



# SDÜ Eğitim Öğretim Bilgi Sistemi Ders İçeriği

Fakülte/Bölüm	Teknoloji Fakültesi \ Mekatronik Mühendisliği						
<b>Ders Bilgileri</b>							
Ders Kodu	Ders Adı	Kredi Teorik	Kredi Pratik	Kredi Lab/U	Kredi Toplam	Kredi AKTS	Yarı Yılı
MKM - 101	<b>Bilgisayar Programlama</b>	2.00	0.00	0.00	2.00	2	1
<b>Ders İzleme</b>							
Dersin Dili	Türkçe						
Dersin Türü	Zorunlu						
Dersin Koordinatörü	Yrd.Doç.Dr. Sinan UĞUZ						
Dersi Verenler	Yrd.Doç.Dr. Sinan UĞUZ						
Dersin Yardımcıları							
Dersin Amacı	Yazılım süreci hakkında genel bilgi sahibi olmak. Bilgisayar programlama ile ilgili genel kavramları ortaya koymak. Görsel programlamanın avantajlarını bilmek. <u>C# dilini ve öğretmek ve .NET platformunda C# uygulamasının hazırlamasını sağlamak.</u>						
Dersin Hedefleri	NET ortamının temel bileşen ve programlama tekniklerini etkin ve verimli olarak öğretmek Temel C# bileşenlerini söylemek Standart nesnelere kullanarak Windows uygulamalarını yapmak						
Dersin Öğrenme Çıktıları	NET ortamının temel bileşen ve programlama tekniklerini etkin ve verimli olarak kullanır. Standart nesnelere kullanarak Windows uygulamalarını tasarlar						
Dersin İçeriği	NET kavramının tanıtımı, .NET program geliştirme ortamı, .NET programlama dilleri, .NET servis birimleri (ASP.NET,ADO.NET,XML). Değişkenler, operatörler, ifadeler, fonksiyonlar, karar yapıları, döngüler, hata yönetimi ve istisnalar. Windows formlarının tanıtımı, menüler, diyalog kutuları, ortak diyalog kontrolleri, MDI ve SDI						
Önkoşul Dersleri							
Önerilen Seçmeli							
Ders Öğretim Biçimi	Yüz yüze						
<b>Değerlendirme Sistemi</b>				<b>Akts/İş Yüğü Tablosu</b>			
Yarıyıl Geçti	Sayısı	Katkı Payı	Etkinlik	Sayısı	Süresi	Toplam	
Ara Sınav	1	100	Ders Süresi (Sınav)	14	3	42	
Kısa Sınav	0	0	Sınıf Dışı Ders çalışma Süresi	14	3	42	
Ödev	0	0	Ödevler	1	4	4	
Devam	0	0	Sunum	0	0	0	
Uygulam	0	0	Proje	1	20	20	
Labaratuva	0	0	Laboratuvar	0	0	0	
Proje	0	0	Arazi ya da Alan	0	0	0	
Atölye	0	0	Ara	1	14	14	
Seminer	0	0	Yarıyıl Sonu	1	15	15	
Arazi	0	0	Öğ Yüğü	30			
Toplam	100		Toplam	136			
Yılıçin Baçarıya	40		Dersin Akts	5			
Finalin Baçarıya	60						
Toplam	100						



# SDÜ Eğitim Öğretim Bilgi Sistemi Ders İçeriği

<b>Kaynaklar</b>		
Ders Notu	C# Baçlangıç Rehberi, Ğ.Karagülle, Türkmen Yayınevi Her yönüyle C#, Sefer Algan, Pusula	
Diğer Kaynaklar	<a href="http://obs.sdu.edu.tr">http://obs.sdu.edu.tr</a> Ders Dökümanlarından ulaşılabilir <a href="http://www.codeproject.com">www.codeproject.com</a>	
<b>Materyal</b>		
Dokümanlar	Ders sunumları	
Ödevler		
Sınavlar		
Materyal Diğer		
<b>Planlanmış Öğrenme Faaliyetleri ve Öğretim Yöntemleri</b>		
Konferanslar, Uygulamalı Dersler, Sunumlar, Seminerler, Projeler, Laboratuar Uygulamaları(gerekirse)		
<b>İş Yerleřtirmeleri</b>		
Her hangi bir eğitimsel öge gibi, krediler yalnızca öğrenme çıktılarına ulaçıldığında ve çıktılar değerlendirildiğinde verilir. Eğer çalıřma yeri planlanmış deęiřimin parçasıysa (örneğin Farabi ve Erasmus), ve verleřtirme için beklenen öğrenme çıktılarına ulaçılmışsa Öğrenme Sözleşmesi verilen kredilerin sayılarını		
<b>Haftalık Konular</b>		
Haft	Konular	Önhazırlık
1	.NET Framework Mimarisi ve C#	
2	Visual Studio Geliřtirme Ortamı ve Görsel Programlama	
3	Temel kavramlar: deęiřkenler, veri türleri	
4	Program Kontrol Deyimleri	
5	Metotlar	
6	Ortak Windows form kontrolleri-1	
7	Ortak Windows form kontrolleri-2	
8	Ortak Windows form kontrolleri-3	
9	Ortak Windows form kontrolleri-4	
10	Hata yakalama, Diyalog Kutuları	
11	Menü Tasarımı, Çoklu Formlarla Çalıřma	
12	Dosyalama, dosya ve klasör işlemleri	
13	Dinamik Kontroller-Programın Çalıřması Sırasında Yeni Kontroller Ekleme	
14	DLL (Dinamik Link Library) oluřturma, Hazır ve kullanıcı tanımlı DLL kullanabilme	



## Ders Notları

## Kaynaklar

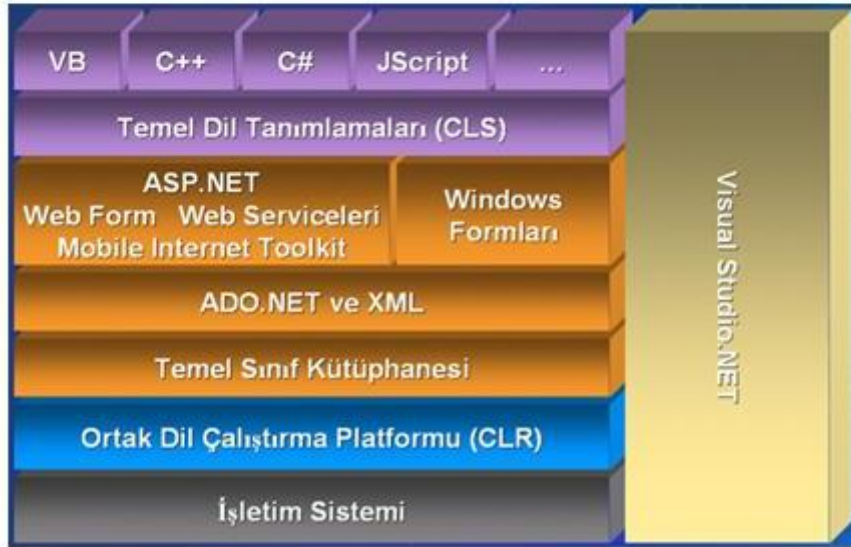


- .NET Framework Mimarisi,
- Visual Studio Kurulumu,
- Geliştirme Ortamı,
- Görsel Programlama

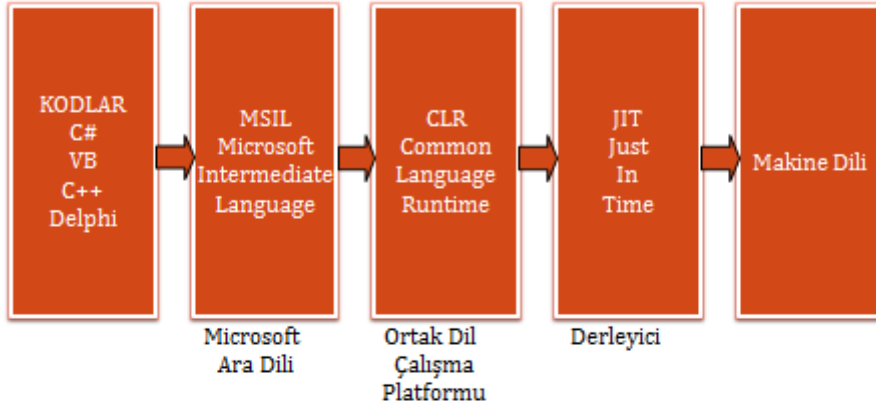
## .Net Nedir?

- .Net Framework Microsoft tarafından geliştirilen “uygulama” geliştirme platformudur.
- .Net Framework **farklı uygulamaların aynı kütüphaneyi kullanarak** web, masaüstü ve mobil uygulamalarının geliştirildiği bir ortamdır.
- Bu kütüphane Base Class Library-Temel Sınıf Kütüphanesi(BCL)dir. BCL ortak bir kütüphanedir ve bu kütüphane sayesinde biz çok sayıda hazır fonksiyona .Net ortamında program yazarken erişebilir ve kullanabiliriz.

## .Net Framework Mimarisi



## .Net Framework Mimarisi



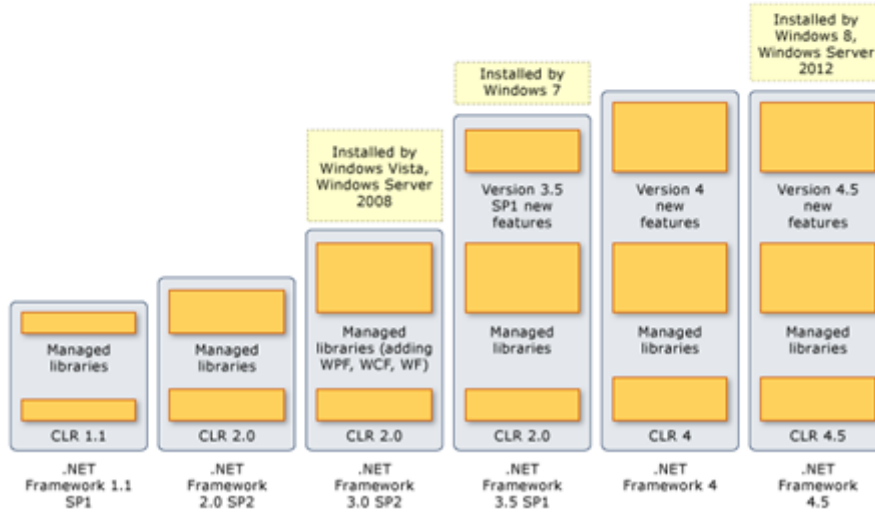
## .Net Framework Getirdikleri

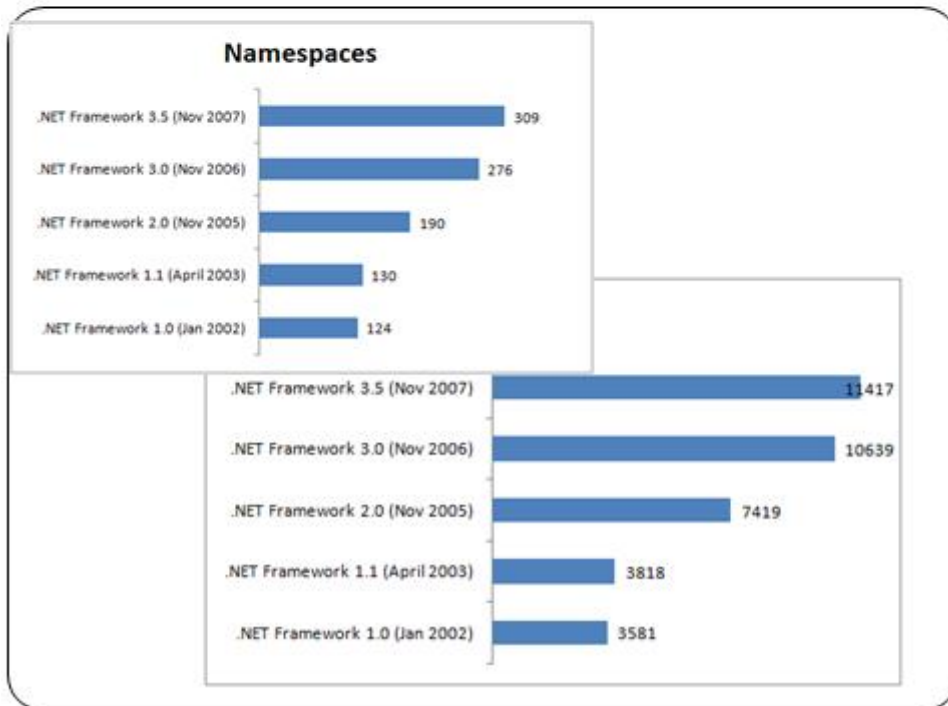
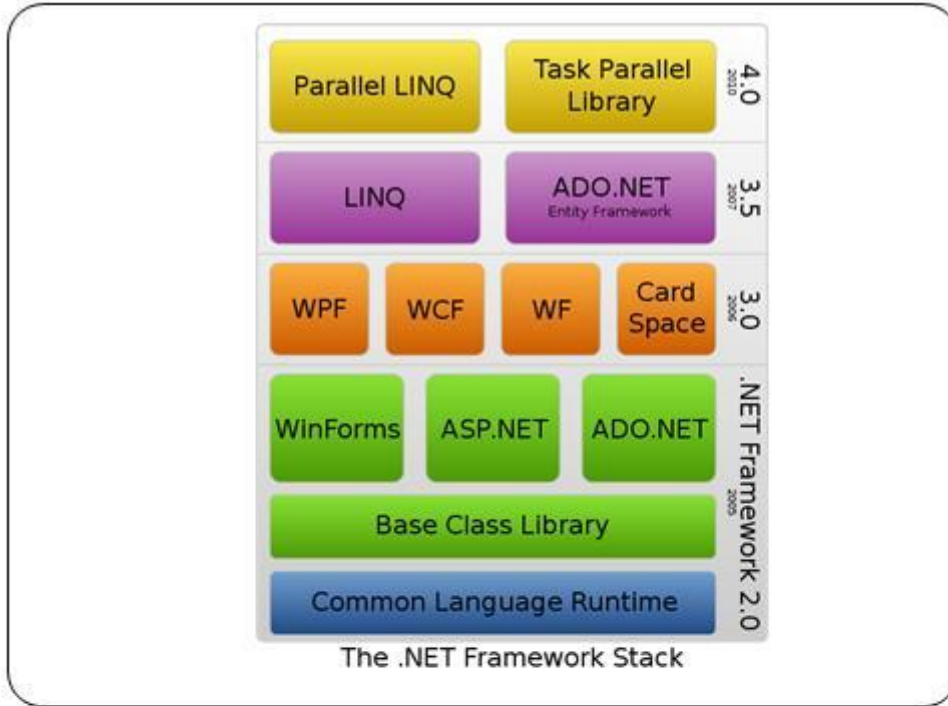
- ✓ yüksek bir verim ,
- ✓ standart bir alt yapı,
- ✓ daha basit uygulama geliştirme ,
- ✓ çoklu dillerin(multilanguage) bulunduğu bir ortam,
- ✓ var olan programlama bilgilerinden yararlanabilme,
- ✓ var olan uygulamalar ile kolay entegre olabilme,
- ✓ internet uygulamalarında kullanabilme ve çalıştırmanın rahatlığını getirir.

## .Net Framework Versiyonları

Nesil	Versiyon Numarası	Çıkış Tarihi	CLR	Geliştirme Ortamı	Dağıtım
1.0	1.0.3705.0	13 Şubat 2002	1.0	Visual Studio .NET	N/A
1.1	1.1.4322.573	24 Nisan 2003	1.1	Visual Studio .NET 2003	Windows Server 2003
2.0	2.0.50727.42	7 Kasım 2005	2.0	Visual Studio 2005	Windows Server 2003 R2
3.0	3.0.4506.30	6 Kasım 2006	2.0	Expression Blend	Windows Vista, Windows Server 2008
3.5	3.5.21022.8	19 Kasım 2007	2.0	Visual Studio 2008	Windows 7, Windows Server 2008 R2
4.0	4.0.30319.1	12 Nisan 2010	4.0	Visual Studio 2010	N/A
4.5	4.5.50709.17929	15 Ağustos 2012	4.5	Visual Studio 2012	Windows 8, Windows Server 2012
4.5.1	4.5.50938.18408	17 Ekim 2013	4.5	Visual Studio 2013	Windows 8.1, Windows Server 2012 R2

## .Net Framework Versiyonları







## Visual Studio Kurulumu



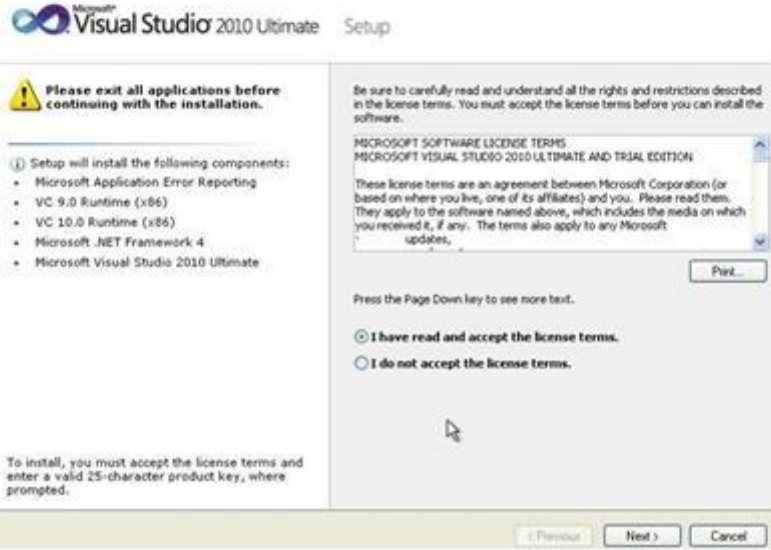
## Visual Studio Kurulumu



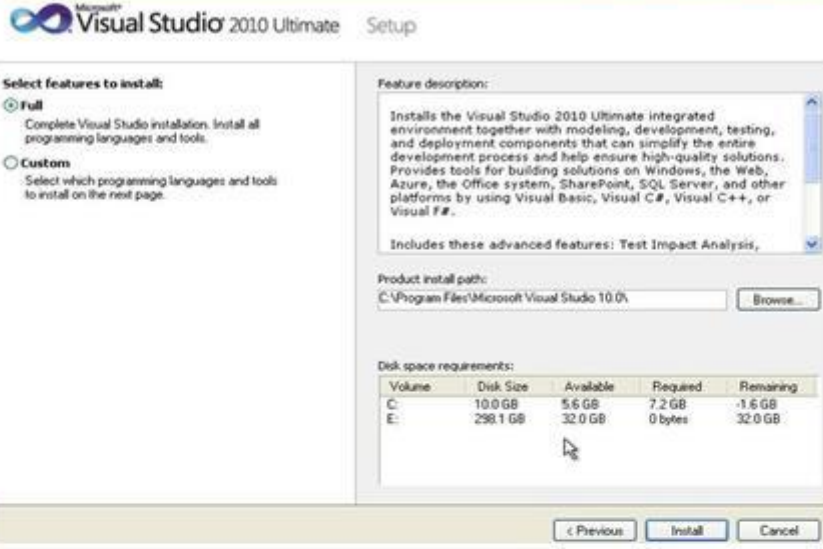
## Visual Studio Kurulumu



## Visual Studio Kurulumu



## Visual Studio Kurulumu



Microsoft Visual Studio 2010 Ultimate Setup

**Select features to install:**

- Full**  
Complete Visual Studio installation. Install all programming languages and tools.
- Custom**  
Select which programming languages and tools to install on the next page.

**Feature description:**

Installs the Visual Studio 2010 Ultimate integrated environment together with modeling, development, testing, and deployment components that can simplify the entire development process and help ensure high-quality solutions. Provides tools for building solutions on Windows, the Web, Azure, the Office system, SharePoint, SQL Server, and other platforms by using Visual Basic, Visual C#, Visual C++, or Visual F#.

Includes these advanced features: Test Impact Analysis,

**Product install path:**  
C:\Program Files\Microsoft Visual Studio 10.0\ [Browse...]

**Disk space requirements:**

Volume	Disk Size	Available	Required	Remaining
C:	10.0 GB	5.6 GB	7.2 GB	-1.6 GB
E:	298.1 GB	32.0 GB	0 bytes	32.0 GB

< Previous Install Cancel

## Visual Studio Kurulumu



Microsoft Visual Studio 2010 Ultimate Setup

**Installing Components:**

- Microsoft Application Error Reporting
- VC 9.0 Runtime (x86)**
- VC 10.0 Runtime (x86)
- Microsoft .NET Framework 4
- Microsoft Windows Installer 4.5 (x86) - Windows XP
- Microsoft Windows Installer 4.5 Update (x86) - Windows XP
- Microsoft Visual F# 2.0 Runtime
- Microsoft Visual Studio Macro Tools
- TFE Object Model (x86)
- .NET Framework 4 Multi-Targeting Pack
- Microsoft Visual Studio 2010 Ultimate
- Microsoft Web Deployment Tool (x86)
- Microsoft ASP.NET MVC 2 - Visual Studio 2010 Tools
- Microsoft ASP.NET MVC 2
- Microsoft Silverlight
- Microsoft Silverlight 2 SDK

Installing VC 9.0 Runtime (x86)...

< Previous Next > Cancel



## Formlar

Formun özelliklerine ulaşmak için this komutundan faydalanılabilir. Formlar SDI(Single Document Interface) ve MDI(Multiple Document Interface) olarak 2'ye ayrılır.

### 1. SDI FORMLAR

Herhangi bir forma bağlı olmaksızın kendi başlarına çalışırlar. Yeni bir projeye başlarken projede bulunan form ve sonradan eklenen formlar SDI formudur.

`this.Text="Muhasebe Programı Versiyon 1.0";`//formun başlığı

- FormBorderStyle Özelliği:
  - Sizeable: Formboyutlandırılabilir.
  - FixedSingle: Boyutları değiştirilemez
  - Fixed3D: Form 3 boyutlu görülür. Boyutları değiştirilemez.
  - FixedDialog: Dialog pencereleri oluşturmak için kullanılır.
  - FixedToolWindow: Ekranı kapla,simge durumu düğmeleri yoktur. Boyutlandırılmaz.
  - SizeableToolWindow: Ekranı kapla,simge durumu düğmeleri yoktur. Boyutlandırılabilir.
  - None: Sadece Alt-F4 ile çıkılabilen hiçbir özelliği olmayan formdur.  
`This.FormBorderStyle = FormBorderStyle.Fixed3D;`
- MaximizeBox,MinimizeBox: Ekranı kapla ve simge durumu düğmelerinin durumu belirlenir.
- ControlBox: False yapılırsa kontrol kutusu görünmez,boyutlandırılabilir ve sadece Alt F4 ile kapanır.
- ShowInTaskBar:Formun görev çubuğunda yer almaması için false yapılmalıdır.
- WindowState: Formun durumu değiştirilebilir yada durumu hakkında bilgi sahibi olunabilir.

`This.WindowState = FormWindowState.Minimized;`

`this.WindowState = FormWindowState.Maximized;`

`this.WindowState = FormWindowState.Normal;`

- StartPosition:

`this.StartPosition=FormStartPosition.CenterParent;`//form içinde bulunduğu formun ortasında açılır

`this.StartPosition = FormStartPosition.CenterScreen;`//form ekran ortasında açılır

`this.StartPosition = FormStartPosition.Manual;`//form tasarlanırken bulunduğu koordinatta açılır.

`this.StartPosition = FormStartPosition.WindowsDefaultLocation;`//formun

koordinatları windows tarafından belirlenir.

`this.StartPosition = FormStartPosition.WindowsDefaultBounds;`//formun

koordinatları ve boyutları windows tarafından belirlenir.

- TopMost: Başka bir programa dahi geçilse form hep üstte kalır.
- MaximumSize,MinimumSize:Formun en büyük ve en küçük boyutları.
- Opacity: Form saydamlaştırılabilir.
- ActiveControl: Form üzerinde o an aktif olan kontrol öğrenilebilir.
- Show(): Bu metod ile form gösterilir.
- Hide():Bu metod ile form gizlenir.
- Close():Bu metod ile form kapatılır.



## SDÜ Eğitim Öğretim Bilgi Sistemi Ders İçeriği

- Activate():Bu metod ile form aktif hale gelir.
- Load Olayı: Form ilk defa yüklenirken bu olay meydana gelir.
- Activated ve DeActivate Olayları:Programda birkaç form varsa bunlardan sadece biri aktiftir. Aktivitenin formlardan diğerine yada başka birprograma geçmesi durumunda aktiviteyi kaybeden formun Deaktivate, aktif olan formun is Activated olayları gerçekleştirir.
- Closing(object sender,System.CompenentModel.CancelEventArgs e)
- Form kapatılırken bu olay meydana gelir. E.Cancel=True; ifadesi formun kapatılmasını önler.

```
private void Form1_FormClosing(object sender, FormClosingEventArgs e)
{
    DialogResult c;
    c = MessageBox.Show("Çıkmak istediğinizden
emin misiniz?","Çıkış",MessageBoxButtons.YesNo,MessageBoxIcon.Information);
    if (c == DialogResult.Yes)
        e.Cancel = false;
    else
        e.Cancel = true;//Çıkışı durdur
}
```

Application.Exit();  
Environment.Exit(0); Closing olayına kod yazıldığında bu 2 metod ile program sonlandırılmamalıdır.Closing() olayının çalışması için formu close metodu ile kapatmalıyız.

- Resize Olayı: Formun boyutlarının değişmesi halinde meydana gelir.
- Programda hangi formun başlangıçta çalışacağı solution penceresindeki program.cs içindeki main() metodunda yazar.
- Bir butona basıldığında diğer form"a geçmek için aşağıdaki yol izlenmelidir.

```
private void button1_Click(object sender, EventArgs e)
{
    Form2 frm = new Form2();
    frm.Show();//Form birden fazla açılabilir
    frm.ShowDialog();//Form bir kez açılır.
}
```

- Açılış Formu İpucu: Eğer bir ana form kapanırsa programda sona erer. Programımızda bir Gıfre formu olsun , açılışta bu form gösterilir ve form kapandığında 2. form yerine geçmeden program kapanacaktır. Bu durumda ana form olarak form2 ayarlanmalı ve form2 nin load olayından form1(Gıfre formu) çağrılmalıdır.



## 2. MDI FORMLAR

MDI formlar, MDI Child formları içinde barındırır. Aynı formun çok sayıda kullanılması gereken durumlarda kullanılır.

`this.IsMdiContainer = true; // MDI form oluşturur.`

```
private void Form1_Load(object sender, EventArgs e)
{
    this.IsMdiContainer = true; // MDI form oluşturur.
    Form2 frm = new Form2();
    frm.MdiParent = this;
    frm.Show();
}
```

```
if (this.ActiveMdiChild != null) ActiveMdiChild.Close(); //aktif
MDIChild kapanır.
```

Aktif olan MDIChildForm dan başka bir MDI Child form a geçildiğinde **MDIChildActivate** olayı gerçekleşir.

```
private void Form1_MdiChildActivate(object sender, EventArgs e)
{
    if (this.ActiveMdiChild != null)
        this.Text = this.ActiveMdiChild.Text;
}
```



## Mesaj Pencereleeri

### 1. MessageBox.Show(mesaj,başlık,düğmeler,simge,varsayılan\_düğme)

Düğmeler açağıdaki gibidir.

```
MessageBox.Show("merhaba", "başlığıımız",  
MessageBoxButtons.AbortRetryIgnore);//durdur-yeniden dene-yoksay  
MessageBox.Show("merhaba", "başlığıımız", MessageBoxButtons.OK);  
MessageBox.Show("merhaba", "başlığıımız", MessageBoxButtons.OKCancel);//tamam-  
iptal  
MessageBox.Show("merhaba", "başlığıımız", MessageBoxButtons.RetryCancel);  
MessageBox.Show("merhaba", "başlığıımız", MessageBoxButtons.YesNo);  
MessageBox.Show("merhaba", "başlığıımız", MessageBoxButtons.YesNoCancel);//evet-hayır-  
iptal
```

Simgeler Açağıdaki gibidir.

```
MessageBox.Show("merhaba", "başlığıımız", MessageBoxIcon.Error);//hata simgesi  
MessageBox.Show("merhaba", "başlığıımız", MessageBoxIcon.Question);//soru  
işareti simgesi MessageBox.Show("merhaba",  
"başlığıımız", MessageBoxIcon.Exclamation);//ünlem  
simgesi MessageBox.Show("merhaba", "başlığıımız",  
MessageBoxIcon.Information);//bilgi simgesi
```

Varsayılan düğme açağıdaki gibidir.

```
MessageBox.Show("merhaba", "başlığıımız", MessageBoxDefaultButton.Button1);
```

Üstteki kodda buton1'e tıklamakla enter'a basmak aynı görevi görecektir. Örneğin bir dosya silinecekse dosyayı sileyim mi? geklinde bir mesaj kutusu çıktığında default olarak buton2 seçilmelidir ki yanlılıkla silinme olmasın.

Dönen Değerler: MessageBox.Show metodundan kullanıcının hangi düğmeyi seçtiğini belirtir bir değer geri döner. Dönen değer DialogResult tipinden tanımlanmış bir değışkene aktarılıp hangi düğmenin seçildiğı açağıdaki değerler ile karşılaştırılarak anlaşılır. DialogResult.OK//tamam

DialogResult.Abort//durdur

DialogResult.Yes//evet

gibi.

## Hata Yakalama

Try-catch-finally blokları kullanılarak hata oluşturmaya müsait komutlar bu bloklara yazılırlar. Hataya müsait işlemler try bloğu içine yazılır. Eğer hata meydana gelirse catch bloğuna yazılan kod çalışacaktır.

Finally bloğu ise mutlaka çalışması gereken kodlar için kullanılır. Hata oluşsun veya oluşmasın finally bloğu altındaki kod çalışacaktır. Finally bloğu kullanılmaya da bilir. catch ifadesi olmadan da bir finally ifadesi kullanabilirsin. try ifadesinin üç farklı kullanım şeklini aşağıda görüyoruz.

- try bloğu, bir veya birden fazla catch bloğu
- try bloğu, bir veya birden fazla catch bloğu, finally bloğu
- try bloğu, finally bloğu

Son yapılandırma da program her hangi bir hata yakalamaz. Kullanıcı bir mesajla bilgilendirilir ve program sonlandırılır. Fakat program sonlandırılmadan finally bloğu çalıştırılır.

Catch bloğunun ne tür bir hatada çalışacağını aşağıdaki gibi bir yapı ile belirleyebiliriz.

*Catch(hata\_türü değişken)*

Örneğin Decimal türdeki bir sayı sifıra bölünmeye çalışıldığında **System.DivideByZeroException** hatası oluşur. Sadece bu hata aşağıdaki gibi yakalanabilir.

```
decimal sonuc;
try
{
    sonuc =
decimal.Parse(textBox1.Text)/decimal.Parse(textBox2.Text);
    MessageBox.Show(sonuc.ToString());
}
catch(System.DivideByZeroExceptionhata)
{
    MessageBox.Show("Sifira bölmeye çalışıyorsun");
}
catch (Exception hata)
{
    MessageBox.Show("Sayı girmediniz\r\n"+hata.Message);
}
```

Örnekte 2. catch bloğu özel bir hata türünü değil bütün hataları yakalayacaktır. Çünkü hata tipi olarak Exception verilmiştir. Hata ile ilgili bilgileri almak için catch bloğunda kullandığımız değişkenden faydalanabiliriz. Hata.message orijinal hata mesajını bize vermektedir.

Eğer taçma kontrolü açıksa taçma olduğunda **OverflowException** (taçma hatası) meydana gelir. Daha öncede anlattığımız bir çok programda klavyeden girilen değerleri bir integer değikene atarken Parse metodunu kullandık. Eğer klavyeden girilen değerler sayısal olmayan değerler içeriyorsa o zaman bir **FormatException** ( Biçim hatası ) meydana gelir. Bir Parse metodu ile string değikene bir null değer atanırsa bir **ArgumentNullException** (Null argüman hatası ) gelişir. Eğer bir dizinin sınırlarının dışında bir indeksleme yapmaya çalışırsanız veya uzunluğunun dışında bir atama yaparsanız **IndexOutOfRangeException** hatası meydana getirirsiniz.





## SDÜ Eğitim Öğretim Bilgi Sistemi Ders İçeriği

**Throw:** Throw hatayı fırlatma anlamına gelir. Hata mesajı verirken MessageBox.Show yerine throw kullanılabilir.

```
private void button1_Click(object sender, EventArgs e)
{
    int sayi1, sayi2;
    int sonuc;
    try
    {
        if (textBox1.Text == "")
            throw new ArgumentNullException("1.sayı", "bu değer boş
olamaz.");

        sayi1 = Int32.Parse(textBox1.Text);
        if (textBox2.Text == "")
            throw new ArgumentNullException("2.sayı", "bu değer boş
olamaz.");

        sayi2 = Int32.Parse(textBox2.Text);
        sonuc = sayi1 + sayi2;
        textBox3.Text = sonuc.ToString();
    }
    catch (FormatException ex)
    {
        MessageBox.Show(ex.Message);
    }
    catch (OverflowException ex)
    {
        MessageBox.Show(ex.Message + "\n Max Value for int32:" +
Int32.MaxValue.ToString());
    }
    catch (ArgumentNullException ex)
    {
        MessageBox.Show(ex.Message);
    } } }
```



## Kontrollere Ait Ortak Özellik-Olay ve Metodlar

**Name Özelliği:** Kontrollerin adının verildiği özelliktir.

**Enabled:** Bir kontrolün enabled özelliği false ise nesne pasif olur, yani görünür ama üzerinde işlem yapılamaz.

```
button1.Enabled = false;
```

**Visible:** Bu özellik true ise nesne form üzerinde görünür.

```
button1.Visible = false;
```

Çok sayıda nesneyi görünmez yapmak için nesneleri GroupBox içine alıp ,o halde gizlemek daha kolay olacaktır.

**Font:** Font'a ait işlemler properties kısmından değiştirilebilir. Çalışma anında özelliklerin değişmesi ise farklıdır. Bunun için font sınıfından bir nesne oluşturup font özelliğine atamak gerekir.

*Font(font\_adi, boyutu, Font\_stili)* Çeklinde parametreler alır.

Font\_stili için aşağıdaki değerler verilebilir. [FontStyle.Bold](#)

[FontStyle.Italic](#)

[FontStyle.Regular](#)

[FontStyle.Strikeout](#)

[FontStyle.Underline](#)

Örneğin textBox1 içindeki font" u Çu Çeklide değiştirebiliriz.

```
textBox1.Font = new Font("Tahoma", 12, FontStyle.Bold | FontStyle.Italic);
```

Bir nesnenin font özelliği diğer nesnelere içinde kullanılabilir. Aşağıda text2 nin font özelliği kalın yapılarak text1'e aktarılmaktadır.

```
textBox1.Font = new Font(textBox2.Font,FontStyle.Italic);
```

Eğer bir özelliği vermek veya o özellik varsa kaldırmak için | yerine ^ kullanılabilir.

Bir fontun orijinal ayarlarına dönmek için [ResetFont\(\)](#) özelliği kullanılır. [FontFamily](#) sınıfının [families](#) özelliği sisteme yüklü fontların listesini bir dizi olarak verir.

**Locked:** Tasarım anında true yapılırsa nesnelere yerinde sabitlenir.

**Location.X ve Location.Y:** Nesnenin form içindeki sol üst köşesinin koordinatlarını belirler.

```
button1.Location = new Point(10, 10);
```

**Size.Width ve Size.Height:** Nesnenin genişlik ve yüksekliğini belirler

```
button1.Size = new Size(100, 60);//geniçlik 100, yükseklik 50
```

**Left ve Top:** Bir nesne Form, panel, GroupBox içinde olabilir. Bu durumlarda içinde bulunduğu nesnenin sol üst koordinatları 0,0 olmak şartıyla Left ve Top bu noktaya olan uzaklıktır.



## SDÜ Eğitim Öğretim Bilgi Sistemi Ders İçeriği

**Right ve Bottom:** Nesnelerin sol ve alt koordinatlarını öğrenebiliriz ama değiştiremeyiz. Bu iki özellik left, top, height, width özelliklerine göre otomatik olarak değişir.

**Scale:** Bir kontrolün ve içinde bulunabilecek kontrolleri belirli oranda büyütür ya da küçültür. Küçültmek için 0-1 arasında bir değer vermek gerekir.

**ClientSize.Width ve ClientSize.Height:** Formun iç boyutlarını verir. **Width** ve **height** ise dış boyutları verir.

**AutoSize:** Bu özelliği true yaparak, Label ve Text box içine yazdığımız yazılar sığmazsa kontrolleri otomatik olarak büyütebiliriz. Textbox larda sadece yükseklik otomatik olarak ayarlanabilir.

**Anchor:** Form boyutları değiştikçe içindeki nesne anchor ile seçilen yöne olan uzaklığını korur. Bu özellik ile programın farklı çözünürlüklerde de aynı şekilde gösterilmesini istiyorsak kontrollerin tümünde anchor özelliklerinin Left,top,bottom ve right değerlerinin tümünü vermeliyiz.

**Dock:** Bir kontrolün formun bir kenarına alınmasını sağlar. Anchor özelliğine benzer fakat farkı koordinatlardan bağımsız olmasıdır. DockPadding özelliği ile dock işleminde kenarlara olan uzaklığı belirleyebiliriz.

**SendToBack ve BringToFront:** Bazı kontroller üst üste yerleşmişse SendToBack ile bu kontrol geriye gönderilebilir. BringToFront ile de öne alınabilir.

**BackColor:** Nesnenin zemin rengini belirler. ResetBackColor ile varsayılan renge döner.

**ForeColor:** Nesnenin yazı rengini belirler. ResetForeColor ile varsayılan renge döner.

```
textBox1.ForeColor = Color.Blue;
```

Renk seçimini kullanıcıya yaptırmak istersek **ColorDialog** kontrolü kullanılabilir. Ekrana bir adet ColorDialog kontrolü yerleştirip aşağıdaki kodu buton içine yazarsak form rengi seçtiğimiz renk olarak değişecektir.

```
if (colorDialog1.ShowDialog() == DialogResult.OK)
    this.BackColor = colorDialog1.Color;
```

**Color.FromArgb (kırmızı,yeşil,mavi)** fonksiyonu ile kendi renklerimizi oluşturabiliriz.

```
textBox1.BackColor = Color.FromArgb(123, 88, 22);
```

**Color.FromArgb(alpha,renk)**//geçirgenliği belirleyebiliriz.

**Text ve TextAlign:** Kontrollerin üzerindeki yazı değiştirilip hizalanabilir.

```
textBox1.TextAlign = HorizontalAlignment.Center;//yazıyı ortalar
```

**TextChanged Olayı:** Kontrollerin text özellikleri, yani üzerinde yazan bilgi değiştiğinde bu olay meydana gelir.

**Image ve ImageAlign:** Image özelliği bulunan PictureBox,buton, label gibi kontrollerin içeriğine resim yükleyebiliriz. ImageAlign ile de resmi hizalayabiliriz.

```
pictureBox1.Image = Image.FromFile("c:\\windows\\winnt.bmp");
```



# SDÜ Eğitim Öğretim Bilgi Sistemi Ders İçeriği

**ImageList ve ImageIndex:** Form a imagelist kontrolü yerleştirip, bu kontrolün images özelliğini kullanarak resimler ekleyebiliriz. Daha sonra bu resimlere her hangi bir kontrolden erişmek için önce kontrolün ImageList Özeliğinden ImageList1 seçilir , daha sonra ImageIndex özelliği ile de resim seçilir.

## Textbox

### CharacterCasing:

```
textBox1.CharacterCasing = CharacterCasing.Upper;//text içindeki yazılar büyük harfe çevrilir.
```

```
textBox1.CharacterCasing= CharacterCasing.Lower;
```

```
textBox1.CharacterCasing= CharacterCasing.Normal; MaxLength:
```

```
textBox1.MaxLength = 5;//text içine max 5 karakter yazılabilir.
```

### PasswordChar:

```
textBox1.PasswordChar = '*';
```

### ReadOnly:

```
textBox1.ReadOnly = true;//Sadece okunabilir.
```

### Undo:

```
textBox1.Undo();//text kutusu içine yazılanlar bir aşamalı olarak geri alınır.
```

**CanUndo:** Geri alınacak bir işlem olup olmadığı anlaşılabilir.

```
if (textBox1.CanUndo)
```

```
    textBox1.Undo();
```

```
    else
```

```
        MessageBox.Show("Geri alınamaz");
```

**Modified:** Text kutusundaki bilginin değişip değişmediği kontrol edilebilir.

### Multiline:

```
textBox1.Multiline = true;//Çok satırlı text box
```

Çok satırlı text bilgilerini öğrenmek için Lines[] kullanılabilir. Lines[] string türünden tanımlanmış bir dizidir. Bu dizinin elemanları ile Text kutusu içindeki satırlara tek tek ulaşılabilir.

```
private void button1_Click(object sender, EventArgs e)
```

```
{
```

```
    MessageBox.Show("Bu text kutusunda" +
```

```
(textBox1.Lines.GetUpperBound(0)+1).ToString() + "satır var");
```

```
    for (int i = 0; i <= textBox1.Lines.GetUpperBound(0);i++)
```

```
        MessageBox.Show((i+1).ToString()+".satır="+
```

```
textBox1.Lines[i]);
```

```
}
```

**AcceptsTab:** Bu özellik true yapıldığında text içinde tab tuşna basılınca başka kontrole atlamaz, boşluk bırakır.

**AcceptsReturn:** AcceptButton ile bir komut düğmesi ayarlanmıŞsa enter"a basılınca düğme tıklanmıŞ gibi olur. AcceptsReturn true yapılırsa bu özellik yerine, text içinde bir alt satıra geçer.

**WordWrap:**MultiLine true olduğunda etkin olur ve text kutusu içindeki sığmayan satırları bir alt satıra indirir.

**ScrollBars:** MultiLine true olduğunda etkin olur ve yazıya kaydırma çubukları ekler.

**Seçili Bölge özellikleri:** textBox1.SelectAll();//tümünü seçer

```
textBox1.SelectionLength>0//seçili bölge varsa
```

```
textBox1.SelectedText//seçilibölgeyi ifade eder
```



# SDÜ Eğitim Öğretim Bilgi Sistemi Ders İçeriği

```
textBox1.SelectionStart//seçili kısmın başlangıç noktasını  
belirler  
textBox1.HideSelection//false ise kontrol başka bir nesneye  
geçsede seçilen kısım görülür.  
textBox1.Select(0,3)//ilk 3 harfi seçmek için kullanılır.
```

## Label

```
label1.BackColor = Color.Transparent;//Bir resmin üzerine transparan bir label yerleştirmek uygun olacaktır.
```

### UseMnemonic:

```
label1.Text = "Öğrencinin &Adı Soyadı:";//A harfi kısa yol tuşu yapılır. Alt-A babilması durumunda tabindex i Label dan bir büyük olan elemana klavye kontrolü konumlanır. Bir butonun textinde &Ekle yazarsa Alt-E ile bu butona tıklanmış gibi olacaktır.
```

## Button

Bir düğmenin enter tuşu ile çalışması için **AcceptButton**, ESC tuşu ile çalışması için **CancelButton** özellikleri ayarlanmalıdır.

## Checkbox

**Checked:** Check kutusunun işaretli olup olmadığı bu özellik ile öğrenilebilir.

**CheckedChanged:** Check kutusunun işareti değiştiğinde bu olay meydana gelir.

**AutoCheck:** false yapılırsa kullanıcı tıklayarak işareti değiştiremez.

**TreeState:** Kararsız kalınan durumlar için checkbox ın bu özelliği true yapılarak 3 alternatif yakalanabilir. İşaretli ,işaretsiz ve kararsız.(kullanıcının kredi alıp almaması bilinmiyorsa gibi.) Bu durum hakkında bilgi almak için CheckState özelliği kullanılır.

```
checkBox1.CheckState = CheckState.Checked;//işaretli  
checkBox1.CheckState = CheckState.Unchecked;//işaretsiz  
checkBox1.CheckState = CheckState.Indeterminate;//kararsız
```

## Radiobutton

Checkbox ta olduğu gibi Checked özelliği ile bir rb un seçili olup olmadığını öğrenebilir, rb işaretlendiğinde yapmasını istediğimiz kodu da CheckedChanged olayına yazabiliriz.



# SDÜ Eğitim Öğretim Bilgi Sistemi Ders İçeriği

## Panel

GroupBox a göre en önemli avantajı kaydırma çubuklarını desteklemesidir. AutoScroll özelliğini true yaparak kaydırma çubukları ekleyebiliriz. BorserStyle özelliğini Fixed3D yaparak hoş bir efekt verebiliriz.

## TabControl

**TabPage özelliği:** ile yeni sayfalar eklenebilir. Tab sayfaları kontrol içine sığmazsa multiline özelliğini true yaparak alt satırdan devam ettirebiliriz.

**Appearance özelliği:** ile tab başlıklarının Çeklini belirleyebiliriz.

**Imageindex:** ile bir imagelist resmini sayfa başlığında gösterebiliriz.

**SelectedTab:** ile aktif olan tab ile ilgili bilgiler öğrenilebilir.

**SelectedIndexChanged**(Object sender, System.EventArgs e): Kullanıcı tab sayfaları arasında geçtiğinde bu olay gerçekleşir.

## Listbox

### Items:

- listBox1.Items.Add("Ali");//listenin sonuna eleman ekler.
- string[] x = { "1","2","3"};  
listBox1.Items.AddRange(x);//Bir grup eleman tek seferde eklenebilir.
- string[] x = { "1","2","3"};  
listBox1.Items.AddRange(x);  
listBox1.Items.Insert(1, "6");//Araya eleman ekler.2. eleman 6 olur.
- listBox1.Items.RemoveAt(0);//numarası verilen elemanı silinir.
- listBox1.Items.Remove ("ali");//içeriği verilen elemanı silinir.
- listBox1.Items.Remove(listBox1.SelectedItem);//Seçili elemanı siler.
- listBox1.Items.Clear();//listedeki tüm elemanları siler.
- label1.Text = listBox1.Items.Count.ToString();//Listedeki eleman sayısını verir.
- listBox1.Items[2] = "veli";//index'i 2 olan elemanı veli olarak değiştirir.
- int yer;  
yer = listBox1.Items.IndexOf("veli");//Bir elemanı aramak için kullanılır.Bulamazsa -1 değerini bulursa index no sunu verir. if (yer > 0)  
MessageBox.Show("Listede" + (yer + 1).ToString() + ".  
elemandır");  
else  
MessageBox.Show("Böyle bir eleman listede yok");
- string deger;



## SDÜ Eğitim Öğretim Bilgi Sistemi Ders İçeriği

```
deger = textBox1.Text.ToString();  
if (listBox1.Items.Contains(deger))//eleman listede varsa true
```

döndürür.

```
label1.Text = "Eleman listede var";
```

```
else
```

```
label1.Text = "Eleman listeye eklendi";  
listBox1.Items.Add(deger);
```

- Items.CopyTo(Dizi,No)//Listedeki elemanları bir diziye aktarmak için kullanılır.  
object[] x = new object[listBox1.Items.Count];//Liste boyutu kadar bir

liste tanımlandı

```
listBox1.Items.CopyTo(x, 0);//list1 içindeki elemanları ilk elemandan
```

itibaren x dizisine kopyala

- listBox1.Text="Ankara";//Ankara yazan elemanı bul ve seç Bu özellikte ayrıca seçili eleman içeriği öğrenilebilir.

```
MessageBox.Show("Seçili eleman:"+listBox1.Text);
```

```
listBox1.SelectedIndex = 5;//Listedeki 6. elemanı seçer
```

- listBox1.ClearSelected();//Listedeki seçili elemanların seçilmişliğini kaldırır.

**FindStringExact(aranan):** Liste içinde gelişmiş arama yapar. Contains ve IndexOf metodları tek kayıt ararken, bu metod liste içindeki elemanın tümünü arar.

```
int yer;  
string aranan;  
aranan = textBox1.Text;  
yer = listBox1.FindStringExact(aranan);  
if (yer < 0)  
    MessageBox.Show("Bulunamadı");  
else  
{  
    MessageBox.Show((yer+1).ToString()+".eleman");  
    listBox1.SelectedIndex = yer;  
}
```

**FindString:** Elemanın tümünde değil bir bölümünde arama yapar. Örneğin ali yazarak aratılırsa için de ali kelimesi ile bağlayan kayıtları bulur.

```
private void button1_Click(object sender, EventArgs e)  
{  
    string[] x={"Ali", "Alim", "canali", "Suzan"};  
    listBox1.Items.AddRange(x);  
}  
  
private void textBox1_TextChanged(object sender, EventArgs e)  
{  
    int yer = listBox1.FindString(textBox1.Text);  
    if (yer >= 0)  
        listBox1.SelectedIndex = yer;  
}
```

**SelectionMode:** Çok seçimli listelerde kullanılır

```
listBox1.SelectionMode = SelectionMode.One;//listeden sadece tek eleman seçilir
```

```
listBox1.SelectionMode = SelectionMode.MultiSimple;//Fare ile tıklanan her eleman seçilir, seçili olanların ki ise kalkar
```

```
listBox1.SelectionMode = SelectionMode.MultiExtended;//Shift veya ctrl tuşu ile birden fazla seçim yapılabilir.
```



# SDÜ Eğitim Öğretim Bilgi Sistemi Ders İçeriği

listBox1.SelectionMode = `SelectionMode.None`; //eleman seçilemez

**SelectedIndices:** Çok seçimli listelerde seçili elemanla ilgili özellikler için kullanılır

- `MessageBox.Show( listBox1.SelectedIndices.Count.ToString());` //seçili eleman sayısını verir.
- `MessageBox.Show( listBox1.SelectedIndices[1].ToString());` //seçili elemanlardan 2. sinin listede kaçınıcı eleman olduğunu verir.

**SelectedItems:**

- `listBox1.SelectedItems.Count`; //seçili eleman sayısını verir.
- `listBox1.SelectedItems[i]`; //seçili elemanlardan birinin içeriği öğrenilebilir.

**HorizontalScrollBar:** Listelere yatay kaydırma çubuğu eklenir. **ScrollAlwaysVisible:** Kaydırma çubuklarının sürekli görünür olmasını istediğimizde true yapılmalıdır.

**MultiColumn:** true ise liste dolduğunda elemanlar açağı doğru değil yana eklenirler.

**IntegralHeight:** true ise listbox, içindeki elemanları tam gösterebilmek için kendisini yeniden boyutlandırır.

## Combobox

**DropDownStyle:** Cb'ın tipi belirlenir.

**Text :** cb içine yazılan bilgi

**Items:** listbox taki items özellikleri ile aynıdır.

**SelectedIndex:** Seçili elemanın numarasını verir.

**SelectedIndexChanged:** Listede bir eleman seçildiğinde bu olay meydana gelir.

## Numerikupdown

**Value:** Nu içindeki sayı bu özellik ile öğrenilip değiştirilebilir.

**Maksimum ,Minimum:** Sayının alabileceği en büyük ve en küçük değerler ayarlanır. Default olarak 0-100 arasındadır.

**Increment:** Sayının kaçar kaçar artıp azalacağı belirlenir. Default olarak 1 dir. **DecimalPlaces:** Bu özelliikle kaç basamaklı ondalık sayı girileceği belirlenebilir. Bu durumda increment özelliğine virgülle sayı girilebilir.

**UpButton() ve DownButton():** `numericUpDown1.UpButton();` ile increment ile belirlenen değer kadar otomatik artar.

**ValueChanged:** Nu içindeki değer düğmelerle değiştirildiğinde bu olay meydana gelir.





## Trackbar

---

**Value:** tB içindeki değeri temsil eder.

**Maksimum ,Minimum:** Değerin alabileceği en büyük ve en küçük değerler ayarlanır. Default olarak 0-10 arasındadır.

**TickStyle:** Tb üzerindeki çizgilerin stilini belirleyebiliriz.

**TickFrequency:** Örneğin maksimum özelliği 10 TickFrequency özelliği 2 ise 5 adet çizgi bulunacaktır.

**Orientation:** Tb „in yatay yada dikey olması sağlanabilir.

Tb“in hareket ettirilmesi ile bu olay meydana gelir.

## Progressbar

---

**Value:** PB içindeki değeri temsil eder.

**Maksimum ,Minimum:** Değerin alabileceği en büyük ve en küçük değerler ayarlanır. Default olarak 0-100 arasındadır.

## Timer

---

Örneğin programımızda her 5 saniyede bir çalışıp belirli işleri yapması gereken kodlar varsa bu kodları timer altına yazabiliriz.

**Interval:** Timer“in çalışacağı milis cinsinden zaman periyodudur. Her 1 sn de çalışacak kod için 1000, her bir dakikada çalışacak kod için 60000 vs. olmalıdır.

**Enabled:** Timer“in çalışmaya başlaması için true yapılmalıdır.

**Tick(object sender, System.EventArgs e):**Timer kontrolünün interval özelliği ile belirtilen süre içinde bu olay meydana gelir.

## Datetimepicker

---

**Value:** Kontrolün göstereceği tarih bu özellik ile belirlenebileceği gibi, kullanıcının seçtiği tarihte bu özellik ile öğrenilir.

**MinDate,MaxDate:** Seçilebilecek tarih aralığı bu 2 özellik ile belirlenebilir. **Format:**

Bu özellik ile tarih farklı şekillerde gösterilebilir. `dateTimePicker1.Format =`

`DateTimePickerFormat.Short;`//kısatarih

`dateTimePicker1.MaxDate = DateTime.Now;`//Bugünün tarihinden büyük olmasınlar

## Monthcalendar

---

**CalendarDimensions:** Takvim üzerinde kaç ay gösterileceği belirlenebilir.

**MinDate,MaxDate:** Seçilebilecek tarih aralığı bu 2 özellik ile belirlenebilir.

**ShowWeekNumbers:** True yapılırsa takvim üzerinde hafta numaraları da gösterilebilir.

**ShowToday:** Bugünün tarihi özel renkte gösterilir.



# SDÜ Eğitim Öğretim Bilgi Sistemi Ders İçeriği

**ShowTodayCircle:**Bugünün tarihi daire içine alınabilir.

**MonthlyBodedDates:** Her ayın belli günleri takvim üzerinde koyu renkle gösterilir.

**AnnuallyBodedDates:** Her yılda belli günler koyu olarak gösterilebilir. Yıl dönümleri için idealdir.

**BodedDates:** Takvimde belli tarihlerin koyu renkli görünmesi sağlanabilir.

## Imagelist

Bir den fazla resmi bir arda tutmak için kullanılır. Images özelliği ile resimler belirlenebilir.

**ImageSize:** Varsayılan değeri 16\*16 dir. İstersek değiştirebiliriz.

**ColorDepth:** Varsayılan degeri Dept8Bit“tir. Resim kalitesi arttıkça bu değeri arttırabiliriz.

**Images[no]:** Listedeki resimlerden biri alınabilir.

**Images.AddStrip(Resim):** Bu metod bir resim dosyasında yan yana dizilmiş görüntülerin her birini ayrı bir resim olarak alabilir. Bu metod resmi parçalarken ImageSize.Width özelliğinin değerini baz alır. Resmin genişliği ImageSize.Width değerinin tam katı olmalıdır.

## OpenFileDialog

**FileName:** kullanıcının seçtiği dosyayı öğrenebiliriz.

**Title:** Diyalog penceresinin başlığında yazacak yazıyı belirtir.

`openFileDialog1.Title="Dosya Aç";`

**InitialDirectory:** diyalog penceresi açıldığında hangi klasörü göstereceği belirlenebilir.

```
openFileDialog1.InitialDirectory= "c:\\";
openFileDialog1.InitialDirectory=
```

`Application.StartupPath;//programın kayıtlı olduğu klasörü gösterir.`

**Filter:** “Dosyatürü\_ açıklaması | dosya\_türleri” şeklinde ifade edilir.

```
openFileDialog1.Filter= "Dökümanlar | *.DOC;*.RTF |" + "Resimler|
*.BMP;*.JPG;*.GIF |" + "Tüm Dosyalar|
*.*";
```

**FilterIndex:** Bağlangıçta hangi filtrenin etkili olacağı belirlenebilir.

```
openFileDialog1.FilterIndex= 2;//resimler filtresi aktiftir.
```

**DefaultExt:** Yaptığımız programın standart bir dosya uzantısı varsa, bu özelliği o uzantıya vererek, kullanıcıyı her seferinde uzantı yazmaktan kurtarabiliriz. `openFileDialog1.DefaultExt= "TXT";`

**CheckFileExists:** True yapılırsa dosyanın diskte olup olmadığı kontrol edilir. Yoksa hata mesajı verir ve başka dosya seçimi sağlar.

**CheckPathExists:** Kullanıcının yazdığı yolun doğruluğu kontrol edilir.

**FileOk(object sender, System.CompenentModel.CancelEventArgs e):** Kullanıcı bir dosya seçip aç düğmesine basınca henüz diyalog penceresi kapanmadan bu olay gerçekleşir. Bu olaya yazılacak kodla kullanıcının seçtiği dosyanın geçerliliği kontrol edilip e.cancel parametresine true değerini atayarak pencerenin kapanması durdurulabilir.



```
private void button1_Click(object sender, EventArgs e)
{
    openFileDialog1.Filter = "Resim Dosyaları" +
    "*.bmp;*.jpg;*.gif;*.tif;*.png";
    openFileDialog1.Title = "Açacağın dosyayı seç";

    if (openFileDialog1.ShowDialog() == DialogResult.OK)
        pictureBox1.Image =
        Image.FromFile(openFileDialog1.FileName);
        pictureBox1.Image = Image.FromFile(openFileDialog1.FileName);
}

private void openFileDialog1_FileOk(object sender, CancelEventArgs e)
{
    try
    {
        pictureBox1.Image =
        Image.FromFile(openFileDialog1.FileName);
    }
    catch
    {
        e.Cancel = true;
        MessageBox.Show("Bu dosya geçerli değildir");
    }
}
```

**MultiSelect:** True yapılırsa kullanıcı birden fazla dosya seçebilir. Bu durumda dosyaları FileNames ile öğrenebiliriz.

## *Savefiledialog:*

---

OpenFileDialog ile aynı özelliklere sahiptir.

## *Colordialog:*

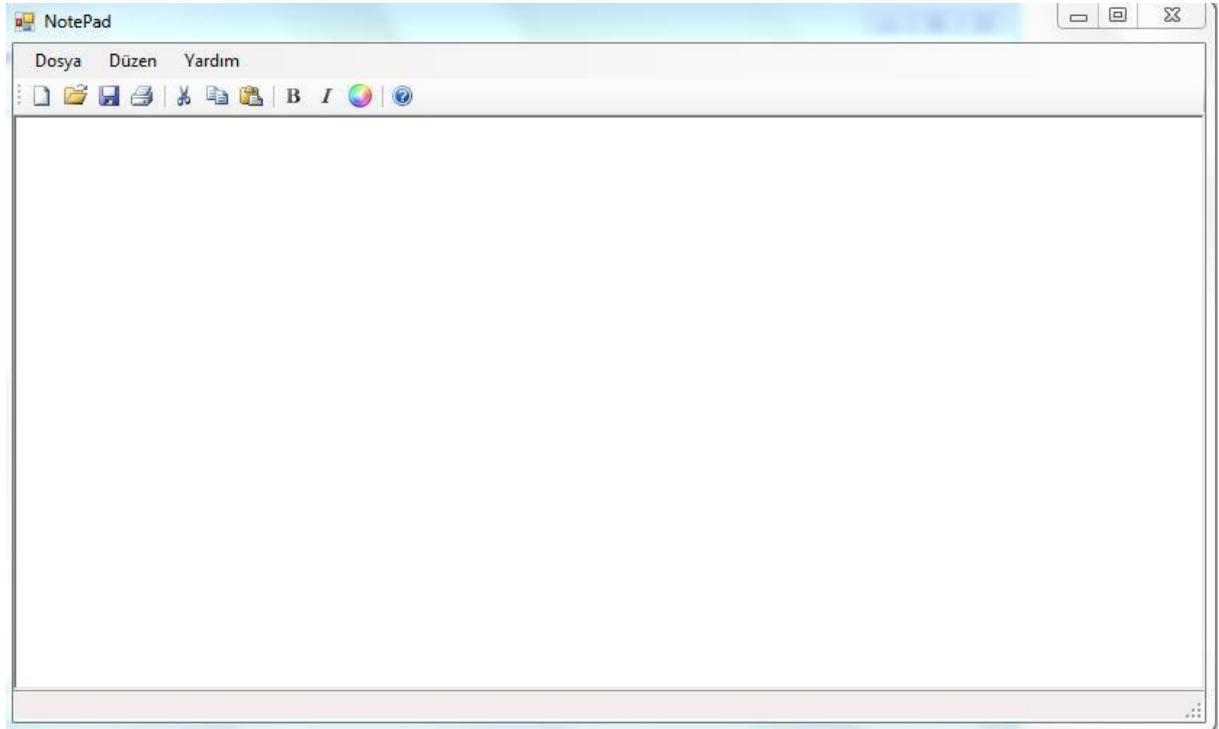
---

**Color:** kullanıcının seçtiği renk öğrenilebilir.

```
if (colorDialog1.ShowDialog() == DialogResult.OK)
    this.BackColor = colorDialog1.Color;
```

## UYGULAMA: NOT DEFTERİ ARAYÜZ OLUŞTURULMASI

1. Forma menustrip ekleyelim. Şekildeki dosya –düzen.. menülerini ekleyelim. Menü içine çizgi koymak için – işareti koymanız yeterlidir. Menüümüzün name özelliğini notepadMenu olarak değiştirelim.
  - a. Menustrip elemanını forma bağlamak için formumuzun Properties(P) kısmından mainmenustrip özelliğini notepadMenu olarak seçelim
2. Formun name özelliğini anaForm, text özelliğini NotePad olarak değiştirelim. Form üzerine 1 adet statusstrip, toolstrip, richtextbox (dock özelliğini fill yapalım)
3. Statusstrip nesnesinden statuslabel ekleyelim(toolstripstatuslabel1)
4. ToolStrip'in sağ üst köşesindeki küçük ok'a tıklayarak insert standart items komutu ile standart düğmeler ekleyelim. Sonra bunların name özelliklerini(tsYeni..), toolTipText özelliklerini değiştirelim.
5. Standart dışı olan düğmeleri ise(kalın,italik,renk) toolstripten ekledikten sonra bu düğmelerinde name ve tooltiptext özelliklerini değiştirelim. Ancak bunlara image eklememiz gerekecektir. Bunun içinde düğmeyi seçtikten sonra image özelliği ile açılan pencereden Project resource file ile daha önce indirdiğimiz icon dosyasını seçelim.
6. ToolStrip özelliklerinden items ile butonların yerlerini değiştirebiliriz.
7. Menü elemanlarına kısayol atamak için elemanı seçip shortcutkeys özelliğini istediğimiz kısayol ile değiştirmeliyiz.
8. Şimdi sağ tuş menüsü için contextmenustrip ekleyelim.(kes-kopyala-yapıştır-font-renk) özelliklerini ekleyelim, name özelliklerini değiştirelim ve kısayollarıda ekleyelim.
9. Daha sonra richtextbox özelliklerinden contextmenuden az önce oluşturduğumuz contextmenunun ismini seçerek bağlayalım.





## KOD YAPISI

```
private void tsMenuYeni_Click(object sender, EventArgs e)
{
    //MessageBox.Show 5.yazım biçimini uyguluyoruz.
    //MessageBox.Show bize dialogResult türünden bir değer göndermektedir.
    //Biz yes,no ve cancel değerlerinin seçilmesi durumunda olacak olayları
    //aşağıda if bloğunda kodluyoruz.
    if (richTextBox1.Modified == true)
    {
        DialogResult dr;
        dr=MessageBox.Show("Belgede değişiklik yaptınız.Değişiklikleri
        kaydetmek istiyormusunuz?", "Yeni Belge", MessageBoxButtons.YesNoCancel);
        if (dr == DialogResult.Yes)
        {
            tsMenuKaydet_Click(sender, e);
            richTextBox1.Clear();
            dosyaAdi = "";
            statusStrip1.Text = "Yeni Belge";
        }
        else if (dr == DialogResult.No)
        {
            richTextBox1.Clear();
            dosyaAdi = "";
            statusStrip1.Text = "Yeni Belge";
        }
    }
}

private void tsMenuAc_Click(object sender, EventArgs e)
{
    DialogResult dr;
    dr= openFileDialog1.ShowDialog();
    if (dr == DialogResult.OK)
    {
        richTextBox1.LoadFile(openFileDialog1.FileName);
        dosyaAdi = openFileDialog1.FileName;
        int o = dosyaAdi.LastIndexOf("\\");
        toolStripStatusLabel1.Text= dosyaAdi.Substring(o + 1, dosyaAdi.Length
        - o - 1);
    }
    //Form üzerine openfiledialog ekleyelim.
    //FileName Belge1 olarak değiştirelim
    //filter özelliğini aşağıdaki gibi değiştirelim
    // Zengin Metin Biçimi(.rtf) | *.rtf | Bütün Belgeler(*.*) | *.*
    //InitialDirectory özelliğini C:\Users\Ogr.Gr-Sinan UGUZ\Documents
    //Title: NetPad
}

private void tsMenuKaydet_Click(object sender, EventArgs e)
{
    //öncelikle kod sayfasının üstünde dosyaAdi adlı bir değişken
    oluşturalım.(compenents kısmında)
    // private string dosyaAdi;
    //form_load olayına dosyaAdi=""; atayalım
```



# SDÜ Eğitim Öğretim Bilgi Sistemi Ders İçeriği

if (richTextBox1.Modified == true)

{



## SDÜ Eğitim Öğretim Bilgi Sistemi Ders İçeriği

```
        if (dosyaAdi != "")
        {
            richTextBox1.SaveFile(dosyaAdi);
            int o = dosyaAdi.LastIndexOf("\\");
            toolStripStatusLabel1.Text = dosyaAdi.Substring(o + 1,
dosyaAdi.Length - o - 1) + "kaydedildi.";
        }
        else
        {
            tsMenuFKaydet_Click(sender, e);
        }
    }
}

private void tsMenuFKaydet_Click(object sender, EventArgs e)
{
    //Form üzerine savefiledialog ekleyelim.
    //FileName Belge1 olarak değiştirelim
    //filter özelliğini aşağıdaki gibi değiştirelim
    // Zengin Metin Biçimi(.rtf)|*.rtf | Bütün Belgeler(*.*)| *.*
    //InitialDirectory özelliğini C:\Users\Ogr.Gr-Sinan UGUZ\Documents
    //Title: NetPad
    DialogResult dr;
    dr = saveFileDialog1.ShowDialog();
    if (dr == DialogResult.OK)
    {
        richTextBox1.SaveFile(saveFileDialog1.FileName);
        dosyaAdi = saveFileDialog1.FileName;
        int o = dosyaAdi.LastIndexOf("\\");
        toolStripStatusLabel1.Text = dosyaAdi.Substring(o + 1, dosyaAdi.Length
- o - 1) + "olarak kaydedildi.";
    }
}

private void tsMenuCikis_Click(object sender, EventArgs e)
{
    if (richTextBox1.Modified == true)
    {
        DialogResult dr;
        dr = MessageBox.Show("Belgede değişiklik yaptınız.Değişiklikleri
kaydetmek istiyormusunuz?", "Yeni Belge", MessageBoxButtons.YesNoCancel);
        if (dr == DialogResult.Yes)
        {
            tsMenuKaydet_Click(sender, e);
            Application.Exit();
        }
        else if (dr == DialogResult.No)
        {
            Application.Exit();
        }
    }
}
}
```



# SDÜ Eğitim Öğretim Bilgi Sistemi Ders İçeriği

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





## SDÜ Eğitim Öğretim Bilgi Sistemi Ders İçeriği

```
        richTextBox1.Cut();
    }

    private void tsMenuKopyala_Click(object sender, EventArgs e)
    {
        richTextBox1.Copy();
    }

    private void tsMenuYapistir_Click(object sender, EventArgs e)
    {
        richTextBox1.Paste();
    }

    private void tsMenuYazitipi_Click(object sender, EventArgs e)
    {
        //Form üzerine bir adet fontdialog ekleyelim.
        DialogResult dr;
        dr = fontDialog1.ShowDialog();
        if (dr == DialogResult.OK)
        {
            richTextBox1.SelectionFont = fontDialog1.Font;
        }
    }

    private void tsMenuRenk_Click(object sender, EventArgs e)
    {
        //Form üzerine bir adet colordialog ekleyelim.
        DialogResult dr;
        dr = colorDialog1.ShowDialog();
        if (dr == DialogResult.OK)
        {
            richTextBox1.SelectionColor = colorDialog1.Color;
        }
    }

    private void tsMenuHakkında_Click(object sender, EventArgs e)
    {
        string mesaj = "NetPad Programı 1.0\n\nBucak ZTYO\n\nHer hakkı saklıdır.
2012";
        MessageBox.Show(mesaj, "NetPad
Hakkında", MessageBoxButtons.OK, MessageBoxIcon.Information);
    }

    private void anaForm_Load(object sender, EventArgs e)
    {
        dosyaAdi = "";
    }

    private void toolStrip1_ItemClicked(object sender,
ToolStripItemClickedEventArgs e)
    {
        if (e.ClickedItem == tsBtnYeni)
        {
            tsMenuYeni_Click(sender, e);
        }
        else if (e.ClickedItem == tsBtnAc)
        {
            tsMenuAc_Click(sender, e);
        }
        else if (e.ClickedItem == tsBtnKaydet)
```



# SDÜ Eğitim Öğretim Bilgi Sistemi Ders İçeriği

```
{  
    tsMenuKaydet_Click(sender, e);
```



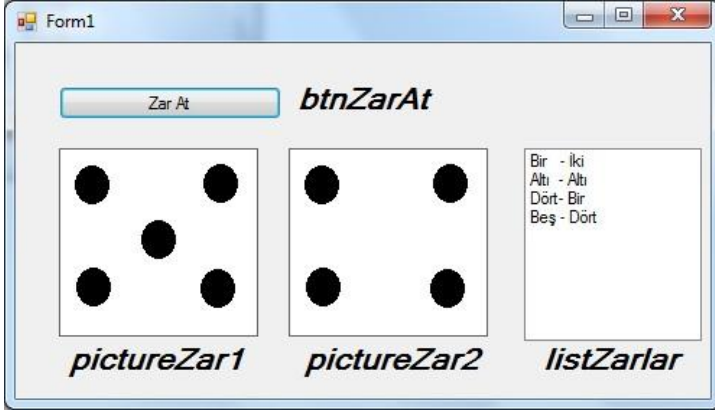
## SDÜ Eğitim Öğretim Bilgi Sistemi Ders İçeriği

```
}  
  
else if (e.ClickedItem == tsBtnKes)  
{  
    tsMenuKes_Click(sender, e);  
}  
else if (e.ClickedItem == tsBtnKopyala)  
{  
    tsMenuKopyala_Click(sender, e);  
}  
else if (e.ClickedItem == tsBtnYapistir)  
{  
    tsMenuYapistir_Click(sender, e);  
}  
else if (e.ClickedItem == tsBtnYardim)  
{  
    tsMenuHakkında_Click(sender, e);  
}  
else if (e.ClickedItem == tsBtnKalin)  
{  
    Font mevcutFont = richTextBox1.SelectionFont;  
    richTextBox1.SelectionFont= new  
Font(mevcutFont.FontFamily,mevcutFont.Size, mevcutFont.Bold ?  
FontStyle.Regular:FontStyle.Bold);  
}  
else if (e.ClickedItem == tsBtnRenk)  
{  
    tsMenuRenk_Click(sender, e);  
}  
}  
  
private void richTextBox1_TextChanged(object sender, EventArgs e)  
{  
    toolStripStatusLabel1.Text = "Yazılıyor.";  
}  
}
```



# SDÜ Eğitim Öğretim Bilgi Sistemi Ders İçeriği

## Çalışma Soruları



yazılmalıdır.

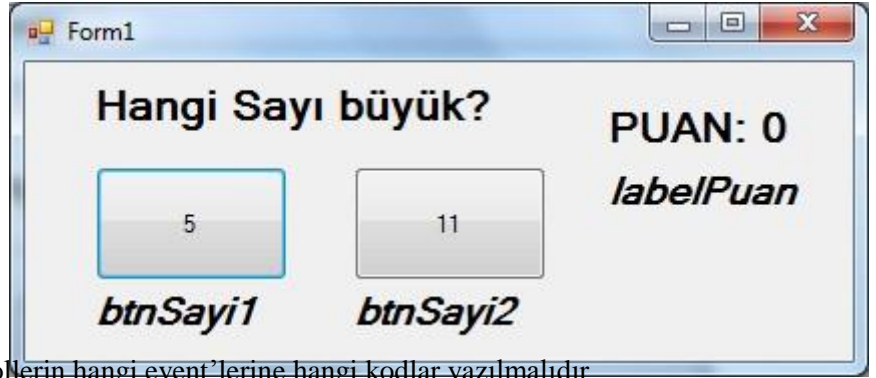
- C kök dizini içinde **Zarlar** klasöründe 6 tane resim bulunmaktadır. Resim dosyalarının adları şu şekildedir. **1.png, 2.png, 3.png, 4.png, 5.png ve 6.png** dir.

btnZarAt düğmesi tıklandığında rastgele olarak iki zar tutulup pictureZar1 ve pictureZar2 pictureBox nesnelere ilgili resimler yüklenecektir.

Her zar tutulduktan sonra kaç tutulduysa listZarlar listBox'ına eklenecektir.

Bu işlemlerin gerçekleşebilmesi için hangi kontrollerin hangi event'lerine hangi kodlar

- Form yüklendiğinde 1-20 arasında rastgele iki sayı tutulacak ve birinci sayı **btnSayi1'in** text özelliğine, ikinci sayı **btnSayi2'in** text özelliğine atanacaktır.
- Kullanıcı büyük sayıyı bulmaya çalışacaktır.
- Eğer kullanıcı büyük olan sayının olduğu doğru butona tıklarsa **labelPuan** daki değer 5 puan artacak.
- Eğer kullanıcı yanlış butona tıklarsa **labelPuan** daki değer 2 puan azalacak.
- Butonlardan ikisinden her hangi birine tıklanınca butonlardaki sayılar rastgele tekrar belirlenecektir.



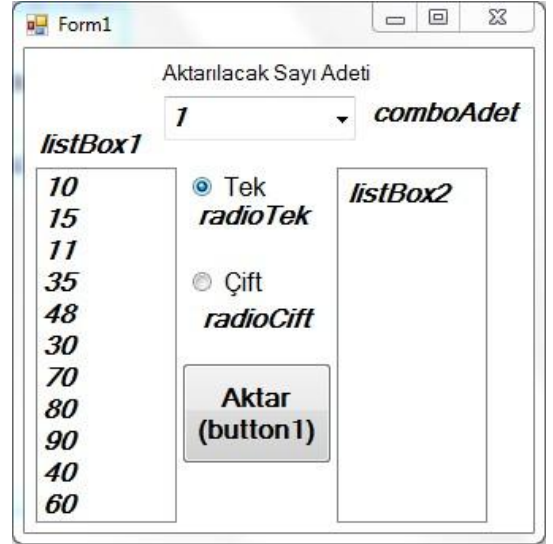
Bu işlemlerin

gerçekleşebilmesi için hangi kontrollerin hangi event'lerine hangi kodlar yazılmalıdır

- Form yüklendiğinde 1-100 arasında 100 tane rastgele sayıyı **listSayilar'** ekleyiniz.
- **btnSil** tıklandığında eğer **radio1** seçili ise **listSayilar** içerisindeki sayılardan tek olanları nı siliniz. Eğer **radio2** seçili ise **listSayilar** içerisindeki sayılardan çift olanlarını siliniz. Bu işlemlerin gerçekleşebilmesi için hangi kontrollerin hangi event'lerine hangi kodlar yazılmalıdır.



- comboAdet içinde sayılar vardır. Bu sayıların olduğunu düşünün.
- Eğer radioTek seçili ise button1 e tıklanınca listBox1 deki elemanlardan sıradan başlayıp comboAdet te seçilen değer kadar tek olan sayılar listBox2 e aktarılacak.
- Eğer radioCift seçili ise button1 e tıklanınca listBox1 deki elemanlardan sıradan başlayıp comboAdet te seçilen değer kadar çift olan sayılar listBox2 e aktarılacak.
- Aktarılan elemler listBox1 den silinmelidir.  
Bu işleri yapan komutları hangi kontrolün hangi olayında olması gerektiğini yazınız.



- Bankalardaki SıraMatik leri benzer bir program yazılacaktır. Bankada sadece 2 tane gişe olduğunu düşünülecek.
- btnSiraAl tıklanınca sıradaki verilecek numara labelSiraVer de kullanıcıya gösterilecek. Lütfen Örnekteki gibi Text sinde mesaj veriniz (Örnek □ Sıranız:10)
- btnGise1 tıkladığında sıradaki numara çağrılacak ve labelSuankiSira ve labelSiraGise1 de örnekteki gibi mesajlar verilecek.
- btnGise2 tıkladığında sıradaki numara çağrılacak ve labelSuankiSira ve labelSiraGise2 de örnekteki gibi mesajlar verilecek.  
Bu işleri yapan komutları hangi kontrolün hangi olayında olması gerektiğini yazınız.



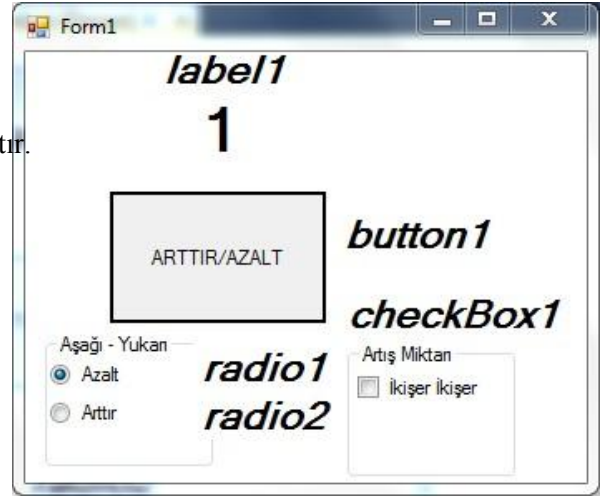


# SDÜ Eğitim Öğretim Bilgi Sistemi Ders İçeriği

Formda kullanılan kontrollerin adları form üzerinde verilmiştir.

- Fare **button1** üzerinde hareket edince **label1** deki değer değişecektir.
- radio1 seçili ise değer azalacak, radio2 seçili ise değer artacaktır.
- Label1 deki değerler -100(eksi yüz) ile 100 (yüz) arasında olmalıdır. Bu işlemlerin gerçekleşebilmesi için hangi kontrollerin

hangi event'lerine hangi kodlar yazılmalıdır.



Formda kullanılan kontrollerin adları form üzerinde verilmiştir.

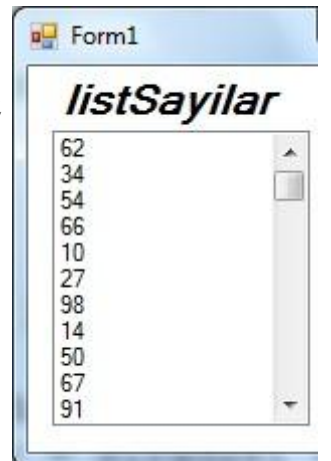
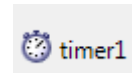
- listBox1 de bilgisayardaki resim yollarının **ekli olduğunu düşünün.**
- pictureBox1 tıklandığında listBox1 in seçili elemanından sonraki eleman seçili olacak ve seçili olan resim yolu pictureBox1 e yüklenecektir. (Eğer En sona ulaşırsa bir sonraki tıklamada ilk baştan tekrar başlamalıdır.)
- checkBox1 seçili ise bir sonraki eleman **RASTGELE** olarak belirlenmelidir.
- Bu işlemlerin gerçekleşebilmesi için hangi kontrollerin hangi event'lerine hangi kodlar yazılmalıdır.



3. Formda kullanılan kontrollerin adları form üzerinde verilmiştir.

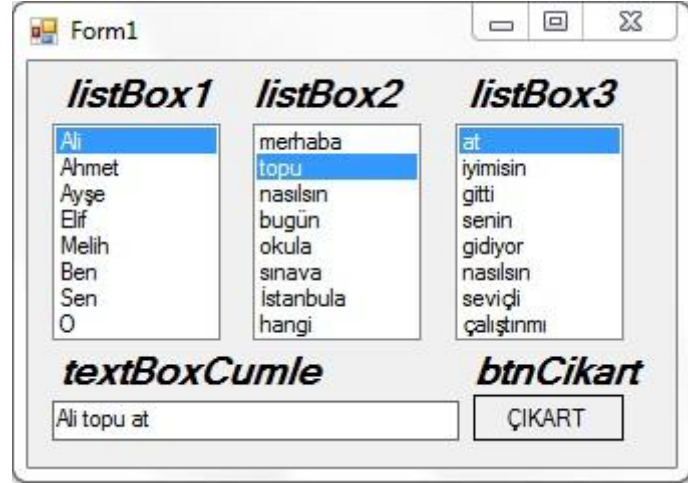
- listSayilar içerisinde rastgele sayılar olduğunu düşünün.
- Form yüklendiğinde timer1 çalışmaya başlayacaktır, timer1 ilk form yüklendiğinde 2 saniyede bir çalıştığı düşünülecektir.
- timer1 her çalışmasında listSayilar içerisindeki en son elemandan itibaren listSayilar içinde eleman kalmayana kadar silmeye başlayacaktır. Elemanların hepsi silinince timer1 in çalışması bitecektir.

Bu işlemlerin gerçekleşebilmesi için hangi kontrollerin hangi event'lerine hangi kodlar yazılmalıdır.



4. Formda kullanılan kontrollerin adları form üzerinde verilmiştir.

- Form açıldığında *listBox1*, *listBox2* ve *listBox3*' ün birinci elemanları seçili olacaktır.
  - *listBox1* *textBoxCumle* nin Text indeki cümlelerin **1.Kelimesini oluşturmakta**. *listBox2* 2.Kelimeyi ve *listBox3* 3.Kelimeyi oluşturmaktadır.
- ListBox lardan her hangi birinde **kelime seçimi değiştiğinde** *textBoxCumle* deki **cümle değişecektir**.
- *btnCikart* butonuna tıklanınca listBox lardaki seçili kelimeler çıkartılacaktır.
- Bu işlemlerin gerçekleşebilmesi için hangi kontrollerin hangi event'lerine hangi kodlar yazılmalıdır.



The screenshot shows a Windows form titled "Form1" with three list boxes and a text box. The list boxes are labeled *listBox1*, *listBox2*, and *listBox3*. *listBox1* contains the following items: Ali, Ahmet, Ayşe, Elif, Melih, Ben, Sen, O. *listBox2* contains: merhaba, topu, nasılsın, bugün, okula, sınava, İstanbula, hangi. *listBox3* contains: at, iyimisin, gitti, senin, gidiyor, nasılsın, seviçli, çalıştınımı. Below the list boxes is a text box labeled *textBoxCumle* containing the text "Ali topu at". To the right of the text box is a button labeled *btnCikart* with the text "ÇIKART".