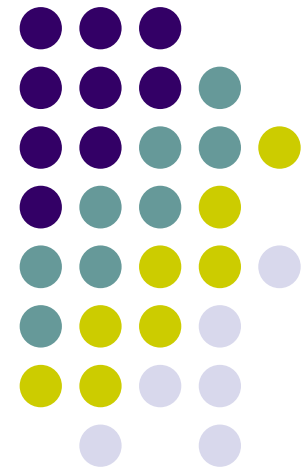
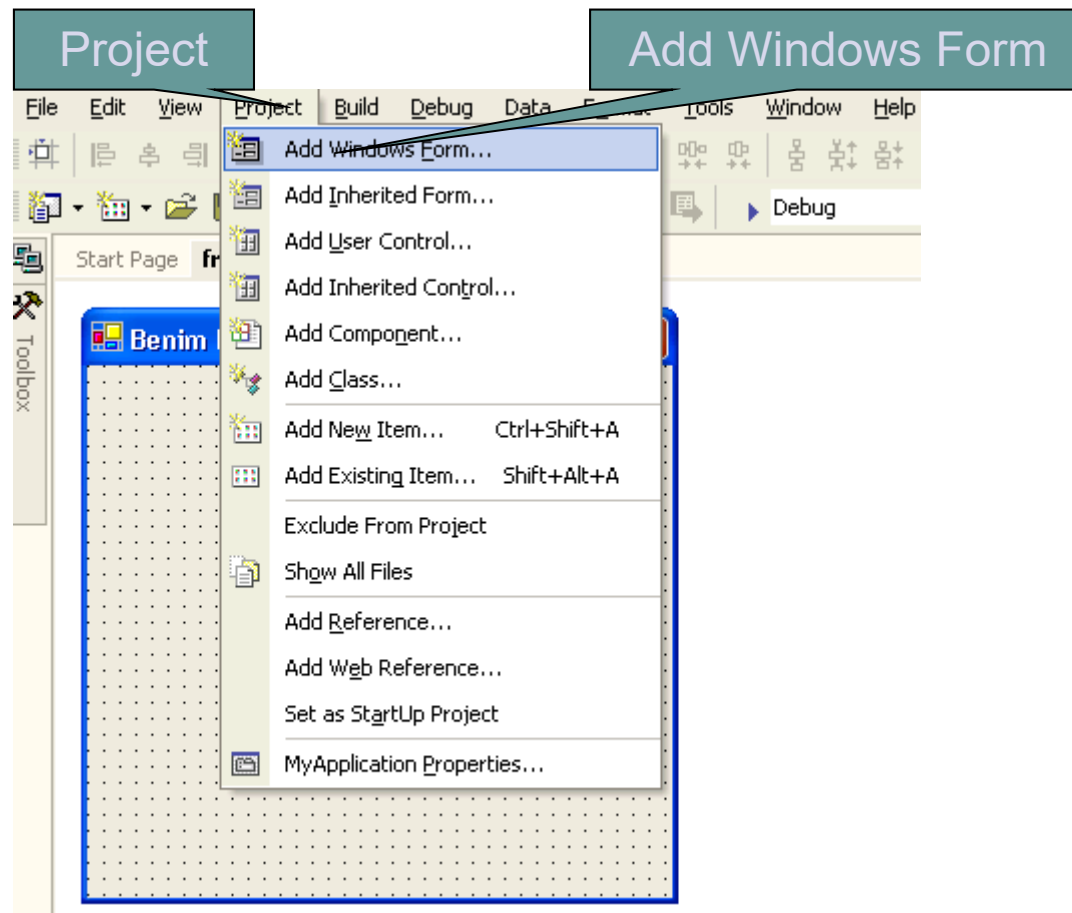
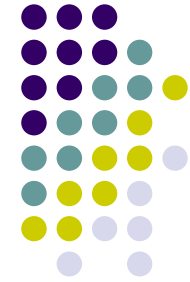


Using Multiple Forms



Creating a New Form

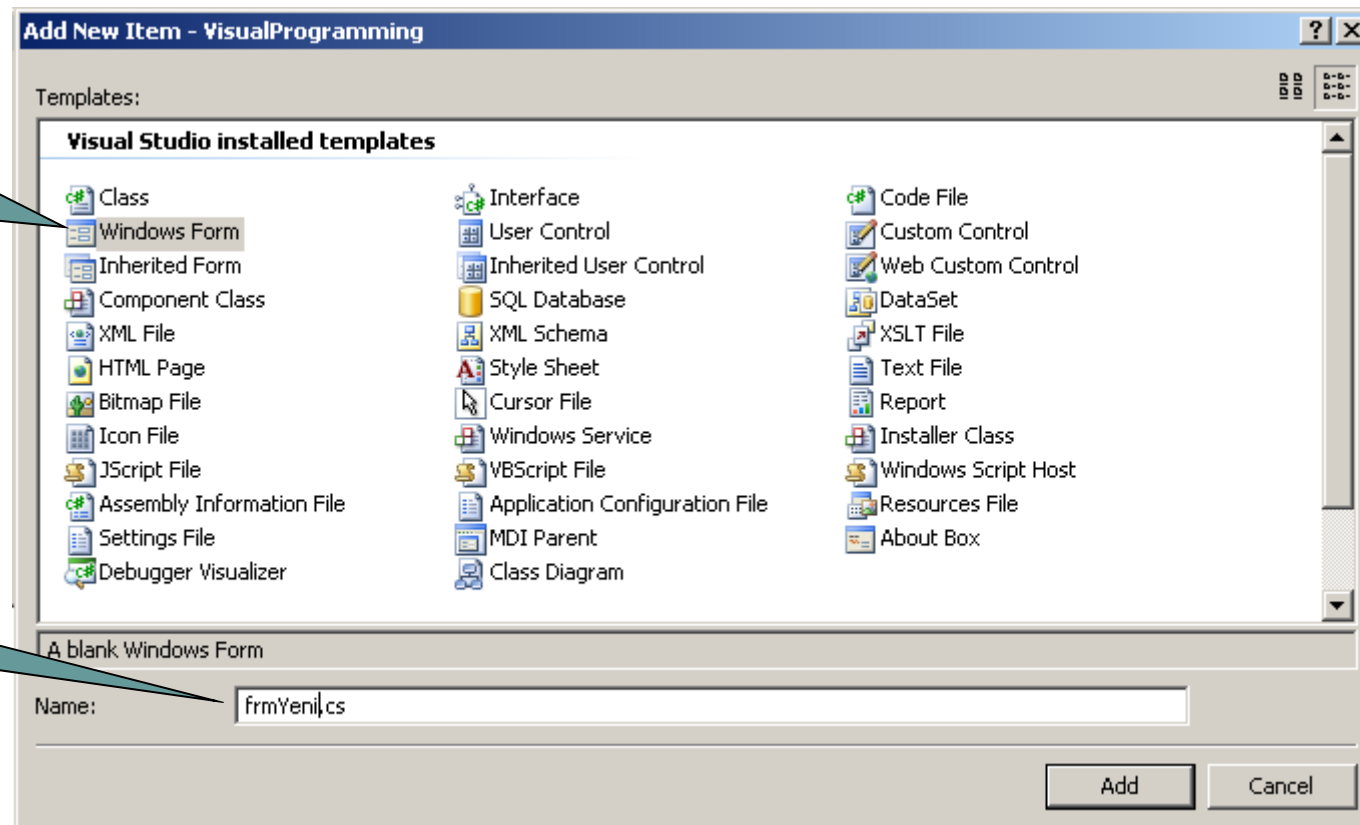


Creating a New Form

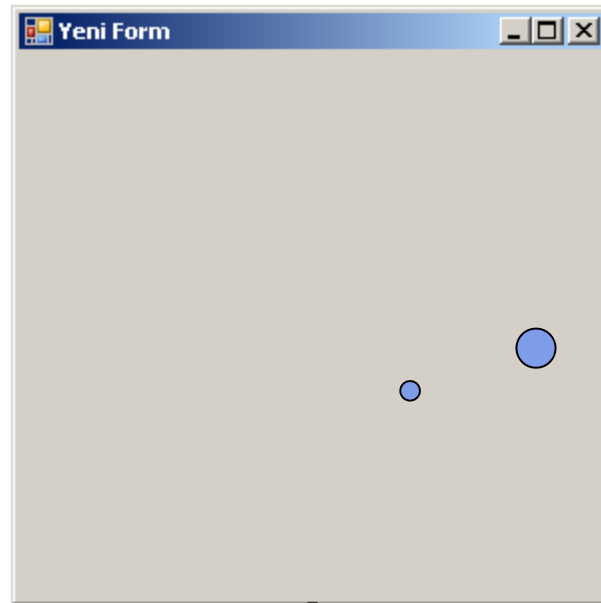


Select
Windows
Form

Name of
the form

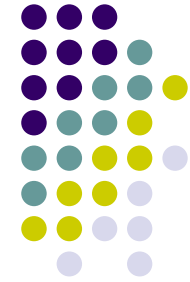


Creating a New Form



You can set properties of the form as usual.

Adding Controls to the Form



Controls

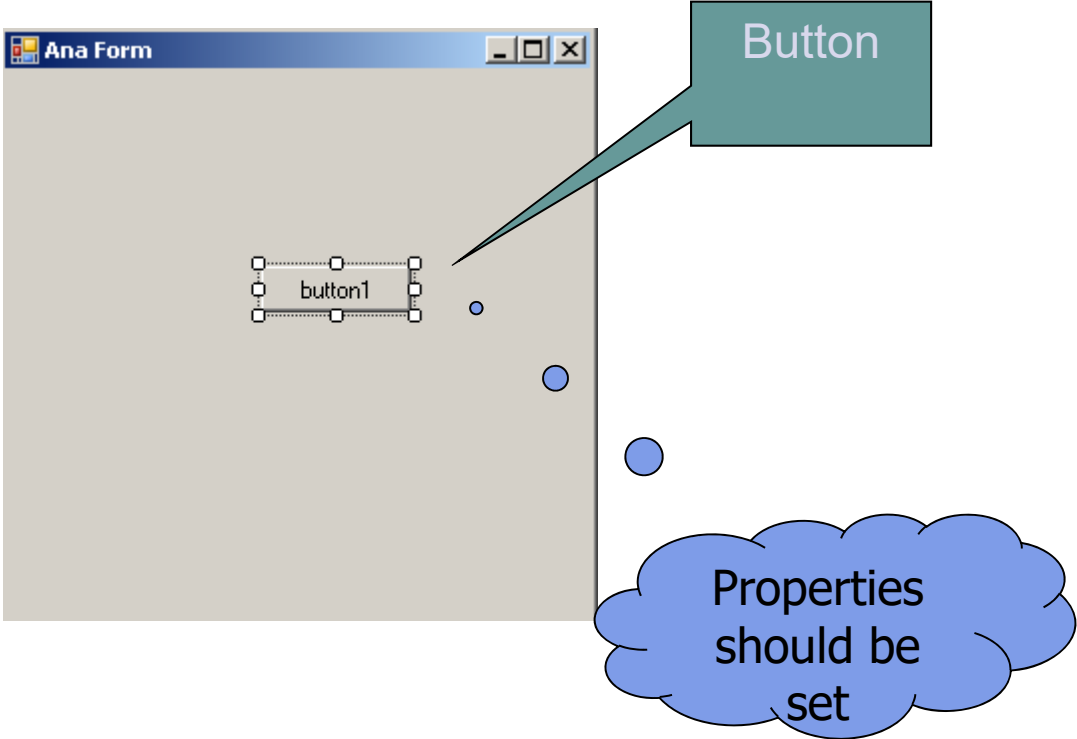
frmYeni.cs [Design]* Form1.cs [Design] Start Page

Yeni Form

Drag and drop controls

The image shows a screenshot of the Visual Studio IDE. On the left is the 'Toolbox' window, which is expanded to show 'Common Controls'. A list of controls is visible, including Pointer, Button, CheckBox, CheckedListBox, ComboBox, DateTimePicker, Label, LinkLabel, ListBox, ListView, MaskedTextBox, MonthCalendar, NotifyIcon, NumericUpDown, and PictureBox. A bracket on the left side of the Toolbox is labeled 'Controls'. On the right is the 'Design' view of a form titled 'Yeni Form'. Three blue circles are placed on the form, representing controls that have been added. A blue cloud-shaped callout bubble with the text 'Drag and drop controls' is positioned over the form. The top of the IDE window shows the file names 'frmYeni.cs [Design]*', 'Form1.cs [Design]', and 'Start Page'.

Button Properties



The image shows the Properties window for a button control named "button1". The window is titled "Properties" and shows various properties categorized into groups like Appearance, Behavior, Configurations, Data, Design, Focus, and Layout. The "Text" property is highlighted in blue.

Property	Value
AccessibleDes	
AccessibleNan	
AccessibleRole	Default
Appearance	
BackColor	Control
BackgroundIm	(none)
Cursor	Default
★ FlatStyle	Standard
Font	Microsoft Sans Seril
ForeColor	ControlText
★ Image	(none)
ImageAlign	MiddleCenter
ImageIndex	(none)
ImageList	(none)
RightToLeft	No
★ Text	button1
TextAlign	MiddleCenter
Behavior	
AllowDrop	False
ContextMenu	(none)
DialogResult	None
★ Enabled	True
TabIndex	0
TabStop	True
★ Visible	True
Configurations	
(DynamicPrope	
Data	
(DataBindings)	
Tag	
Design	
★ (Name)	button1
Locked	False
Modifiers	Private
Focus	
CausesValidati	True
Layout	
Anchor	Top, Left
Dock	None
Location	72; 40
Size	75; 23

Making "Button" Functional



Properties

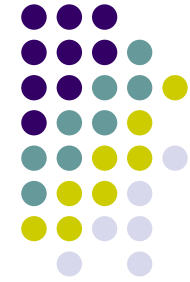
Events

Click Event

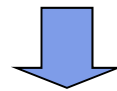
Double click to create a "handler" to the "Click" event.

Category	Event
Action	Click
Appearance	Paint
Behavior	ChangeUICues
Behavior	HelpRequestec
Behavior	QueryAccessib
Behavior	StyleChanged
Behavior	SystemColorsC
Data	(DataBindings)
Drag Drop	DragDrop
Drag Drop	DragEnter
Drag Drop	DragLeave
Drag Drop	DragOver
Drag Drop	GiveFeedback
Drag Drop	QueryContinue
Focus	Enter
Focus	Leave
Focus	Validated
Focus	Validating
Key	

Generated Code

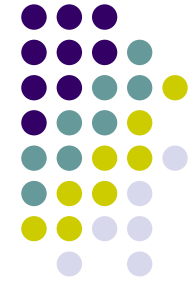


```
private void InitializeComponent()  
{  
    this.btnAc = new System.Windows.Forms.Button();  
    this.SuspendLayout();  
    //  
    // btnAc  
    //  
    this.btnAc.Location = new System.Drawing.Point(91, 89);  
    this.btnAc.Name = "btnAc";  
    this.btnAc.Size = new System.Drawing.Size(111, 23);  
    this.btnAc.TabIndex = 0;  
    this.btnAc.Text = "Yeni Form Aç";  
    this.btnAc.UseVisualStyleBackColor = true;  
    this.btnAc.Click += new System.EventHandler(this.btnAc_Click);  
}
```



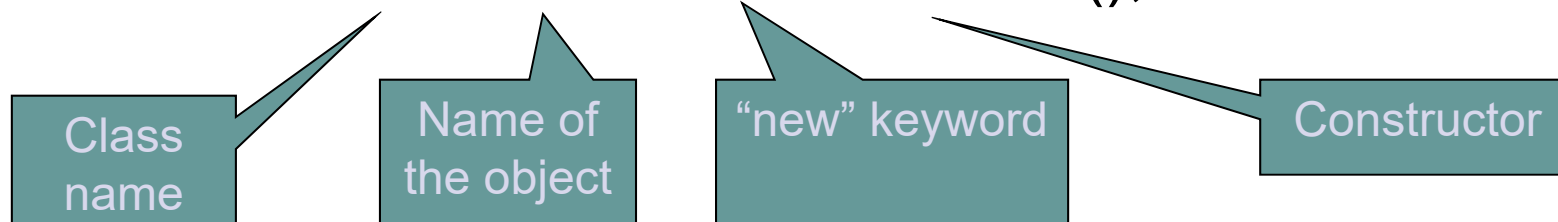
```
private void btnAc_Click(object sender, EventArgs e)  
{  
  
}
```


Creating and Opening new Form



- Creating a new form object

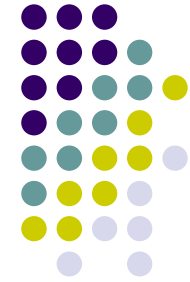
- frmNew frm = **new** frmNew ();



- Modal and Modeless Forms

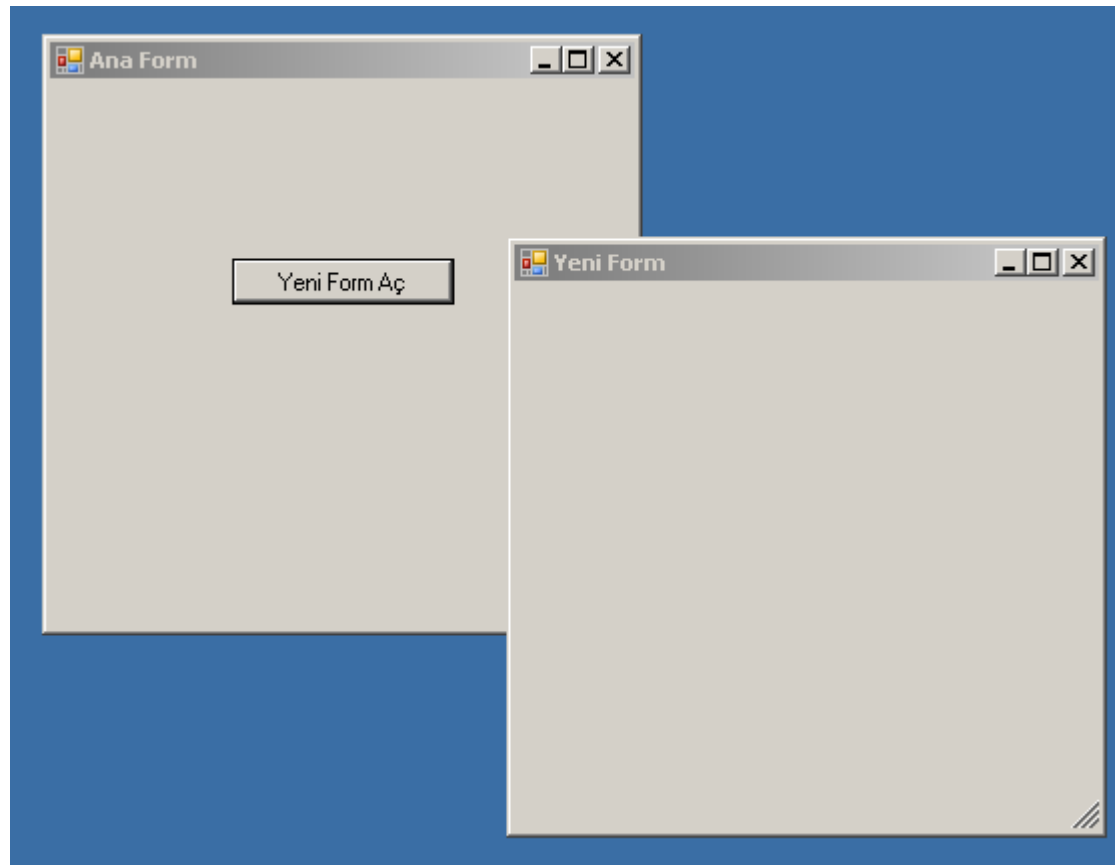
- Modal -> `frm.ShowDialog(this);`
- Modeless -> `frm.Show();`

Creating and Opening new Form - Code



```
private void btnAc_Click(object sender, EventArgs e)
{
    frmYeni frm = new frmYeni();
    frm.ShowDialog(this);
}
```

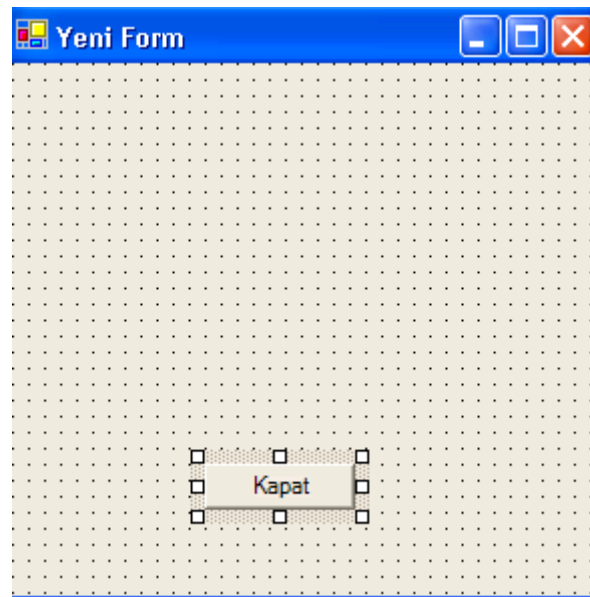
Creating and Opening new Form - Result





Closing the Form

To write the code that will close the form, first add a button to the form named "Close". Then double click on the button and add the code below.



Code

```
private void btnKapat_Click(object sender, System.EventArgs e)
{
    this.Close();
}
```



Transferring Data between forms

- Send data via constructor.

```
public partial class frmYeni : Form
{
    public frmYeni()
    {
        InitializeComponent();
    }

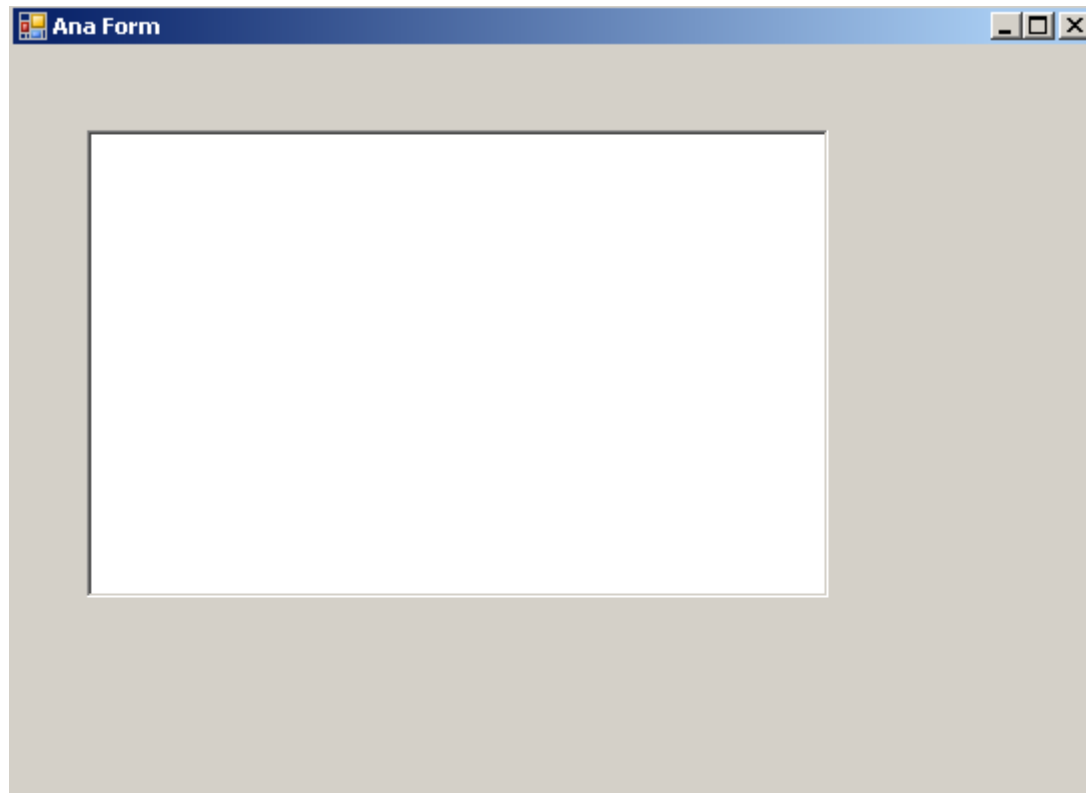
    private void btnKapat_Click(object sender, EventArgs e)
    {
        this.Close();
    }
}
```

Constructor



Example

- Add a new `ListView` to `frmMain`.



List View Properties



Properties

listView1 System.Windows.Forms.I

Accessibility

AccessibleDescr	
AccessibleName	
AccessibleRole	Default

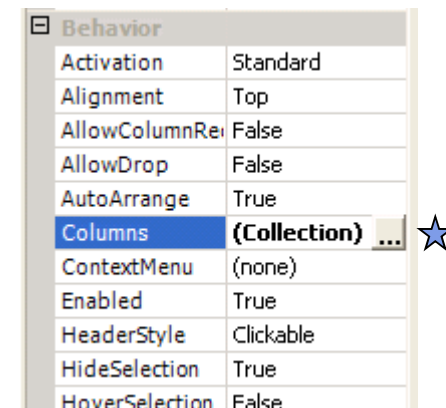
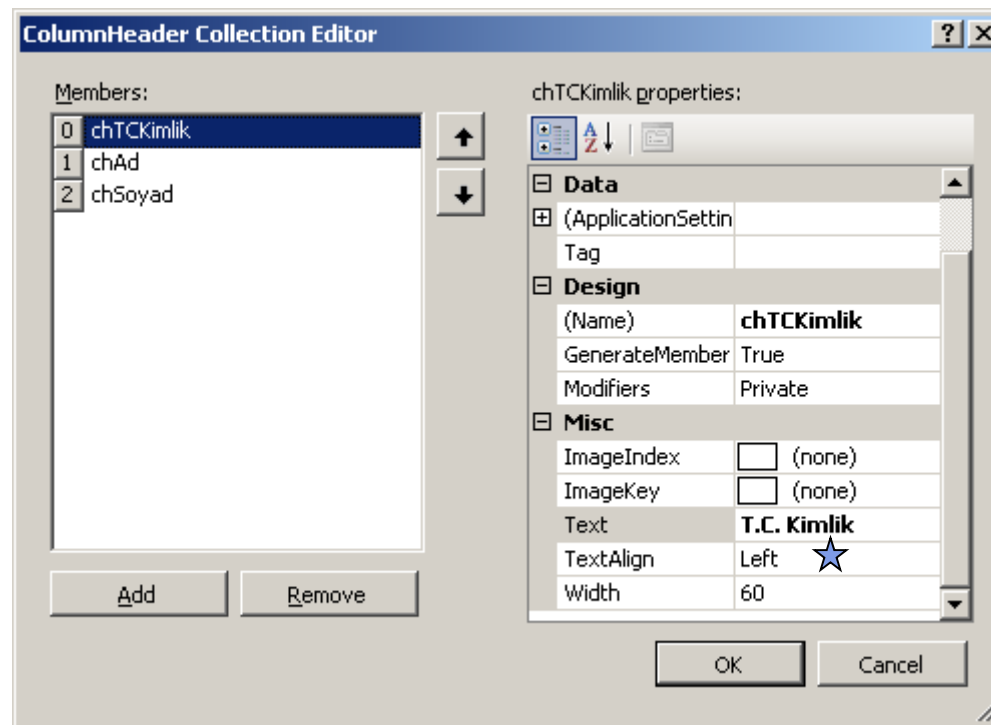
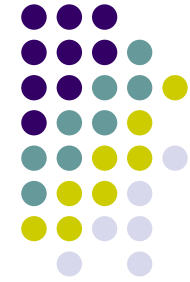
Appearance

BackColor	<input type="text" value="Window"/>
BorderStyle	Fixed3D
CheckBoxes	False
Cursor	Default
Font	Microsoft Sans Ser
ForeColor	<input type="text" value="WindowText"/>
FullRowSelect	False
GridLines	False
RightToLeft	No
View	LargeIcon

Behavior

Activation	Standard
Alignment	Top
AllowColumnRe	False
AllowDrop	False
AutoArrange	True
Columns	(Collection)
ContextMenu	(none)
Enabled	True
HeaderStyle	Clickable
HideSelection	True
HoverSelection	False
ImeMode	NoControl
Items	(Collection) ...
LabelEdit	False
LabelWrap	True
LargeImageList	(none)
MultiSelect	True
Scrollable	True
SmallImageList	(none)
StateImageList	(none)
TabIndex	1
TabStop	True
Visible	True

Adding Columns to ListView



Adding Columns to ListView



Ana Form

T.C. Kimlik	Ad	Soyad
-------------	----	-------

Kapat

Appearance

BackColor	Window
BorderStyle	Fixed3D
CheckBoxes	False
Cursor	Default

Font

Font	Microsoft Sans Ser
ForeColor	WindowText
FullRowSelect	False
GridLines	False
RightToLeft	No

View

Details

Behavior

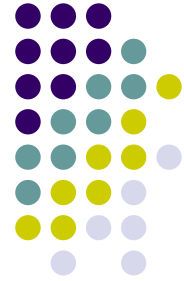
Activation	Details
Alignment	SmallIcon
AllowColumnRe	List
AllowDrop	False
AutoArrange	True



Working with data



- Operations:
 - Add/Delete/Modify
- We will need to create a new form for addition and modification operations.
- Deletion operation will delete selected record.



Add/Delete Data

- To perform deletion and modification operations, first we add two new buttons to the form and assigning appropriate properties.

The screenshot shows a Windows application window titled "Ana Form". The window contains a table with three columns: "T.C. Kimlik", "Ad", and "Soyad". Below the table, there are two buttons: "Ekle" with a star icon and "Sil" with a star icon. At the bottom center of the window, there is a "Kapat" button.



Add new record

- To transfer data, first define a “struct” named Uye.

```
public struct Uye
{
    public string sTCId;
    public string sAd;
    public string sSoyad;
}
```



Add/Modify Form

- Create a new form named “frmKayitEkleDuzelt” to perform add and modify operations. Add three textboxes and labels, one OK and one Cancel buttons to the form.

The screenshot shows a Windows form titled "Kayıt Ekle / Düzelt". The form has a blue title bar. The main area has a dotted background. There are three textboxes arranged horizontally, labeled "TC Kimlik", "Ad", and "Soyad". Below the textboxes are two buttons labeled "Tamam" and "İptal".



Displaying records in add/modify form

- Get data in Uye structure via constructor and assign to the controls.

```
public partial class frmKayitEkleDuzelt : Form
{
    public frmKayitEkleDuzelt(Uye uye)
    {
        InitializeComponent();
        //aktarma
        txtTCKimlik.Text = uye.sTCId;
        txtAd.Text = uye.sAd;
        txtSoyad.Text = uye.sSoyad;
    }
}
```

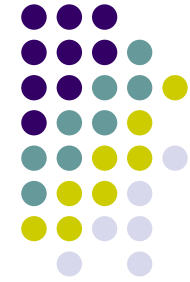
Assign Operation

OK and Cancel Buttons



- How can we understand that user pressed OK or Cancel buttons?
- How to send data back?

Solution



First, define a global Uye structure.

```
public partial class frmKayitEkleDuzelt : Form
{
    private Uye m_Uye;★

    public frmKayitEkleDuzelt(Uye uye)
    {
        InitializeComponent();
        //aktarma
        txtTCKimlik.Text = uye.sTCId;
        txtAd.Text = uye.sAd;
        txtSoyad.Text = uye.sSoyad;
    }
}
```


Solution - Continues



Double click to OK and Cancel buttons, add following codes to handler functions.

```
public partial class frmKayitEkleDuzelt : Form
{
    private Uye m_Uye;

    public frmKayitEkleDuzelt(Uye uye) {...}

    private void btnTamam_Click(object sender, EventArgs e)
    {
        m_Uye = new Uye();
        m_Uye.sTCId = txtTCKimlik.Text;
        m_Uye.sAd = txtAd.Text;
        m_Uye.sSoyad = txtSoyad.Text;

        this.DialogResult = DialogResult.OK;
    }

    private void btnIptal_Click(object sender, EventArgs e)
    {
        this.DialogResult = DialogResult.Cancel;
    }
}
```

} Structure to be returned.



Solution - Continues

Create a **public** function on “frmKayitEkleDuzelt” form. This function will return the global struct **m_Uye** which is created by OK button click.

```
public partial class frmKayitEkleDuzelt : Form
{
    private Uye m_Uye;

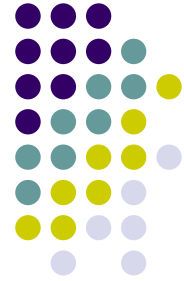
    public frmKayitEkleDuzelt(Uye uye) {...}

    private void btnTamam_Click(object sender, EventArgs e) {...}

    private void btnIptal_Click(object sender, EventArgs e) {...}

    public Uye VeriGetir()
    {
        return m_Uye;
    }
}
```

} Code that will return structure.



Main form- Add

To add a new record, add following code to the handler function of the **Ekle** button

```
private void btnEkle_Click(object sender, System.EventArgs e)
{
    Uye uye = new Uye();
    uye.sTCId = string.Empty;
    uye.sAd = string.Empty;
    uye.sSoyad = string.Empty;

    frmKayitEkleDuzelt frm = new frmKayitEkleDuzelt(uye);
    if(frm.ShowDialog(this) == DialogResult.OK)
    {
        uye = frm.Verigetir();
        ListViewItem lvi = new ListViewItem(new string[] {
            uye.sTCId,
            uye.sAd,
            uye.sSoyad});
        lvUyeler.Items.Add(lvi);
    }
}
```

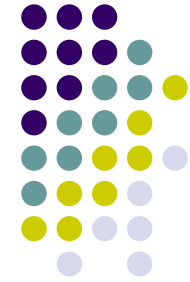
Adding record form

Empty Uye struct

If OK button is pressed

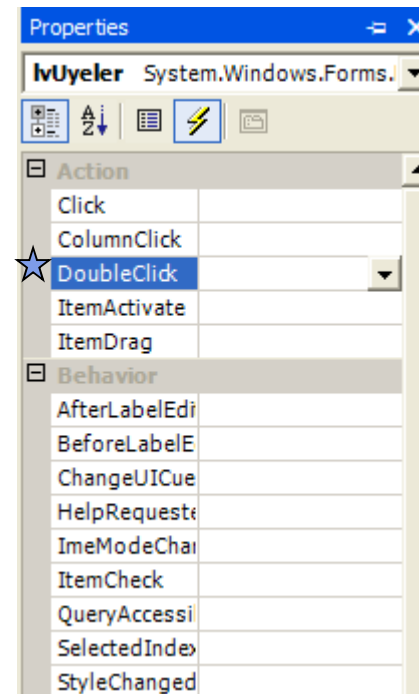
Add new element to listview

Main form - Modify

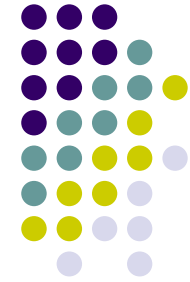


To modify a record let's create a handler function to double click event of the listview. When a row is double clicked, we will open it to be edited.

To achieve this, create a event handler function for double click event of listview.



Main form - Modify



Add following code to handler function.

```
private void lvUyeler_DoubleClick(object sender, System.EventArgs e)
{
    if(lvUyeler.SelectedItems.Count>0)
    {
        ListViewItem lvi = lvUyeler.SelectedItems[0];
        Uye uye = new Uye();
        uye.sTCId = lvi.SubItems[0].Text;
        uye.sAd = lvi.SubItems[1].Text;
        uye.sSoyad = lvi.SubItems[1].Text;

        frmKayitEkleDuzelt frm = new frmKayitEkleDuzelt(uye);
        if(frm.ShowDialog(this) == DialogResult.OK)
        {
            uye = frm.VeriGetir();

            lvi.SubItems[0].Text = uye.sTCId;
            lvi.SubItems[1].Text = uye.sAd;
            lvi.SubItems[2].Text = uye.sSoyad;
        }
    }
}
```

Selected element

At least one row is selected.

Record to be modified

If OK button is pressed

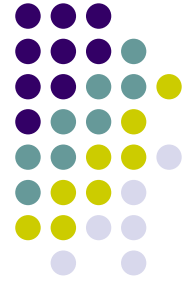
Modify record



Main form - Delete

To delete selected row, create a handler function for Sil button.

The screenshot shows a Windows application window titled "Ana Form". Inside the window, there is a table with three columns: "T.C. Kimlik", "Ad", and "Soyad". The table is currently empty. To the right of the table, there are two buttons: "Ekle" (Add) and "Sil" (Delete). The "Sil" button is highlighted with a dashed border, indicating it is the selected element. At the bottom center of the form, there is a "Kapat" (Close) button.



Main form - Delete

Add following code to handler function.

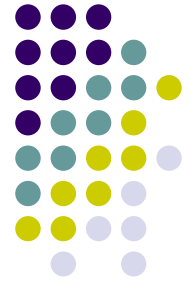
```
private void btnSil_Click(object sender, System.EventArgs e)
{
    if (lvUyeler.SelectedItems.Count > 0)
    {
        if (MessageBox.Show(this, "Emin misiniz?",
                            "Soru",
                            MessageBoxButtons.YesNo,
                            MessageBoxIcon.Question) == DialogResult.Yes)
        {
            lvUyeler.Items.Remove(lvUyeler.SelectedItems[0]);
        }
    }
}
```

If user is sure
of what she is
doing

At least one
element is
selected

Remove
selected
element
from listview

Main Menu

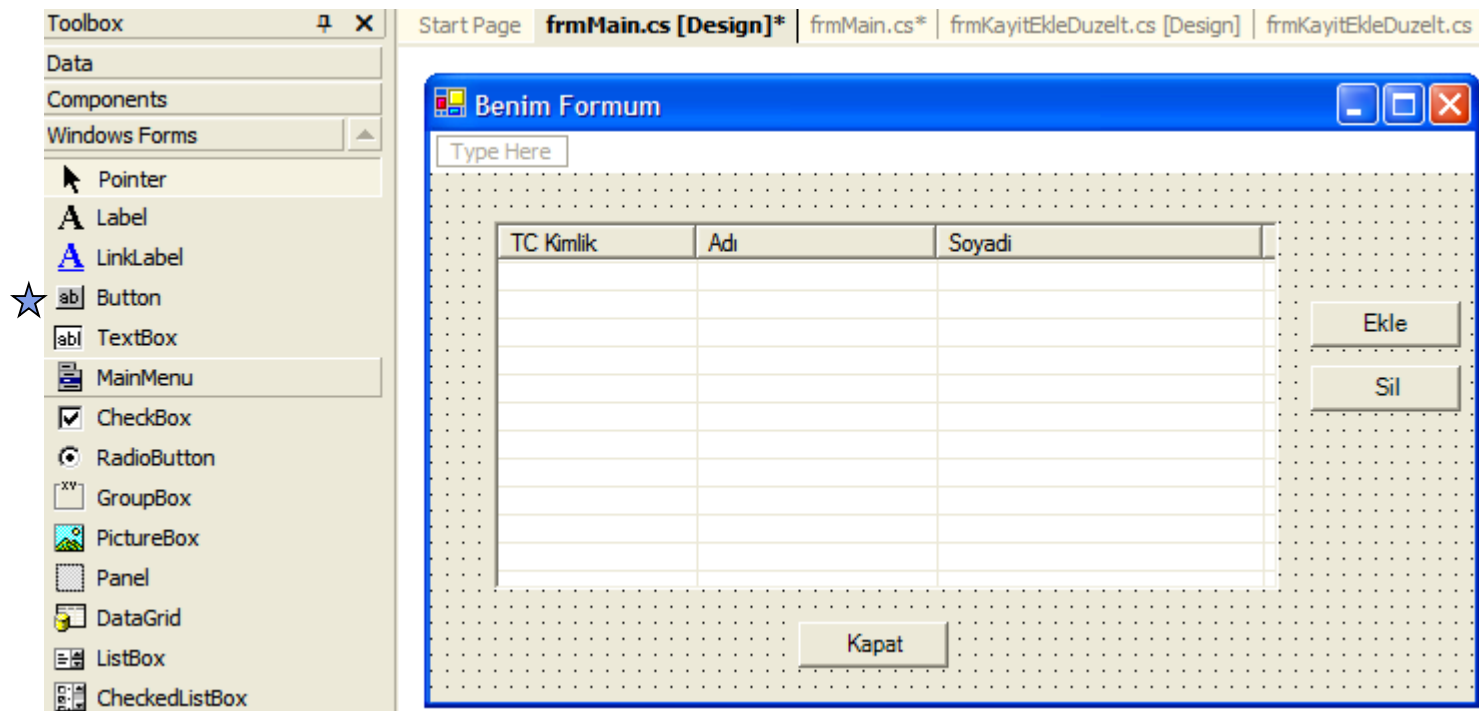


- Main menu is the control that resides upper section of the form.
- It provides easy access for frequently used operations.

Adding Main Menu



MainMenu is added to the form by dragging and dropping from.



Adding new element to the Main Menu



To add new items to the MainMenu, just type the name of the element to the areas written “Type Here” in the MainMenu.

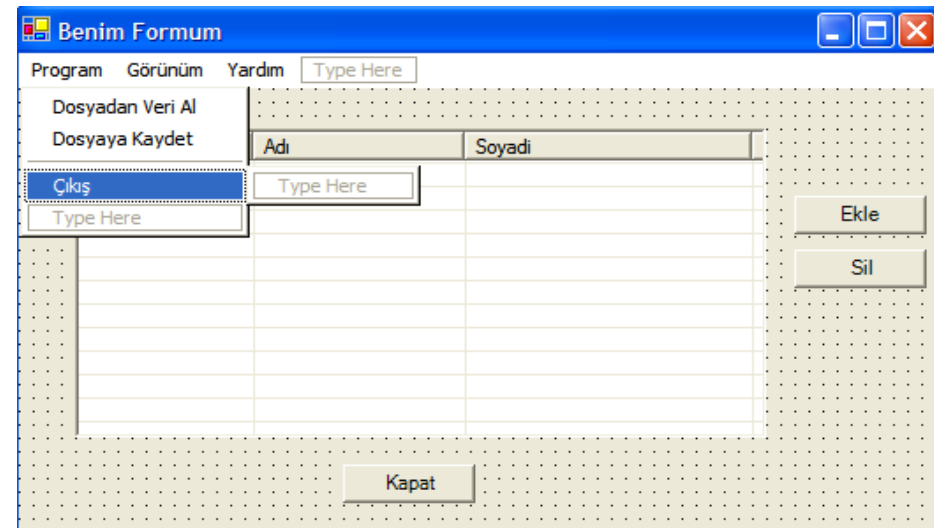
The screenshot shows a Windows application window titled "Benim Formum". The window has a menu bar with the following items: "Program", "Görünüm", "Yardım", and a text box containing "Type Here". The "Program" menu is open, displaying a list of options: "Dosyadan Veri Al", "Dosyaya Kaydet", "Çıkış", and another text box containing "Type Here". The main area of the window contains a table with two columns, "Adı" and "Soyadı", and several empty rows. To the right of the table are two buttons labeled "Ekle" and "Sil". At the bottom center of the window is a button labeled "Kapat".

Making menu element operational



Just double click on the menu element to create handler function to their default events.

You may enter whatever code you wanted to the handler function.



```
private void miCikis_Click(object sender, System.EventArgs e)
{
    if (MessageBox.Show(this, "Kaydetmeden çıkmak istediğinizden emin misiniz?",
        "Soru",
        MessageBoxButtons.YesNo,
        MessageBoxIcon.Question) == DialogResult.Yes)
    {
        this.Close();
    }
}
```



Writing to a file

- In C# language `System.IO` namespace is used for file operations.
- To enable Turkish language support, it is suggested to use “StreamWriter” class.
- To get file name to be written from user, you may use `SaveFileDialog` class.

Example



Add following code to the click event handler function of the MainMenu element named "Save".

```
private void miDosyaYaz_Click(object sender, System.EventArgs e)
{
    SaveFileDialog sfd = new SaveFileDialog();
    sfd.Filter = "txt files (*.txt)|*.txt|All files (*.*)|*.*" ;

    if(sfd.ShowDialog() == DialogResult.OK)
    {
        using (StreamWriter swr = new StreamWriter(sfd.FileName, false, Encoding.GetEncoding("iso-8859-9")))
        {
            foreach(ListViewItem lvi in lvUyeler.Items)
            {
                swr.WriteLine(string.Format("{0,20:G}", lvi.SubItems[0].Text) +
                    string.Format("{0,20:G}", lvi.SubItems[1].Text) +
                    string.Format("{0,20:G}", lvi.SubItems[2].Text));
            }
        }
    }
}
```



SaveFileDialog

SaveFileDialog used used for asking to user which file will be selected for write operation.

Create
SaveFileDialog
object

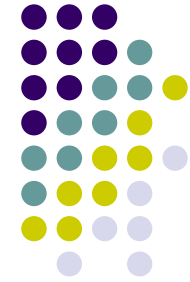
Filter definition

```
SaveFileDialog sfd = new SaveFileDialog();  
sfd.Filter = "txt files (*.txt)|*.txt|All files (*.*)|*.*" ;
```

```
if(sfd.ShowDialog() == DialogResult.OK)
```

Show dialog box and
decide whether OK
button is pressed.

StreamWriter



This object is used for writing to a file. Its constructor accepts two arguments, one for file name to be written, one for “encoding” which decides code page. WriteLine function write data to the file.

Create StreamWriter object

Add Turkish support

```
using (StreamWriter swr = new StreamWriter(sfd.FileName, false, Encoding.GetEncoding("iso-8859-9")))
{
    foreach(ListViewItem lvi in lvUyeler.Items)
    {
        swr.WriteLine(string.Format("{0,20:G}", lvi.SubItems[0].Text) +
            string.Format("{0,20:G}", lvi.SubItems[1].Text) +
            string.Format("{0,20:G}", lvi.SubItems[2].Text));
    }
}
```

Write a line to the file

Create special formatted strings

Get data from listview



Try the program – New record

The screenshot shows a Windows application window titled "Benim Formum" with a menu bar containing "Program", "Görünüm", and "Yardım". The main area contains a table with three columns: "TC Kimlik", "Adı", and "Soyadı". To the right of the table are two buttons: "Ekle" and "Sil". At the bottom of the window is a "Kapat" button. A modal dialog titled "Kayıt Ekle / Düzelt" is open in the foreground, featuring three input fields for "TC Kimlik", "Ad", and "Soyad". The "TC Kimlik" field contains "11111111111", the "Ad" field contains "bill", and the "Soyad" field contains "gates". Below the input fields are two buttons: "Tamam" and "İptal".

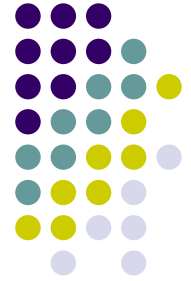
TC Kimlik	Adı	Soyadı

Kayıt Ekle / Düzelt

TC Kimlik	Ad	Soyad
11111111111	bill	gates

Tamam İptal

Try the program – New record



Benim Formum

Program Görünüm Yardım

TC Kimlik	Adı	Soyadı
11111111111	bill	gates

Ekle

Sil

Kapat



Try the program – Modify a record

Benim Formum

Program Görünüm Yardım

TC Kimlik	Adı	Soyadı
111111111111	bill	gates

Ekle

Sil

Kapat

Kayıt Ekle / Düzelt

TC Kimlik Ad Soyad

29962598516 Reha Oğuz ALTUĞ

Tamam İptal

Try the program – Modify a record



Benim Formum

Program Görünüm Yardım

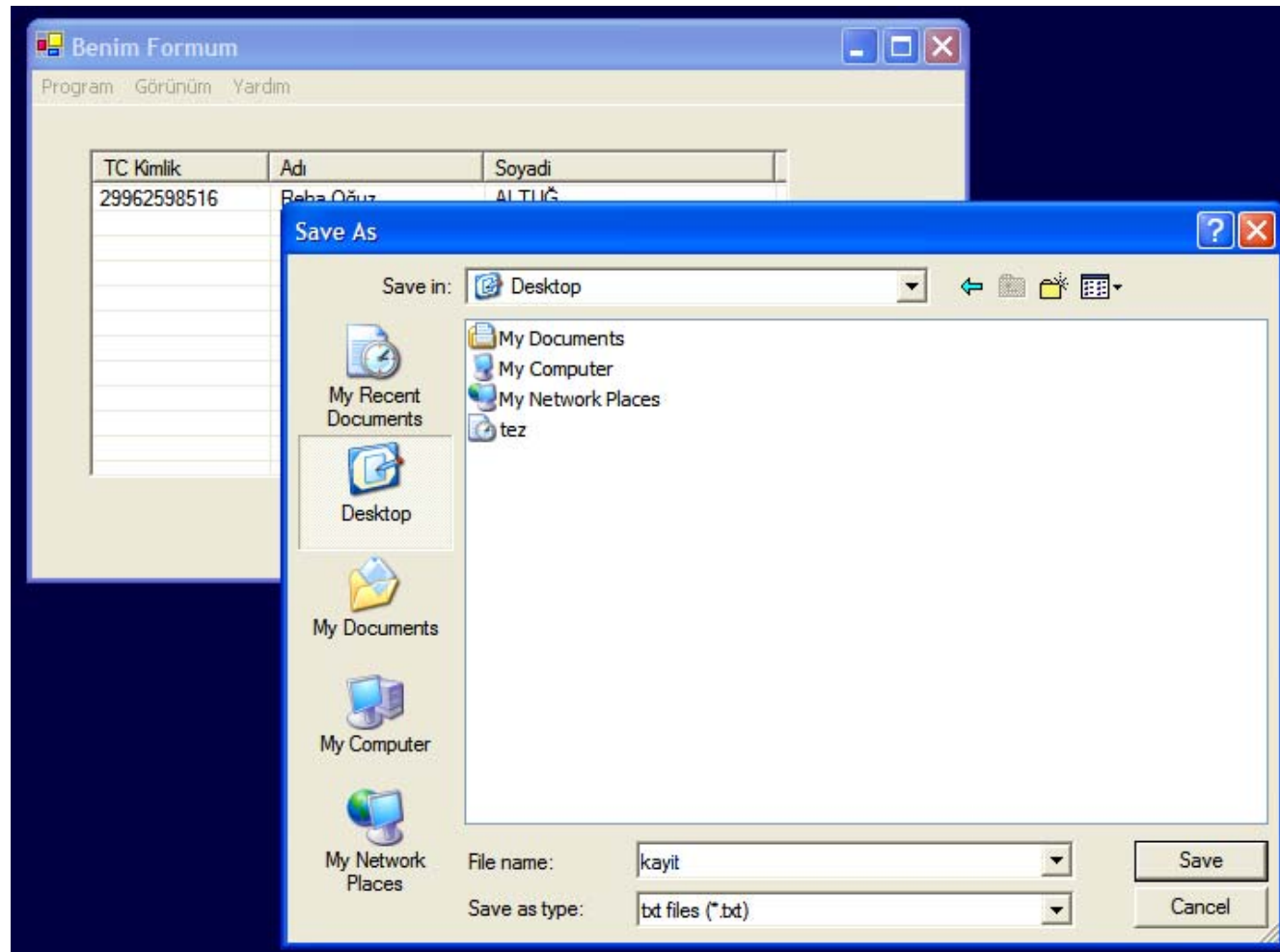
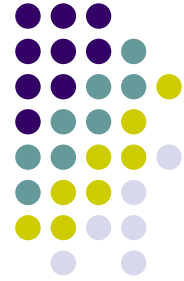
TC Kimlik	Adı	Soyadı
29962598516	Reha Oğuz	ALTUĞ

Ekle

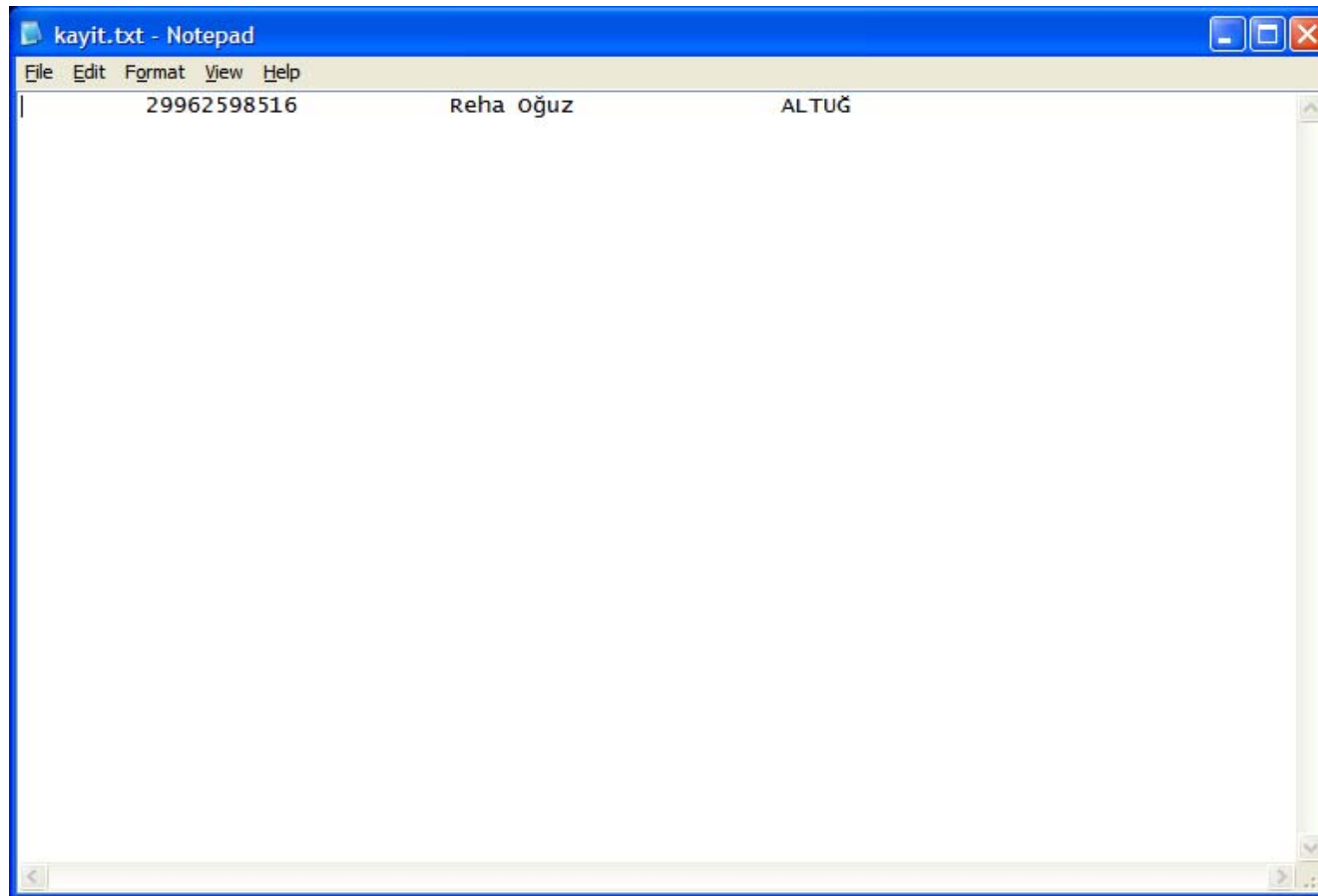
Sil

Kapat

Try the program – Saving to a file



Try the program – Saving to a file





Read from file

- As in writing file, **System.IO** name space includes classes which are used for reading from a file.
- Use “StreamReader” class for Turkish language support.
- Use **OpenFileDialog** class to ask user which file is meant to be read.

Example



Add following code to the click event handler function of the “Read from file” menu item.

```
private void miDosyaOku_Click(object sender, System.EventArgs e)
{
    OpenFileDialog ofd = new OpenFileDialog();
    ofd.Filter = "txt files (*.txt)|*.txt|All files (*.*)|*.*" ;
    if(ofd.ShowDialog() == DialogResult.OK)
    {
        using (StreamReader srdr = new StreamReader(ofd.FileName,Encoding.GetEncoding("iso-8859-9")))
        {
            string sRead = string.Empty;
            while ((sRead = srdr.ReadLine()) != null)
            {
                ListViewItem lvi = new ListViewItem(new string[] {sRead.Substring(0,20).Trim(),
                                                                    sRead.Substring(20,20).Trim(),
                                                                    sRead.Substring(40,20).Trim()});

                lvUyeler.Items.Add(lvi);
            }
        }
    }
}
```


OpenFileDialog



OpenFileDialog is used for selecting the file name to be opened for reading.

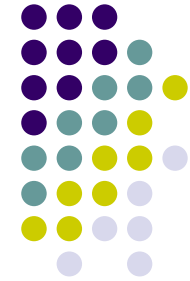
Create OpenFileDialog object

Define filter

```
OpenFileDialog ofd = new OpenFileDialog();  
ofd.Filter = "txt files (*.txt)|*.txt|All files (*.*)|*.*" ;  
if(ofd.ShowDialog() == DialogResult.OK)  
{
```

Show dialog box and decide whether OK button is pressed.

StreamReader



This is the object that reads from a file. Its constructor accepts two arguments, one for file name to be read, one for “encoding” which decides code page. ReadLine function reads one line from file.

Create StreamReader

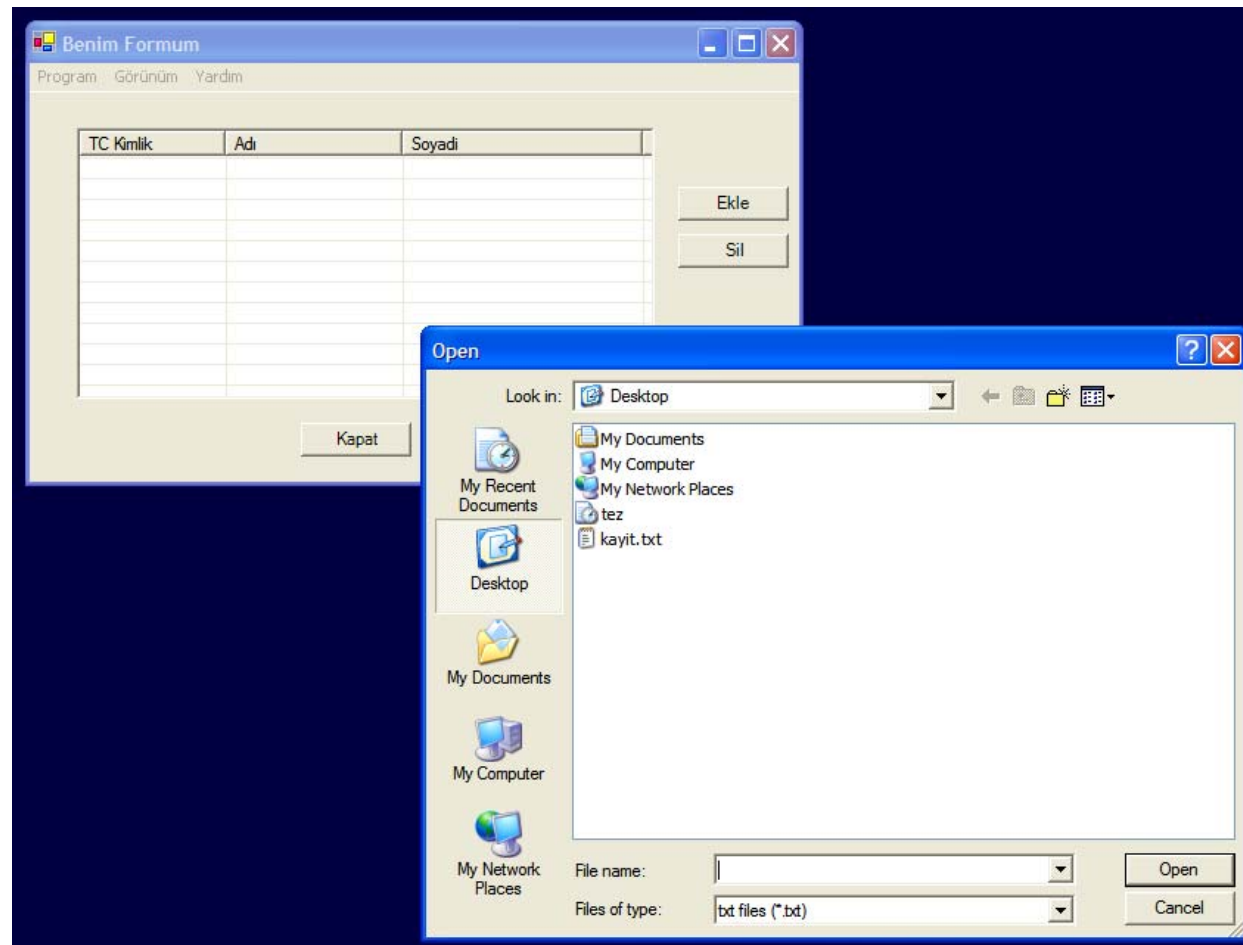
Read with Turkish support

```
using (StreamReader srdr = new StreamReader(ofd.FileName, Encoding.GetEncoding("iso-8859-9")))
{
    string sRead = string.Empty;
    while ((sRead = srdr.ReadLine()) != null)
    {
        ListViewItem lvi = new ListViewItem(new string[] {sRead.Substring(0,20).Trim(),
                                                         sRead.Substring(20,20).Trim(),
                                                         sRead.Substring(40,20).Trim() });
        lvUyeler.Items.Add(lvi);
    }
}
```

Read a line from file

Add read data to the listview

Try the program – Read from a file



Try the program – Read from a file



Benim Formum

Program Görünüm Yardım

TC Kimlik	Adı	Soyadı
29962598516	Reha Oğuz	ALTUĞ

Ekle

Sil

Kapat