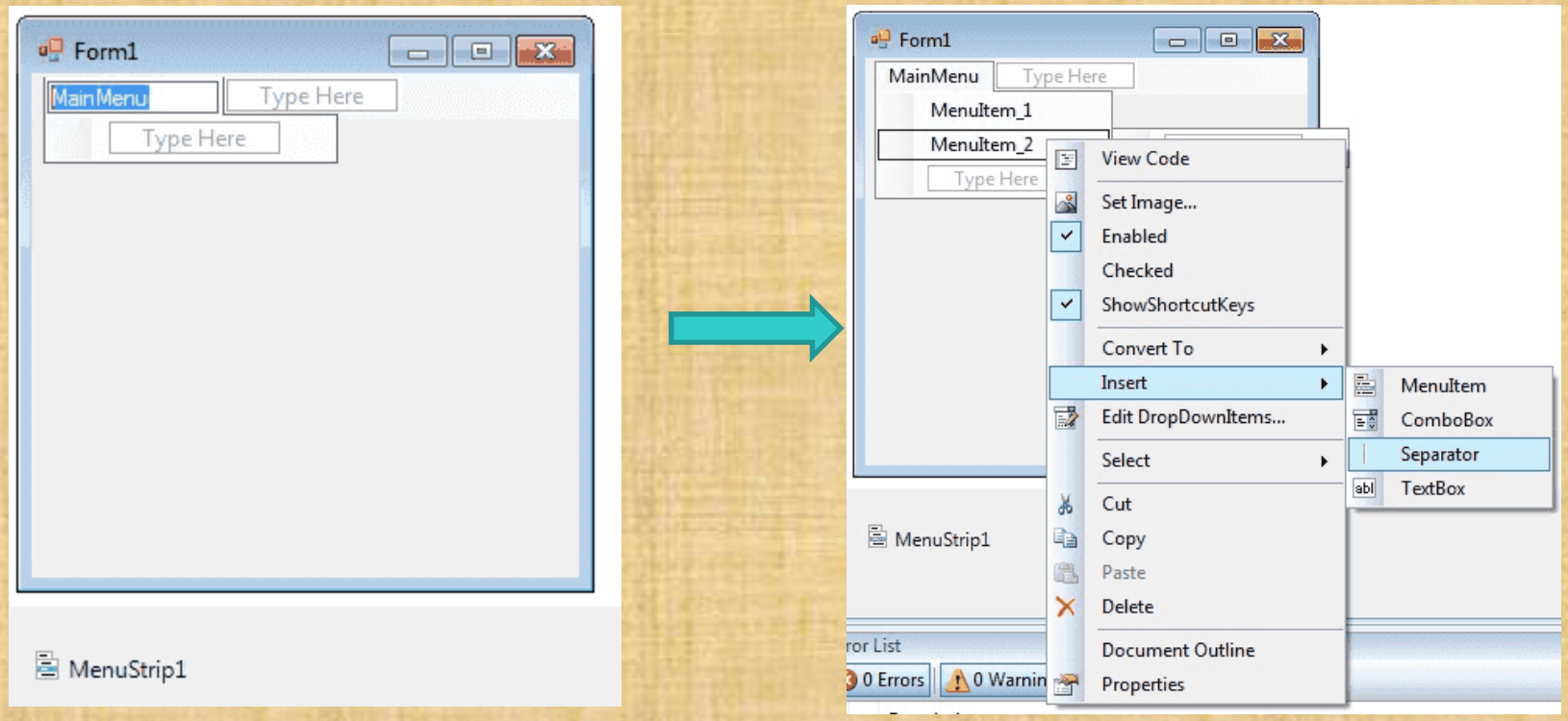
C# Menu Control

A Menu on a Windows Form is created with a MainMenu object, which is a collection of MenuItem objects. MainMenu is the container for the Menu structure of the form and menus are made of MenuItem objects that represent individual parts of a menu. You can add menus to Windows Forms at design time by adding the MainMenu component and then appending menu items to it using the Menu Designer.



C# Windows Bileşenleri Giriş

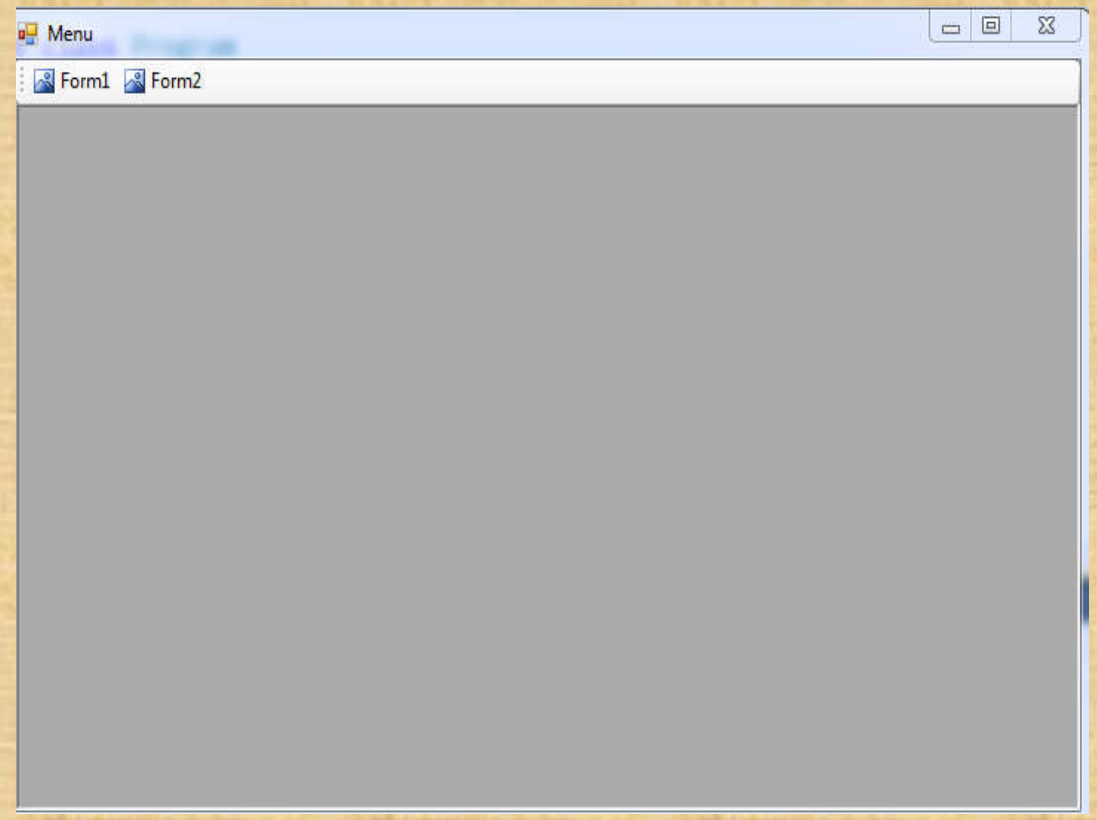
- ***Windows Bileşenleri (Components/Controllers)***



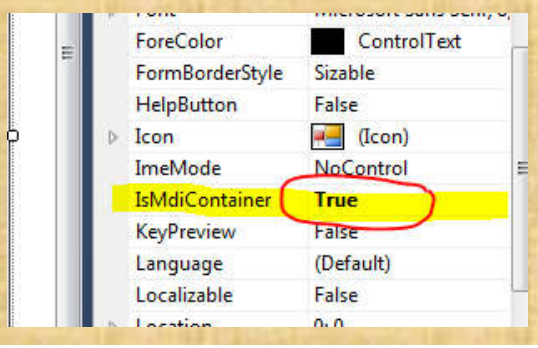
If you need a seperator bar , right click on your menu then go to insert->Seperator.

C# Windows Bileşenleri Giriş

- ***MDI Formlarla çalışma***

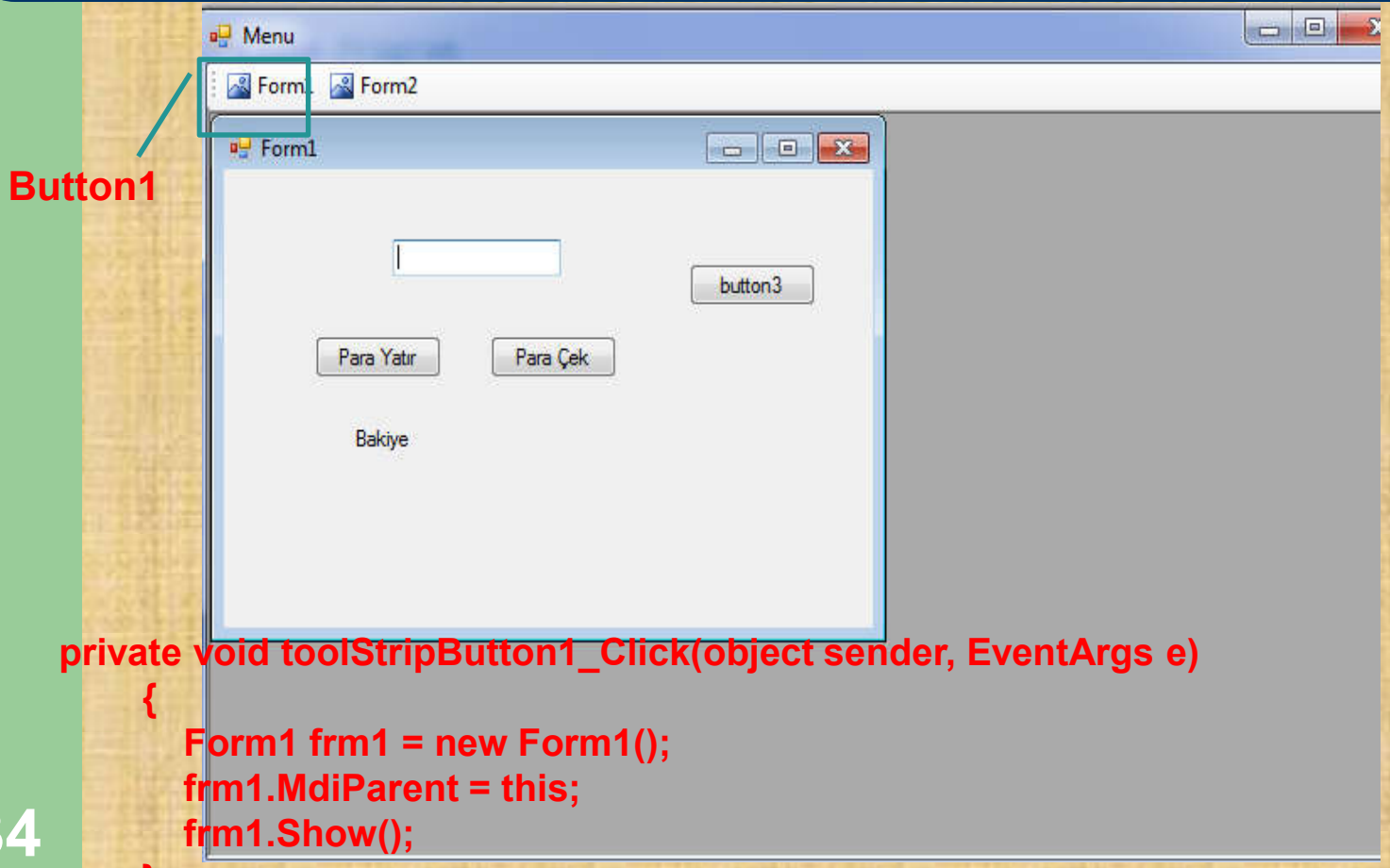


MDI(MultipleDocumentInterface-Çoklu Doküman Arayüzü) için örnekte 3 form oluşturulmuş. Burada ***menu*** isimli formun **IsMdiContainer** seçeneği true yapılarak ***menu*** formu diğer formları barındıracak parent form olmuştur.



C# Windows Bileşenleri Giriş

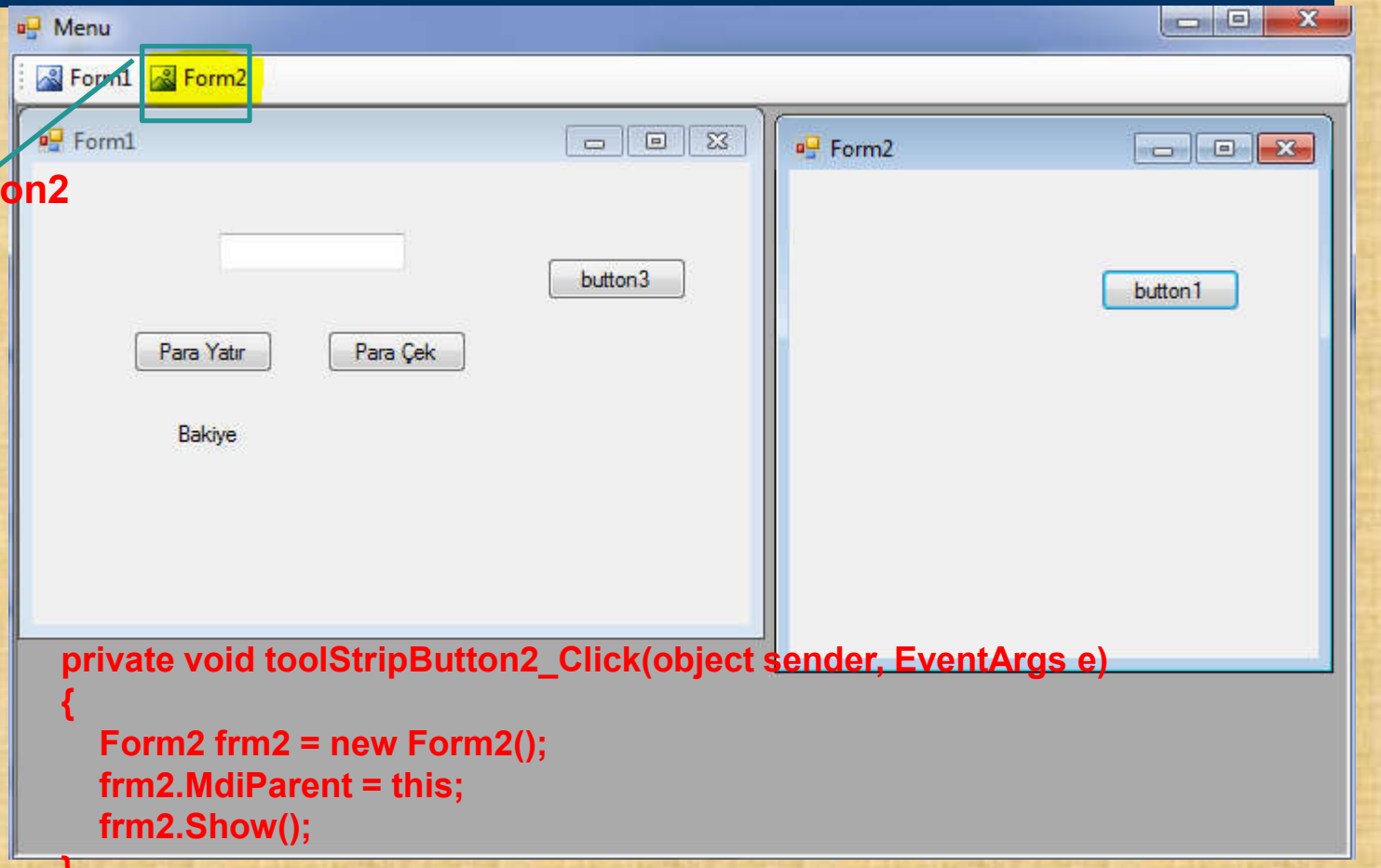
- *MDI Formlarla çalışma*




```
private void toolStripButton1_Click(object sender, EventArgs e)
    {
        Form1 frm1 = new Form1();
        frm1.MdiParent = this;
        frm1.Show();
    }
```

C# Windows Bileşenleri Giriş

- *MDI Formlarla çalışma*




```
private void toolStripButton2_Click(object sender, EventArgs e)
{
    Form2 frm2 = new Form2();
    frm2.MdiParent = this;
    frm2.Show();
}
```

C# Windows Bileşenleri Giriş

- ***MDI Formlarla çalışma***

```
private void toolStripButton1_Click(object sender, EventArgs e)
    {
        Form1 frm1 = new Form1();
        frm1.MdiParent = this;
        frm1.Show();
    }

    private void toolStripButton2_Click(object sender, EventArgs e)
    {
        Form2 frm2 = new Form2();
        frm2.MdiParent = this;
        frm2.Show();
    }
```

C# Windows Bileşenleri Giriş

- ***MDI Formlar arası veri transferi***

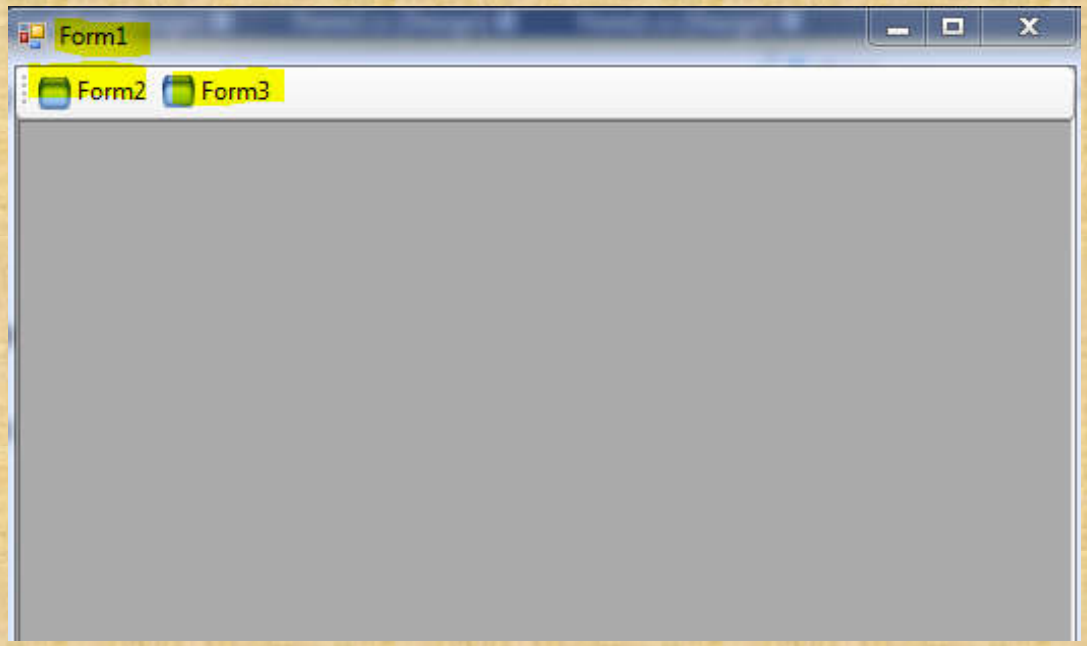
Form1,Form2 ve Form3 isimli 3 form oluşturuldu.Form1 mdiContainer olacak. Form2 içinde textbox var ve buradaki bilgi form3 içindeki label'a transfer olacak. Form3 içinde «***deger***» isimli public global bir değişken mevcut. Bunun yanında Form1 parent form olduğu için , child formlar arası geçiş için (Form2'den Form3'teki label'a veri transfer edilecek!) Form1 içinde Form3'ü referans gösteren «***frm3***» isimli global bir değişken kullanıldı.

Eğer form3 açıksa buton ile yada menüden tekrar form3'ü açmaya kalkarsa uyarıda bulunulacak. Açık olan form ya kapatılacak yada textbox aktif olduğunda form3 (frm3 null yapılıyor,close ediliyor) kapatıalcak.

C# Windows Bileşenleri Giriş

- ***MDI Formlar arası veri transferi***

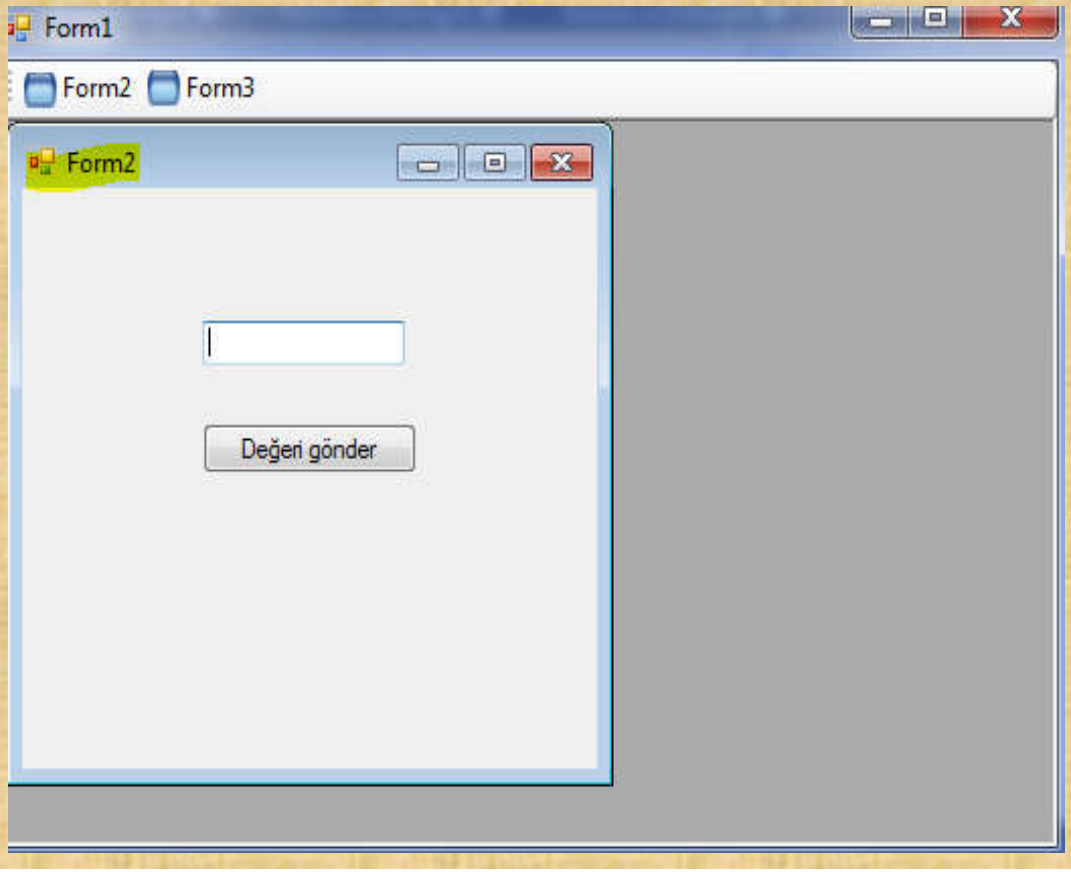
Form1 container ekranı



C# Windows Bileşenleri Giriş

- ***MDI Formlar arası veri transferi***

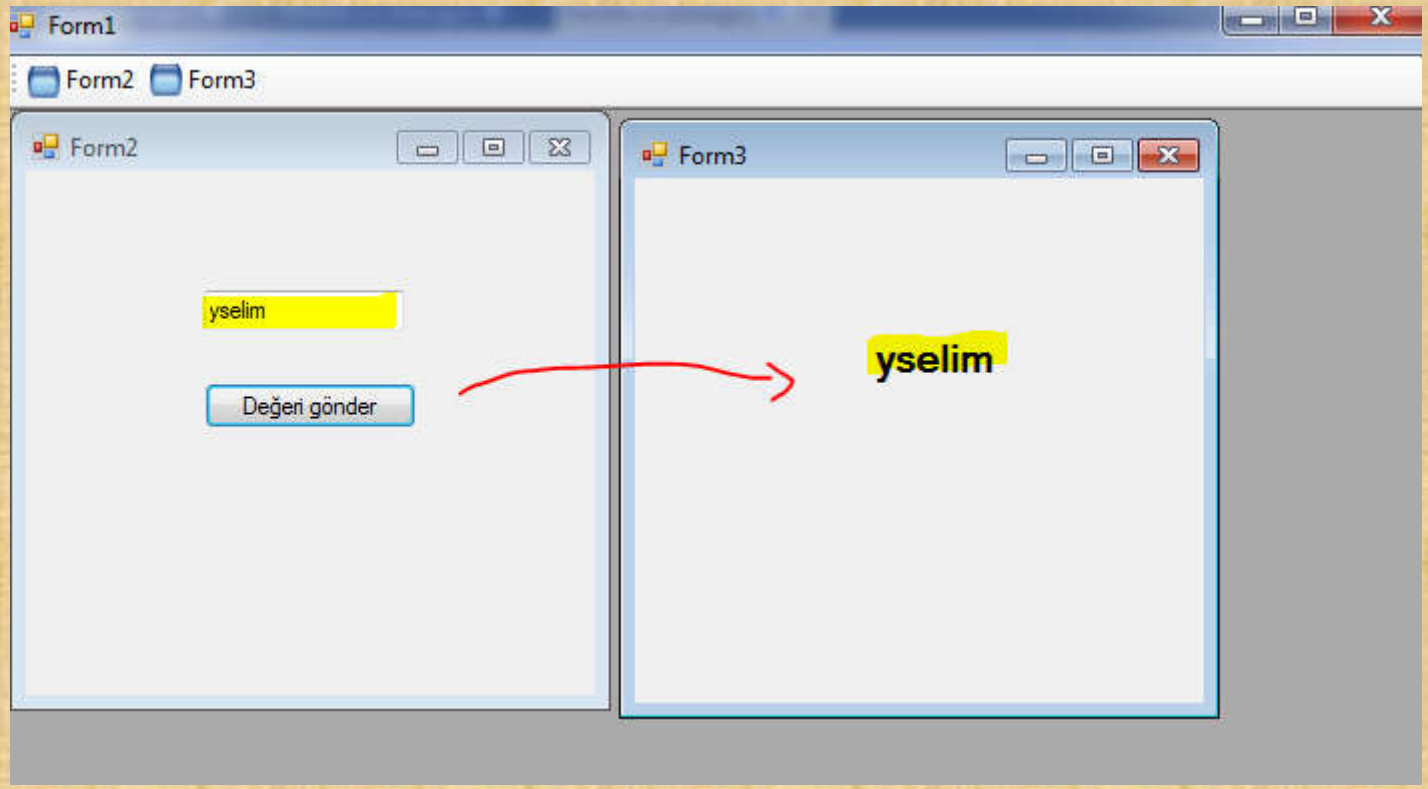
Form2 ekranı



C# Windows Bileşenleri Giriş

- ***MDI Formlar arası veri transferi***

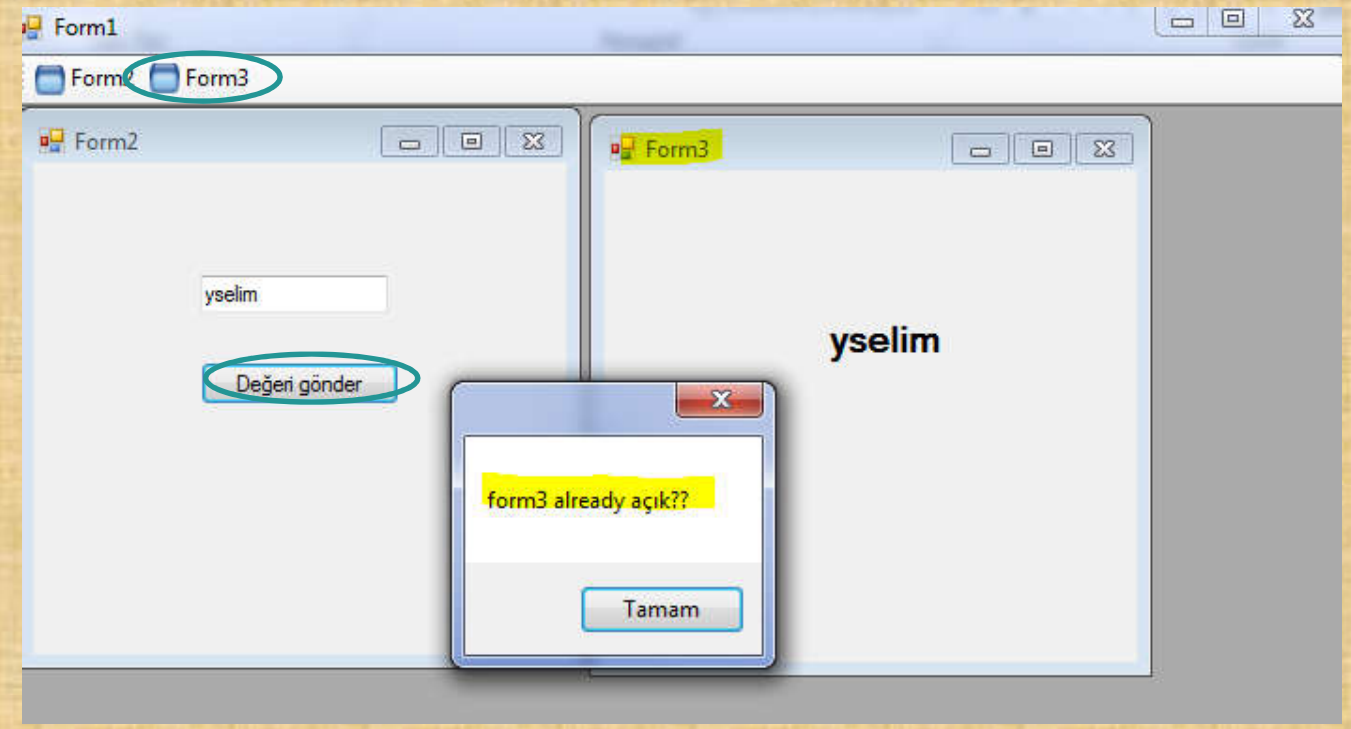
Form2 textbox'taki bilgi Form3 label'a geçti.



C# Windows Bileşenleri Giriş

- ***MDI Formlar arası veri transferi***

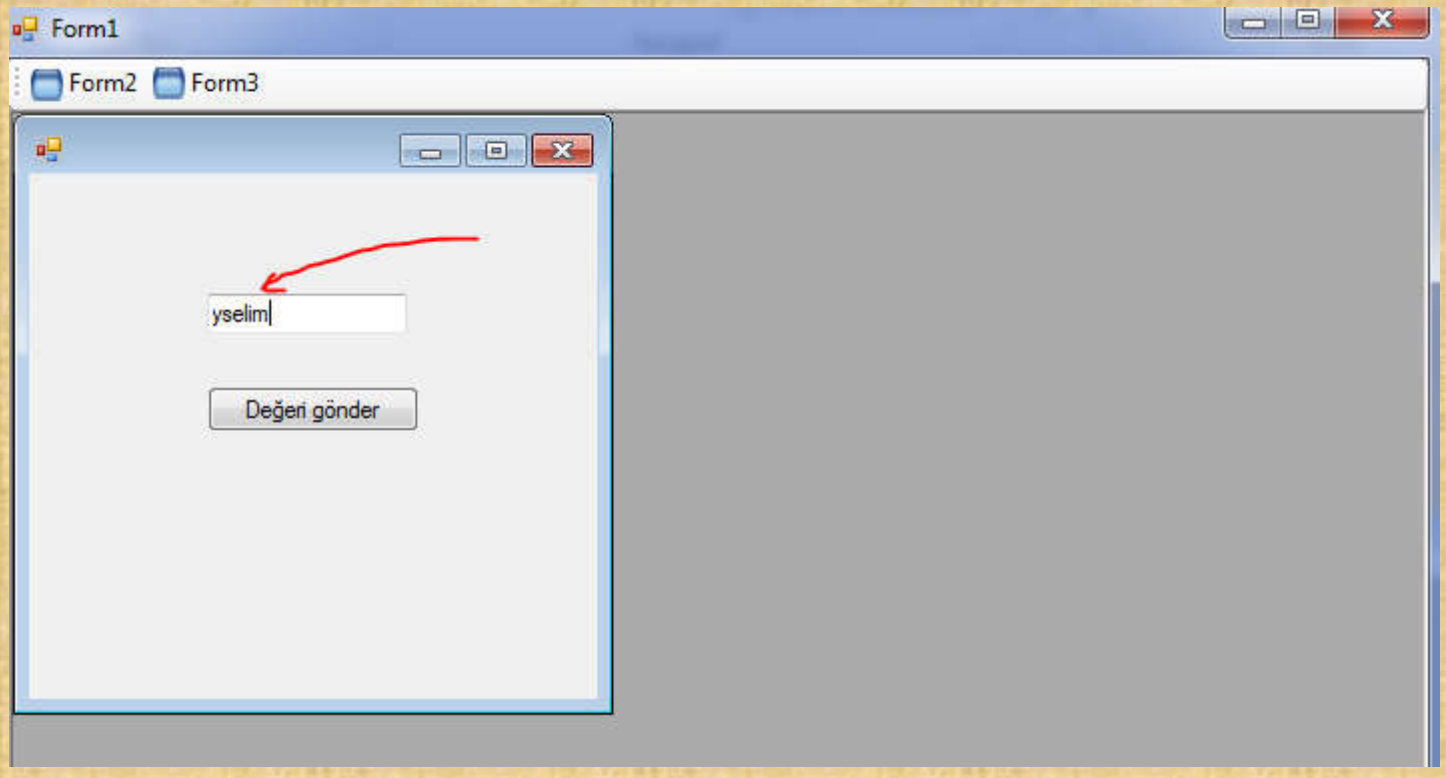
Form3 açıksa her durumda kontrolü yapılıyor. İster butona basılsın tekrar yada isterse kullanıcı yukardaki menüden Form3 seçsin. Her durum için kontrol var.



C# Windows Bileşenleri Giriş

- *MDI Formlar arası veri transferi*

Metin kutusu seçildiği anda form3 kapatılıyor. Kullanıcıda form3'ü kendi kapatabilir.



C# Windows Bileşenleri Giriş

- ***MDI Formlar arası veri transferi***

Form1 kodu;

```
public partial class Form1 : Form
   {
      public static Form3 frm3;
      public Form1()
      {
         InitializeComponent();
      }

      private void toolStripButton1_Click(object sender, EventArgs e)
      {
         Form2 frm2 = new Form2();
         frm2.MdiParent = this;
         frm2.Show();
      }
```

C# Windows Bileşenleri Giriş

- ***MDI Formlar arası veri transferi***

```
private void toolStripButton2_Click(object sender, EventArgs e)
    {

        if (frm3 == null)
        {
            frm3 = new Form3();
            frm3.MdiParent = this;
            frm3.Show();
        }
        else
            MessageBox.Show("form3 already açık??");
    }
}
```

C# Windows Bileşenleri Giriş

- ***MDI Formlar arası veri transferi***

Form2 kodu;

```
public partial class Form2 : Form
   {
       private void button1_Click(object sender, EventArgs e)
       {
if (Form1.frm3 == null)
           {
               Form1.frm3 = new Form3();
               Form3.deger = textBox1.Text;
               Form1.frm3.MdiParent = Form.ActiveForm;
               Form1.frm3.Show();
           }
           else
               MessageBox.Show("Form açık zaten");
       }
```

C# Windows Bileşenleri Giriş

- ***MDI Formlar arası veri transferi***

Form2 kodu (devamı)

```
    private void textBox1_Enter(object sender, EventArgs e)
    {
        if (textBox1.Text != "" && Form1.frm3!=null)
        {
            //Form1.frm3.Dispose();
            Form1.frm3.Close();
            Form1.frm3 = null;

            this.Text = "";
        }
    }
}
```

C# Windows Bileşenleri Giriş

- ***MDI Formlar arası veri transferi***

Form3 kodu

```
public partial class Form3 : Form
   {
      public static string deger;
      private void Form3_Load(object sender, EventArgs e)
      {
         label1.Text = deger;
      }
 private void Form3_FormClosed(object sender, FormClosedEventArgs
e)
      {
         Form1.frm3 = null;
      }
   }
```

C# Windows Bileşenleri Giriş

- *DataGridView (Auto update,CurrentRow etc.)*

```
namespace WindowsFormsApplication1
{
    public partial class Form1 : Form
    {SqlConnection baglanti;
    SqlCommand komut;
    SqlDataAdapter da;
    DataSet ds;
    bool degisim = false;
        public Form1()
        {
         baglanti = new SqlConnection("Data
Source=.\\sqlexpress;Initial
Catalog=elginkan;Integrated Security=true");
         InitializeComponent();
}
```

C# Windows Bileşenleri Giriş

- *DataGridView (Auto update,CurrentRow etc.)*

```
 private void Form1_Load(object sender, EventArgs e)
        {
            baglanti.Open();
            Listele();
}

 void Listele()
        {
          if (baglanti.State==ConnectionState.Closed)
            baglanti.Open();
     da=new SqlDataAdapter("Select * from ogrTablo",baglanti);
         ds = new DataSet();
         da.Fill(ds);
        dataGridView1.DataSource = ds.Tables[0];

        }
//listele grid'i dolduruyor.
```

C# Windows Bileşenleri Giriş

- ***DataGridView (Auto update,CurrentRow etc.)***

```
 private void dataGridView1_CellClick(object sender,
DataGridViewCellEventArgs e)
        {
            if (degisim == true)
            {
     da = new SqlDataAdapter("Select * from ogrTablo", baglanti);
            ds = new DataSet();
            da.Fill(ds);
            dataGridView1.DataSource = ds.Tables[0];
             }
     int secilenSatir = dataGridView1.SelectedCells[0].RowIndex;
             string ogrAdi =
dataGridView1.Rows[secilenSatir].Cells[1].Value.ToString();
            textBox1.Text = ogrAdi;
        }
```

//'degisim' bool değeri true ise değişiklik var ve datagrid güncelleniyor.

C# Windows Bileşenleri Giriş

- *DataGridView (Auto update,CurrentRow etc.)*

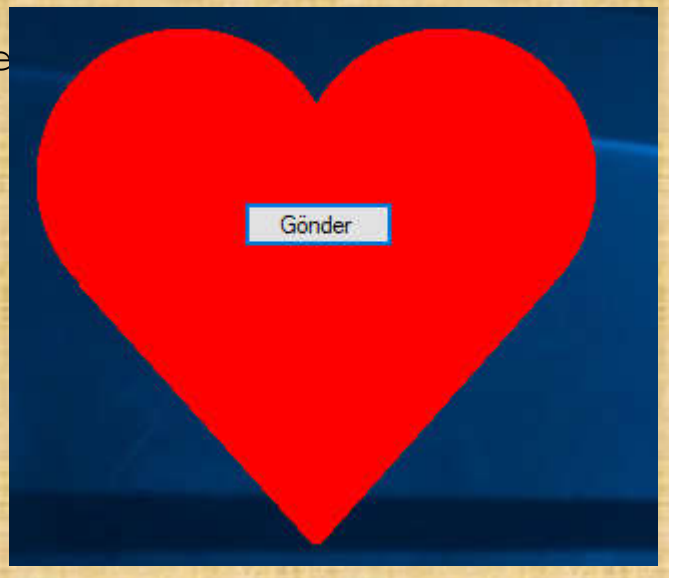
```
 private void dataGridView1_CellEndEdit(object sender,
DataGridViewCellEventArgs e)
        {
            if (baglanti.State==ConnectionState.Closed)
            baglanti.Open();
            komut = new SqlCommand("update ogrTablo set ogrAdiSoyadi='" +
dataGridView1.CurrentRow.Cells[1].Value.ToString() + "' where ogrNo='" +
dataGridView1.CurrentRow.Cells[0].Value.ToString() + "'", baglanti);
            komut.ExecuteNonQuery();
            degisim = true;
        }
    }
}
```

*//Cell edit modda olduğu için datagrid datasource'unu ds(dataset) ile besleyemiyoruz, «**SetCurrentCellAddressCore**» hatası veriyor*
*//bu yüzden '**degisim**' degiskeni tanımlandı ve kişi edit işi bitip başka bir hücreyi tıklayınca //(CellClick event'te) '**degisim**' degiskenine bakarak true ise ds gridview'in source'unu //besleyecek güncelleme olacak*

C# Windows Bileşenleri Giriş

- *Kalp şeklinde Form :)*

```
 System.Drawing.Drawing2D.GraphicsPath path =
new System.Drawing.Drawing2D.GraphicsPath();
path.AddArc(70, 10, 150, 150, 135, 195);
path.AddArc(200, 10, 150, 150, 210, 195);
path.AddLine(92, 139, 210, 270);
path.AddLine(327, 139, 210, 270);
path.AddLine(327, 139, 92, 139);
this.FormBorderStyle = FormBorderStyle.None
this.Region = new Region(path);
```



C# Windows Bileşenleri Giriş

- ***Kalp şeklinde Formu taşınma (Win API ile)***

```
/*Constants in Windows API
0x84 = WM_NCHITTEST - Mouse Capture Test
0x1 = HTCLIENT - Application Client Area
0x2 = HTCAPTION - Application Title Bar
```

```
This function intercepts all the commands sent to the application.
It checks to see of the message is a mouse click in the application.
It passes the action to the base action by default. It reassigns
the action to the title bar if it occured in the client area
to allow the drag and move behavior.
*/
    protected override void WndProc(ref Message m)
       {
           switch (m.Msg)
           {
               case 0x84:
                   base.WndProc(ref m);
                   if ((int)m.Result == 0x1)
                       m.Result = (IntPtr)0x2;
                   return;
           }

           base.WndProc(ref m);
       }
```