

# WINDOWS FORM KONTROLLERİ I



## İÇİNDEKİLER

- Standart Kontroller
  - Label Kontrolü
  - Linklabel Kontrolü
  - Textbox Kontrolü
  - Button Kotrolü
  - Radiobutton Kontrolü
  - Checkbox Kontrolü
  - Checkedlistbox Kontrolü
  - Listbox Kontrolü
  - Combobox Kontrolü



## HEDEFLER

- Bu üniteyi çalıştıktan sonra;
  - Visual Studio kontrollerini tanıyabilecek,
  - Araç kutusu ile çalışmayı öğrenebilecek,
  - Standart kontrolleri kullanabilecek,
  - Kontrollerin özelliklerine hakim olabileceksiniz.



**Atatürk Üniversitesi**  
Açıköğretim Fakültesi

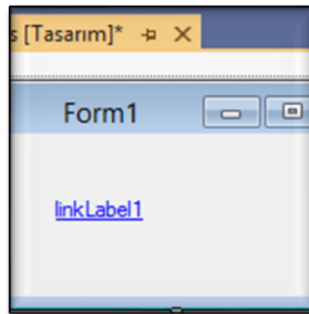
**GÖRSEL  
PROGRAMLAMA I**  
Öğr. Gör.  
**Daha ORHAN**

**ÜNİTE**  
**8**

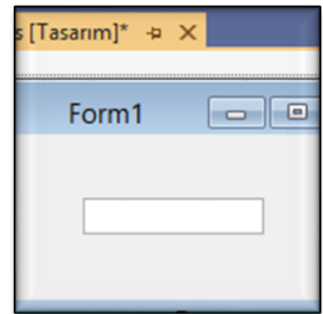
Label



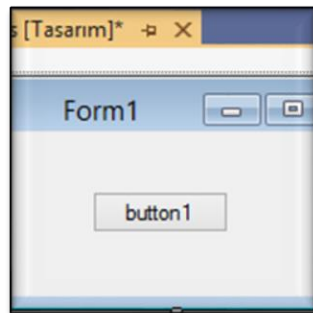
LinkLabel



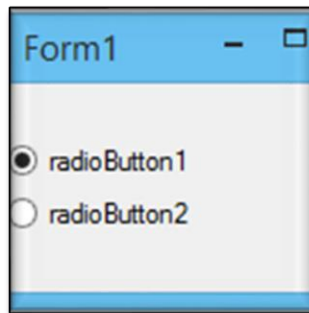
TextBox



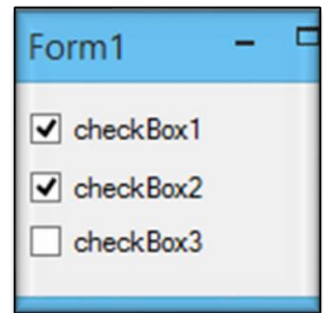
Button



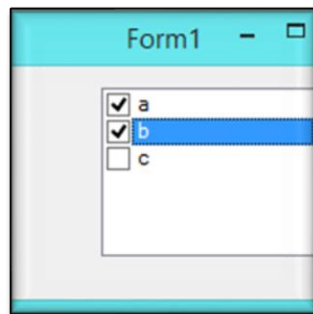
RadioButton



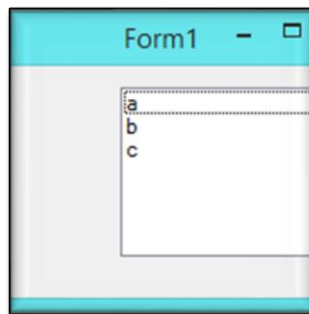
CheckBox



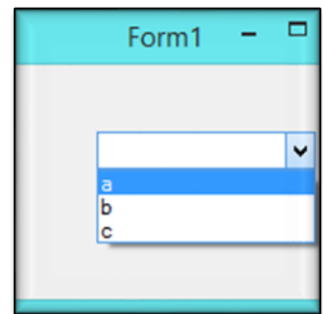
CheckedListBox



ListBox



ComboBox



## GİRİŞ

Günümüzde bilginin hızlı bir şekilde artması, elde edilmek istenen bilgiye ulaşmak için belirli bir çaba sarf edilmesini ortaya çıkarmaktadır. Bu nedenle dikkati yönlendirmek ve aranılan bilgiye kolay bir şekilde ulaşabilmek için hemen hemen her alanda görsel öğelerden, grafiklerden ve resimlerden faydalanılmaktadır. Yazılım dünyasında da bu durum benzer şekilde ilerlemektedir.



Araç kutusu kapalı ise  
Ctrl + Alt + X tuş  
kombinasyonu  
kullanılarak aktif hale  
getirilebilir.

Gerek programcılar gerekse kullanıcılar açısından yazılımların görsel öğelerle desteklenmesi kullanım kolaylığı sağlamaktadır. Resimlerin, grafiklerin ve diğer görsel öğelerin kullanımı ile geliştirilen programlamaya görsel programlama denilmektedir. Windows işletim sisteminin geliştirilmesi, kullanıcılar tarafından yaygın bir şekilde kullanılması ve konsol tabanlı uygulamaların yetersiz kalması görsel programlamaya olan ihtiyacı artırmaktadır.

Bu ünite Visual Studio 2019 platformu üzerinde bulunan C# programlama dili kullanılarak görsel öğelerle desteklenen Windows tabanlı uygulamalar üzerinde durulacaktır. Windows form uygulamalarının kullanıcı ara yüzü tasarlanırken *araç kutusunda* bulunan Windows form kontrollerinden istifade edilmektedir. *Araç kutusu* paneli Şekil 8.1.'de gösterildiği gibi 10 sekmeden oluşmaktadır. Bunlar;

- Tüm Windows Formları,
- Ortak Denetimler,
- Kapsayıcılar,
- Menüler ve Araç Çubukları,
- Veri,
- Bileşenler,
- Yazdırma,
- İletişim Kutuları,
- WPF Birlikte Çalışabilirlik,
- Genel.



Şekil 8.1. Araç Kutusu

Araç kutusu kapalı ise **Ctrl + Alt + X** tuş kombinasyonu kullanılarak veya görünüm menüsü üzerinden **Araç Kutusu** seçeneği işaretlenerek aktif hale getirilebilir. Windows formlara eklenebilecek bütün kontrollere araç kutusundan ulaşılabilir. Burada bir istisna bulunmaktadır: Formlarda bir kontrol olmasına rağmen araç kutusunda yer almamaktadır. Projemize yeni bir form eklemek için **Ctrl + Shift + A** tuş kombinasyonunu kullanılarak açılan **Yeni Öğe Ekle** penceresini kullanabiliriz.

Windows tabanlı uygulamalar geliştirilirken kullanıcı ara yüzünü tasarlamak için formlar üzerine araç kutusunda bulunan kontrolleri eklememiz gerekmektedir. Bu kontroller üç farklı şekilde formlar üzerine yerleştirilmektedir. Bunlar;

- Eklenilmek istenen kontrol araç kutusundan form üzerine sürükleyip bırak yöntemi kullanarak eklenebilmektedir.
- Araç kutusunda eklenilmek istenen kontrolün üzerine gelerek farenin sol tuşu ile çift tıklama işlemi gerçekleştirilip form içine eklenebilmektedir.
- Araç kutusundan eklenilmek istenen kontrol seçildikten sonra, fare imlecini form üzerine getirilerek istenilen boyutta kontrolün çizimi gerçekleştirilip kontrol form üzerine eklenebilmektedir.



Label kontrolü, genellikle içeriği sabit olan bilgilerin gösteriminde kullanılmaktadır.

Her kontrol için özellikler panelinde ortak seçenekler bulunsa da bütün kontroller belirli amaçlar doğrultusunda geliştirilmiştir. Bu üniteye birçok projede sıklıkla kullanılan kontroller ve o kontrollere ait özellikler, yöntemler ve olaylar anlatılacaktır. Bununla beraber kontrolleri form üzerine yerleştirildikten sonra etkin bir ara yüz tasarlamak için bu kontrolleri hizalamamız gerekmektedir. Bu nedenle ünite sonunda birden çok kontrolün seçimine ve belirli bir düzen içinde görünümüne yönelik bilgilere de yer verilecektir.

## STANDART KONTROLLER

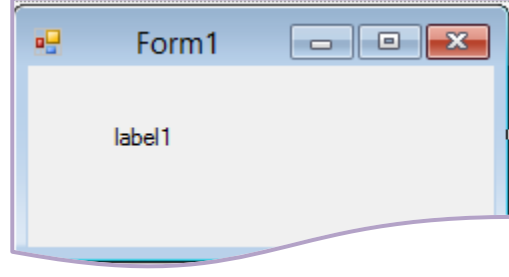
Windows form projelerinde gerek programcının gerek kullanıcının ihtiyaç duyduğu bilgilendirme, etkileşim, bilgi girişi gibi birçok işlem bulunmaktadır. Bu işlemleri gerçekleştirmek üzere Visual Studio .Net içerisinde kullanıma hazır bir şekilde geliştirilen pek çok kontrol mevcuttur. Bu üniteye hemen hemen her projede sıklıkla kullanılan kontroller üzerinde durulacaktır.

### Label Kontrolü

Bazı durumlarda form üzerinde bulunan nesnelerin ne işe yaradıklarını ifade etmek veya belirli konularda kullanıcıya bilgi vermek için etiketleme işleminin yapılmasına ihtiyaç duyulmaktadır. Böyle durumlarda **label** kontrolünün kullanımı ihtiyaca cevap verebilmektedir. Genellikle içeriği sabit kalıp üzerinde değişiklik yapılmayacak bilgilerin gösteriminde kullanılmaktadır.

Label kontrolü form üzerine eklendikten sonra özellikler panelinde bulunan **Text** özelliği sayesinde görüntülenecek metin düzenlenebilmektedir. Bu işlem genellikle proje çalıştırılmadan önce sabit bir şekilde ayarlanmaktadır. Ancak uygulamanın özelliğine bağlı olarak kod editörü yardımıyla gerekli etkileşim sağlanarak metin içeriği, punto değeri, yazı tipi gibi özellikler

değiştirilebilmektedir. Form üzerinde bulunan Label kontrolüne ait ekran görüntüsü Şekil 8.2.'de gösterilmektedir.



Şekil 8.2. Label Kontrolü

## LinkLabel Kontrolü

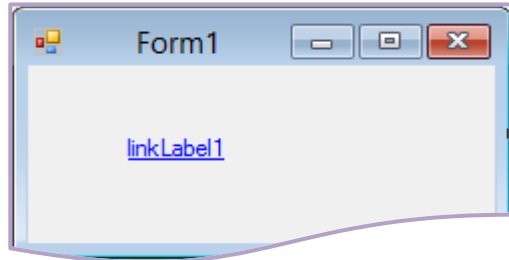


Form üzerinde görüntülenmesi istenilen bağlantı metni, linklabel kontrolünün **Text** özelliğine yazılmalıdır.

Projelerimizde bazı durumlar için web sitesine bağlantı yapılması gerekebilir. Böyle durumlarda **LinkLabel** kontrolü kullanılmaktadır. Form üzerinde görüntülenmesi istenilen bağlantı metni, linklabel kontrolünün **Text** özelliğine yazılmalıdır. Ancak text özelliğine yazılan metin bağlantının yapılmasını sağlamamaktadır. Bu alan sadece kullanıcının görüntülediği isimdir. LinkLabel kontrolüne tıklandığı zaman yönlendirilecek olan bağlantı adresi kod editöründe belirtilmelidir. Özellikler panelinde bağlantı adresini yazacağımız bir alan bulunmamaktadır.

LinkLabel kontrolünü formumuza ekledikten sonra kontrol üzerinde çift tıklayarak kod editöründe en sık kullanılan **LinkClicked** metodunu açabiliriz. Burada süslü parantezler içindeki kısımda **System.Diagnostics** isim alanı içinde bulunan **Process** sınıfındaki **Start()** metodunu kullanarak bağlantımızı gerçekleştirebiliriz. Start metodunda bulunan parantezler içine ziyaret edilecek Web sitesinin adresi **çift tırnak ("" )** içinde yazılması gerekmektedir.

Windows işletim sisteminde bağlantı metinleri genellikle mavi renkle gösterilmektedir. Form üzerine eklediğimiz LinkLabel kontrolü de varsayılan değer olarak mavi renkle bizi karşılamaktadır. Değişiklik yapılmadan form üzerine eklenen Linklabel kontrolünün ekran görüntüsü Şekil 8.3.'de verilmiştir.



Şekil 8.3. Linklabel Kontrolü

Hazırladığımız projelerde bu renk üzerinde değişiklik yapmak için LinkLabel kontrolünün özellikler panelindeki **LinkColor** seçeneğini kullanabiliriz. LinkColor seçeneğine tıkladığımızda **Özel**, **Web** ve **Sistem** olmak üzere üç farklı sekmenin bulunduğu bir panel karşımıza gelmektedir. Bu alan üzerinden tasarımımıza uygun bir renk tercihinde bulunabiliriz. Ayrıca projelerimizde tıklanmış bağlantı metni ile

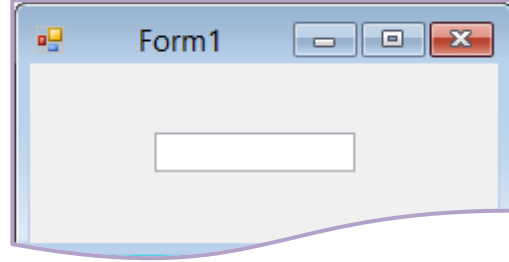
tıklanmamış bağlantı metnini birbirinden ayırt etmek için *LinkVisited* özelliğini *True* olarak ayarlamamız gerekmektedir. Bu sayede tıklanan bağlantı metinleri mor tonda gösterilecektir. Tıklanan bağlantı metninin rengini belirlediğiniz bir tonda ayarlamak istiyorsanız *VisitedLinkColor* özelliğini değiştirmeniz gerekmektedir.

## TextBox Kontrolü



TextBox kontrolünün *ReadOnly* özelliği *True* konumunda olduğu zaman TextBox içerisindeki bilgiler okunabilir ama bilgi girişi yapılamaz.

Programcı tarafından görüntülenmesi istenen bilgilerin yer alabileceği ve kullanıcı tarafından bu bilgilerin değiştirilerek bilgi girişi yapabileceği metin kutularıdır. Özellikler panelinde bulunan *Text* özelliği sayesinde 255 karakteri aşmamak üzere görüntülenecek metin girişi yapılabilir. Özellikler panelinin haricinde TextBox kontrolüne değer aktarmak için *çift tırnak ("")* kullanmak gerekmektedir. Form üzerine eklenmiş TextBox kontrolünün ekran görüntüsü Şekil 8.4. verilmektedir. Windows form uygulamalarında textbox kontrolü sıklıkla kullanıldığı için bu kontrole ait özellikleri, metotları ve olayları incelemenin faydalı olacağı düşünülmektedir.



Şekil 8.4. Textbox Kontrolü

## TextBox kontrolünün özellikleri

Textbox kontrolünün içerisindeki metnin kullanıcı tarafından değiştirilmesini istemediğimiz durumlar olabilir ya da projemizde bulunan şartlara göre bilgi girişinin açık veya kapalı olmasını sağlayabiliriz. Bu durumda kontrolün *ReadOnly* özelliğini ayarlamamız gerekmektedir. *ReadOnly* özelliği *True* konumunda olduğu zaman TextBox içerisindeki bilgiler okunabilir ama bilgi girişi yapılamaz. TextBox içerisine bilgi girişi için izin vermek istiyorsak *ReadOnly* özelliğini *False* olarak ayarlamamız gerekmektedir.

TextBox içerisine girilen bilgilerin başkaları tarafından görüntülenmemesi ve gizliliğin kullanıcı ara yüzünde sağlanması istenebilir. Bu durum genellikle kullanıcının şifre bilgisini girerken ihtiyaç haline gelmektedir. Her bir karakterin özel bir simge ile gösterilmesi için *PasswordChar* özelliği kullanılır. Bu özelliğin yanındaki alana girilen simge TextBox içerisine girilen her karakterin yerini alacaktır.

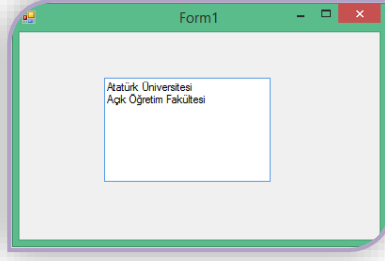
*CharacterCasing* özelliği ile TextBox içine yazılan yazıların büyük ya da küçük harflerle yazılması mümkün olmaktadır. Bu özelliğe ait üç seçenek bulunmaktadır. Varsayılan değer olan *Normal* seçeneğinin yazının olduğu haliyle kalmasını sağlamaktadır. *Upper* seçeneği işaretlendiğinde girilen metnin büyük harflere dönüştürülerek görünmesini sağlamaktadır. *Lower* seçeneği işaretlendiğinde ise girilen metnin küçük harflere dönüştürülerek görüntülenmesini sağlamaktadır.

TextBox içerisine girilen metinlerin yazı tipini ayarlamak için özellikler panelinde bulunan **Font** seçeneğini kullanabiliriz. Bu özelliğe ait butona tıkladığımızda karşımıza **Yazı Tipi** penceresi gelecektir. Bu pencereden istediğimiz seçenekleri belirleyerek tamam butonuna tıkladıktan sonra TextBox üzerindeki metinler ayarladığımız özelliklerde görüntülenecektir. Ayrıca Textbox içerisine girdiğimiz metinlerin yazı rengini özellikler panelinde bulunan ForeColor seçeneği ile gerçekleştirebiliriz.



TextBox kontrolünün çok satırlı olmasını Multiline özelliği ile ayarlayabiliriz.

Uzun miktarda bilgi girişinin yapılacağı durumlar için TextBox kontrolünün tek satır olarak kalması girilen metnin tamamını görüntüleme noktasında yetersiz kalabilir. Böyle bir durum karşısında kontrolün **Multiline** özelliğini **True** olarak ayarlamamız daha fazla satır halinde bilgilerimizi görüntülemeye imkan sağlayacaktır. MultiLine özelliği true olarak ayarlanmış textbox kontrolüne ait ekran görüntüsü Şekil 8.5.'de gösterilmektedir.



**Şekil 8.5.** Multiline Özelliği True Olarak Ayarlanmış Textbox Kontrolü

Multiline özelliğini aktif hale getirmemize rağmen TextBox kontrolü form üzerinde ayarladığımız alanla sınırlı kalacaktır. Yani 5 satırlık bir alan tanımlı ise uygulamamıza 6 satırlık bir bilgi girişi yaptığımızda ilk satır görüntülenmeyecek ve giriş yaptığımız her satır için üstten bir satır eksilerek görüntülenecektir. Bu durumun sebebi **ScrollBars** özelliğinin varsayılan değer olarak **None** seçeneğinin aktif olmasından kaynaklanmaktadır. None seçeneği aktif olduğunda TextBox üzerinde kaydırma çubukları görüntülenmez. ScrollBars özelliğindeki diğer **Vertical** seçeneğini işaretlediğimizde dikey kaydırma çubuğu eklenmektedir. Bir diğer seçenek olan **Horizontal** seçeneğini işaretlediğimizde de yatay kaydırma çubuğu eklenmektedir. ScrollBars özelliğindeki son seçenek olan Both işaretlendiğinde de yatay ve dikey kaydırma çubuklarının ikisi de eklenmektedir. TextBox kontrolünün multiline özelliği etkinleştirildiğinde girişini yapacağımız bilgilerin uzunluğuna göre ScrollBars özelliğine ait seçeneklerden birini tercih edebiliriz.

**Lines** özelliği sayesinde Textbox kontrolü bir dizi değişkeni gibi işlev görebilmektedir. Bu özelliğe ait butona tıkladığımızda **Dizi Koleksiyonu Düzenleyicisi** penceresi karşımıza gelmektedir. Bu pencereden ilgili TextBox içerisine istediğimiz bilgileri aktarabiliriz. Multiline özelliğini aktif hale getirerek TextBox içerisine girdiğimiz her bir satırı dizi elemanı gibi kullanarak kolay bir şekilde bilgi aktarımı gerçekleştirebiliriz.

Windows form uygulamalarında kullanıcının veri girişi için belirli sınırlamalar getirmek istenebilir. Böyle bir durumda TextBox kontrolü için girebileceğimiz

azami karakter sayısı **MaxLengt** özelliği ile ayarlanabilmektedir. Bu özelliğe ait varsayılan değer 32767 olarak belirlenmiştir.

TextBox içerisine girilen metnin satır sonuna geldiği zaman bir alt satıra geçmesini istiyorsak **WordWrap** özelliğini True olarak ayarlamamız gerekmektedir. Bu özelliği kullanabilmemiz için ilgili TextBox'a ait Multiline özelliğinin de aktif olması gerekmektedir.

### TextBox kontrolünün metodları

TextBox kontrolü kes, kopyala, yapıştır vb. pano işlemlerini metotlar yardımıyla gerçekleştirebilmektedir. Sık kullanılan metotları inceleyelim.

- **SelectAll**: TextBox kontrolünün içerisindeki tüm yazıyı seçer.
- **Cut**: TextBox içerisindeki seçili alanı keserek başka bir yere yapıştırmak üzere hafızada tutar.
- **Copy**: TextBox içerisindeki seçili olan metni silmeden başka bir alan üzerine yapıştırmak için hafızaya alır. **Cut** ve **Copy** metotlarının çalıştırılabilmesi için seçim işleminin yapılması gerekmektedir. Bu durumu test edebilmek için TextBox kontrolünün **SelectionLength** özelliğinin sıfırdan büyük olma şartı kontrol edilebilir.
- **Paste**: Cut veya Copy metodu ile daha önceden hafızaya alınan metni, işaretlenen alana yapıştırmayı sağlar.
- **Undo**: Çalışma esnasında TextBox içerisinde yapılan en son işlemin geri alınmasını sağlar. Burada geri alınacak bir işlemin varlığını kontrol etmek için TextBox kontrolüne ait **CanUndo** özelliği sorgulanabilir.
- **Clear**: TextBox içerisindeki yazıları siler.

### TextBox kontrolünün olayları

Çalışma anında TextBox kontrolü üzerinde yapılan değişiklikleri izleme ihtiyacımız olabilir. Bu durumda TextBox kontrolüne ait olaylardan istifade etmemiz gerekmektedir. TextBox kontrolüne ait birçok olay mevcuttur. Bu olaylardan sık kullanılanları inceleyelim.

Form üzerine yerleştirilen TextBox kontrolüne çift tıkladığımız zaman varsayılan olarak **TextChanged** olayı kod editöründe açılacaktır. Bu alan üzerinde süslü parantezler içine yazdığımız kodlar TextBox kontrolünün içeriği değiştikçe işleyecektir.

Bazı Windows form uygulamalarında çok sayıda Textbox kontrolüne ihtiyaç duyabiliriz. Kullanıcı bu TextBox kontrolleri üzerinde tab tuşu ile gezinirken doğru bir şekilde ilerleyemeyebilir. Böyle bir karmaşıklığı önlemek adına kontrole ait **Enter** ve **Leave** olaylarını kullanmak faydalı olacaktır. Örneğin kullanıcının aktif olarak işlem yaptığı TextBox kontrolünü vurgulamak için, Enter olayının kod editöründe TextBox kontrolünün arka plan rengini değiştirebiliriz. Aynı TextBox kontrolünün Leave olayının kod editöründe de varsayılan arka plan rengini beyaz olarak ayarlayabiliriz. Böylece kullanıcı fare ile TextBox kontrolleri üzerinde gezinirken, aktif olan kontrol farklı bir renkle ön plana çıkacaktır. Bu durumu bir örnek üzerinde incelemenin faydalı olacağı düşünülmektedir.



TextBox kontrolüne çift tıkladığımız zaman varsayılan olarak TextChange olayı kod editöründe açılacaktır.

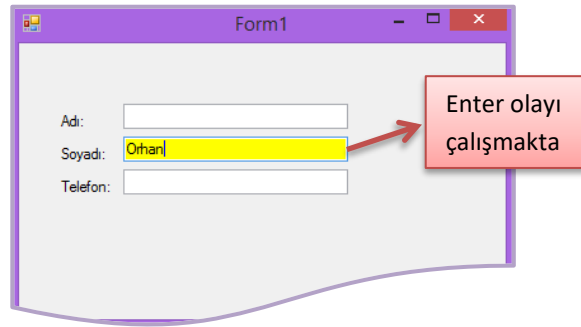




## Örnek

- Form üzerine yerleştirilen üç textbox kontrolünden ikinci sırada olan Soyadı: etiketine sahip textbox'ın Enter ve Leave olaylarına yazılan kodlar aşağıda verilmiştir. Bu kodlar yazıldıktan sonra programın çalışma anındaki ekran görüntüsü Şekil 8.6.'da görülmektedir.

```
private void textBox2_Enter(object sender, EventArgs e)
{
    textBox2.BackColor = Color.Yellow;
}
private void textBox2_Leave(object sender, EventArgs e)
{
    textBox2.BackColor = Color.White;
}
```



Şekil 8. 6. Enter ve Leave Olayları Hazırlanan Textbox



TextBox kontrolüne giriş yapıldığında Enter, çıkış yapıldığında da Leave olayı devreye girer.



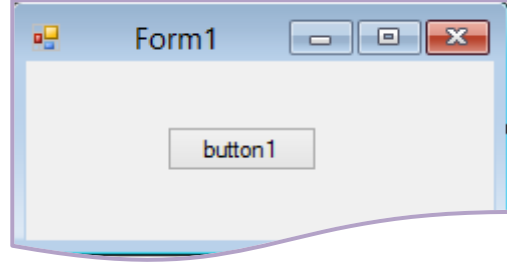
## Bireysel Etkinlik

- Yukarda verilen örnekte kodlar sadece TextBox2 için hazırlanmıştır. Bu örneğin aynısını diğer textbox kontrolleri için hazırlayıp çalışma anındaki hareketlerinize bağlı olarak kontrollerin nasıl bir etki oluşturduğunu gözlemleyiniz.

Kullanıcının aktif olarak işlem yaptığı TextBox kontrolünden ayrıldığı zaman yapılması gereken işlemlerin olduğu durumlar söz konusu olabilir. Örneğin girilen metindeki yazıların font özelliklerini değiştirmek veya bütün harfleri büyük harfe dönüştürmek isteyebilirsiniz. Böyle durumlarda TextBox kontrolünün *Validated* olayından istifade edebilirsiniz. Böylelikle kullanıcı metin girişini tamamlayıp TextBox kontrolünden ayrılmak istediği zaman yapılacak işlemler Validated olayının içerisine yazılarak işletilebilir.

## Button Kontrolü

Windows form uygulamalarının hemen hemen tamamında yer alan Button kontrolü herhangi bir işlemi veya komutu çalıştırmak amacıyla kullanılmaktadır. Form üzerine eklenen button kontrolüne ait ekran görüntüsü Şekil 8.7.'de görülmektedir. Button kontrolünü form üzerine yerleştirdikten sonra yazı tipi, boyutu, rengi gibi tasarım seçeneklerini özellikler paneli üzerinden düzenleyebiliriz. Ancak programın çalışma anında Button kontrolüne tıklanmasıyla birlikte yapılacak işlemleri kod editöründe ayarlamamız gerekmektedir. Kod editörüne geçmek için form üzerinde bulunan Button nesnesine çift tıklamamız gerekmektedir. Bu durumda kontrole ait en sık kullanılan *Click* olayı açılacaktır.



Şekil 8.7. Button Kontrolü

Button kontrolünün form üzerindeki ismini değiştirmek için özellikler panelinde bulunan *Text* seçeneğini kullanabiliriz. Yani bir buttonun ne işe yaradığını text özelliğini kullanarak ayarlayabileceğimiz gibi resimler ekleyerek görsel bir yolla ifade etmemiz de mümkündür. Button kontrolü üzerine resim yerleştirebilmek için özellikler panelinde bulunan image seçeneğini kullanmamız gerekmektedir. Bu seçeneğin yanındaki düğmeye tıkladığımızda karşımıza *Kaynak Seç* penceresi gelecektir. Bu pencereden *Yerel Kaynak* seçeneğini işaretledikten sonra *İçe Aktar* butonuna tıklayarak istediğimiz resim dosyasını seçip tamam butonuna tıklarız. Özellikler panelinden yaptığımız bu işlemi kod yazarak da gerçekleştirmemiz mümkündür. Kod editöründe *Bitmap* sınıfına ait *FromFile()* metodunun içerisine *çift tırnak* kullanarak dosya yolunu yazmamız gerekmektedir.



RadioButton kontrolü birden fazla seçenek arasından sadece birini seçebilmemize izin verir.

## RadioButton Kontrolü

RadioButton kontrolü birden fazla seçenek arasından sadece birini seçmeniz gereken bir durum olduğu zaman kullanılmaktadır. Form üzerine beş adet RadioButton eklediğimizde bunlardan biri seçili konumdayken başka birini işaretlediğinizde önceden seçili olan RadioButton kontrolüne ait seçim işleminin kalktığını gözlemleyebilirsiniz. Bu kontrol genellikle anlam yönünden birbirleriyle zıt olan ilişkilerde kullanılmaktadır. Örneğin kullanıcıya ait medeniyet durumunun öğrenilmek istendiği bir formda üç farklı RadioButton kullanabiliriz. Bunlar; evli, bekâr ve dul olarak listelenecektir. Böyle bir durumda kullanıcı bu seçeneklerden sadece birini işaretleyebilir. Mantıksal olarak olayı düşündüğümüzde de bir bireyin her iki durumu veya üç durumun hepsini işaretlemesi doğru olmayacaktır. Ancak burada dikkat etmemiz gereken bir ölçüt bulunmaktadır. Hem medeni halin hem de cinsiyetin aynı form üzerinde sorgulandığını düşünecek olursak formumuza beş adet RadioButton eklememiz gerekecektir. Böyle bir durumda cinsiyet ve medeni

hal farklı sorgular olduğu için ilgili RadioButton kontrollerini gruplamamız gerekmektedir. Aksi takdirde bu kontroller birbirlerini etkileyeceğinden erkek olan birinin bekâr olamayacağı veya kadın olan birinin aynı zamanda evli olamayacağı gibi bir mantıksal çelişki ortaya çıkabilir.

RadioButton kontrollerini gruplamak için araç kutusunda bulunan **GroupBox** kontrolünü kullanmamız gerekmektedir. Araç kutusunda bulunan Groupbox kontrolünü işaretledikten sonra form üzerindeki ilgili RadioButton kontrollerini bu bileşenin içine alarak gruplananların dışarda kalanlardan bağımsız bir şekilde çalıştığını gözlemleyebiliriz. Bu duruma benzer şekilde hazırlanan aşağıdaki örneğimizi inceleyelim.



Örnek

- Form üzerine beş adet Radiobutton ekleyerek text özelliklerini düzenleyelim. Bunlardan iki tanesini cinsiyet başlığı altında bir Groupbox içine alalım. Geriye kalan Radiobuttonları da medeni hal başlığı altında başka bir Groupbox içine alalım. Programı çalıştırdığımız zaman elde edebileceğimiz ekran görüntüsü Şekil 8.8.'de gösterilmektedir.

Şekil 8. 8. Groupbox Kontrolleri ile Radiobutton Kullanımı



RadioButton kontrolüne ait en sık kullanılan özellik Checked özelliğidir.

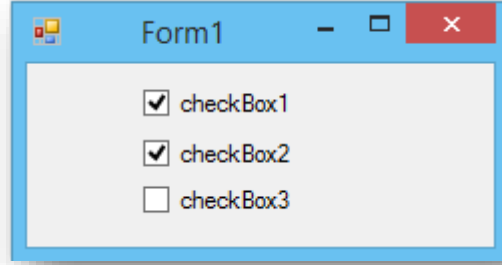
Bu örnekte RadioButton kontrollerini gruplamadığımız zaman cinsiyet ve medeni hal durumunun birbiri üzerindeki etkisi ortaya çıkacaktır. Bu şekilde olabilecek mantıksal bir hatanın önüne geçebilmek için GroupBox kontrolü kullanılmıştır.

Form üzerine yerleştirilen kontroller arasında geçiş işlemi genellikle tab tuşu ile gerçekleştirilmektedir. Ancak bu durum RadioButton kontrolü için geçerli değildir. Formumuzda bulunan RadioButton kontrolleri arasındaki geçiş işlemini ok tuşlarını kullanarak gerçekleştiririz.

RadioButton kontrolüne ait en sık kullanılan özellik *Checked* özelliğidir. Bu özellik *True* değerini alıyorsa kontrolün tıklanmış olduğu, *False* değerini alıyorsa tıklanmamış olduğunu belirtir. Ayrıca uygulamamıza eklediğimiz RadioButtona ait metin bilgisi düğmenin sağ kısmında yer almaktadır. Düğme ile düğme metninin yerlerini, kontrolün özellikler panelindeki *CheckAlign* özelliği ile açılan alandan belirlediğimiz şekilde değiştirebiliriz.

## CheckBox Kontrolü

*CheckBox* kontrolü kullanıcıya belirli seçenekler arasından bir veya daha fazla seçim yapabilmesini sağlamaktadır. RadioButton kontrolünde kullanıcının yaptığı bir seçim diğer kontrolü etkilerken, CheckBox kontrolünde yapılan işlemler birbirinden bağımsız bir şekilde çalışmaktadır. Form üzerine eklenen CheckBox kontrolüne ait ekran görüntüsü Şekil 8.9.'da gösterilmiştir.



Şekil 8.9. Checkbox Kontrolü



CheckBox kontrolü kullanıcıya belirli seçenekler arasından bir veya daha fazla seçim yapabilmesini sağlamaktadır.

Form üzerine eklenen CheckBox kontrolünün seçili olma durumu *Checked* özelliği ile kontrol edilir. Bu özellik *True* değerini aldığı anda kontrol işaretlenmiş, *False* değerini aldığı anda kontrol işaretlenmemiş anlamı ortaya çıkar.

Bazı uygulamalarda CheckBox kontrolünün kullanıcı tarafından değiştirilmesi istenmeyebilir. Bunun için özellikler panelinde bulunan *AutoCheck* seçeneğine *False* değerini atamamız gerekmektedir. Böylelikle çalışma anında kullanıcı kontrolü işaretleyemez veya kontrolün işaretini kaldıramaz.

CheckBox kontrolünün seçili olup olmadığına ait bilgiye *CheckState* özelliğinden ulaşılabilmektedir. Bu özellik üç farklı değer alabilir. Bunlar;

- *Unchecked*: İşaretsiz.
- *Checked*: İşaretli.
- *Indeterminate*: Belirsiz.

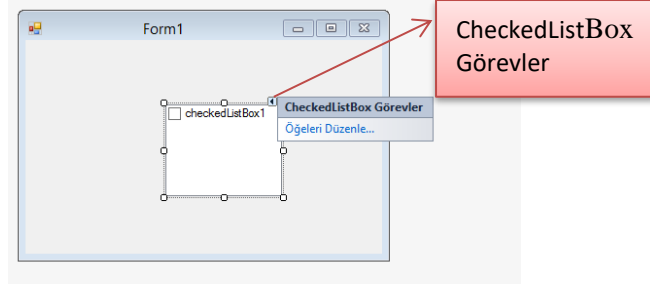
CheckState özelliğinde olduğu gibi çalışma anında da CheckBox kontrolünün kullanıcı tarafından üç farklı değerle işaretleme yapmasına izin verebiliriz. Bunun için özellikler panelinde bulunan *ThreeState* özelliğini *True* olarak ayarlamamız yeterli olacaktır.

CheckBox kontrolünü, uygulamamızın tasarımına ve amacına bağlı olarak farklı görünüm stilleri ile sunabiliriz. *FlatStyle* özelliği ile kontrolün; *Flat*, *Popup*, *Standard* ve *System* olmak üzere dört farklı görünümünden birini tercih edebiliriz.

Bunun haricinde CheckBox'ın bir buton gibi görünmesini istiyorsak *Appearance* özelliğini kullanabiliriz.

## CheckedListBox Kontrolü

CheckedListBox kontrolü, CheckBox elemanlarını ListBox içerisinde görüntülenmesini sağlamaktadır. Yani her bir öğenin sol kısmında onay kutusunun bulunduğu bir listeleme şeklidir. CheckedListBox kontrolüne eleman eklemek için, Şekil 8.10.'da gösterildiği gibi kontrolün sağ üst kısmında bulunan oka tıkladıktan sonra *Öğeleri Düzenle...* seçeneğini işaretlememiz gerekmektedir. Bu işlemin ardından karşımıza gelecek olan *Dize Koleksiyonu Düzenleyicisi* penceresinde her bir öğeyi satırlar halinde girip tamam butonuna tıklamamız gerekmektedir. Kontrole öğe ekleme işini yukarda bahsedildiği üzere tasarım ekranından yapabileceğimiz gibi özellikler panelinden *Items* seçeneğiyle de yapabilmemiz mümkündür.



Şekil 8.10. CheckedListbox Kontrolü



CheckedListBox kontrolü dizi mantığı ile çalışmaktadır. Her bir elemana ait index numarası vardır.

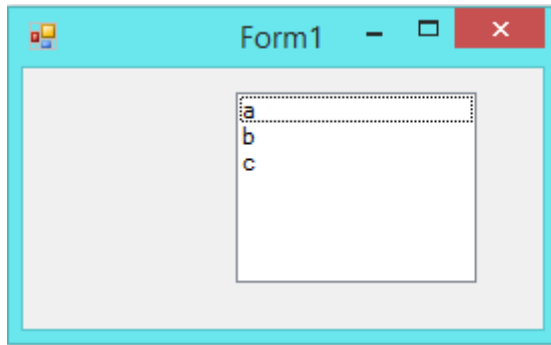
CheckedListBox kontrolünde bulunan öğelerle işlem yapabilir, birini veya daha fazlasını seçebiliriz. Kontrol içerisinde bulunan öğelerle çalışırken *Items* özelliği kullanılmaktadır. Bütün öğeler yerine seçili olan öğe ile çalışmak içinde *CheckedItems* özelliğini kullanmamız gerekmektedir.

CheckedListBox kontrolü dizi mantığı ile çalışmaktadır. Her bir elemana ait *index* numarası vardır. Elemanların index numarası sıfırdan başlayarak ilerler. *SelectedIndex* özelliği kullanılarak CheckedListBox üzerinde seçim işlemi gerçekleştirilen elemanın sıra numarasını görüntüleyebiliriz. Bununla birlikte sıra numarası bilinen bir elemanın değerini görüntülemek için de *CheckedListBoxAdı.Items[sıra numarası];* kodunu yazmamız gerekmektedir. Bu kod satırını MessageBox gibi bir yöntemle görüntülerken, eleman değerini metin türüne dönüştürmeyi ihmal etmemeliyiz.

Çalışma anında kontrol üzerinde bulunan öğelerin seçim ve onay işlemlerini birbirleriyle karıştırmamak gerekir. Bir öğeye ait onay kutusunu işaretli hale getirebilmek için, öğe üzerine gelerek farenin sol tuşu ile tıklayarak seçim yapıldıktan sonra işaretlemek için tekrar tıklamamız gerekmektedir. Yukarda bahsettiğimiz *SelectedIndex* özelliği kontrol üzerinde en son seçim yapılan öğeye ait sıra numarasını verir. Eğer biz CheckedListBox kontrolünde bulunan öğelerden kaç tanesinin onaylı olduğunu öğrenmek istiyorsak, *CheckedListBoxAdı.CheckedItems.Count()* kodunu kullanmamız gerekmektedir.

## ListBox Kontrolü

ListBox kontrolü içerisindeki elemanları liste şeklinde görüntülemeyi sağlamaktadır. Bu kontrolde bulunan elemanlara *index* numaraları ile ulaşılabilir. Kontrole girilen ilk elemanın index numarası sıfır değerini alır. ListBox içerisine değer girebilmek için özellikler panelinde bulunan Items seçeneği kullanılabilir. ListBox, ComboBox ve CheckedListBox kontrollerine ait özellik, metod ve olaylar neredeyse aynıdır. Örneğin CheckedListBox kontrolünde anlatılan SelectedIndex özelliğini ListBox kontrolünde de aynı şekilde uygulayabilirsiniz. Bu kontroller incelenirken anlatılan özelliklere tekrar değinilmeyecek olup farklı olanlardan bahsedilecektir. ListBox kontrolünün form üzerindeki ekran görüntüsü Şekil 8.11.'de verilmiştir.



Şekil 8.11. Listbox Kontrolü



SelectionMode özelliği, listbox kontrolündeki öğelerin seçiminin farklı şekillerde yapılabilmesini ayarlayabileceğimiz özelliktir.

Çalışma anında kontrol üzerindeki öğelerin seçiminin farklı şekillerde yapılabilmesini sağlayabilirsiniz. Bu işlemi gerçekleştirmek için özellikler panelinde bulunan *SelectionMode* seçeneğini kullanmamız gerekmektedir. Bu özellik *None*, *One*, *MultiSimple* ve *MultiExtended* olmak üzere dört farklı seçenek barındırmaktadır. Bunlardan ilk olan None seçeneği işaretlendiğinde, kullanıcı ListBox kontrolü üzerinde bulunan hiçbir öğeyi seçemez. One seçeneği aktif hale getirildiği zaman, ListBox kontrolü üzerinde sadece bir öğe seçilebilir hale gelecektir. Farklı bir seçim yapıldığı takdirde bir önceki seçim ortadan kalkacaktır. MultiSimple seçeneği aktifleştirildiğinde, kullanıcının yapmış olduğu farklı seçimlerin hepsi işaretli konumda olur. Seçilen öğenin seçimini kaldırmak için o öğe üzerine tekrar tıklanması gerekmektedir. Son olarak kontrole ait SelectionMode seçeneği MultiExtended olarak ayarlandığında da, One seçeneğine benzer şekilde işlem yapılmaktadır. Ancak One seçeneğinde farklı olarak seçim işlemi yapılırken Ctrl ve Shift tuşlarının kullanımına izin verilmektedir. Ctrl tuşu basılı tutularak yapılan seçimlerde birden fazla işaretleme yapılabilir. Shift tuşu basılı tutularak yapılan seçimlerde ise, ilk seçilen öğe ile ikinci seçilen öğe arasında kalan öğelerinde seçimi sağlanmaktadır. SelectionMode özelliği CheckedListBox kontrolünde de aynı şekilde kullanılmaktadır.

Uygulamalarımızda bazen yaptığımız işlemlerin temizlenmesi gerekebilir. Böyle bir durumda *Clear* metodunu kullanabiliriz. Çalışma anında ListBox üzerinde bulunan elemanların silinmesi için, ListBoxAdı.Items.Clear(); kodunu herhangi bir butonun içine veya program akışındaki istediğimiz herhangi bir alana yazmamız gerekmektedir.

## ComboBox Kontrolü

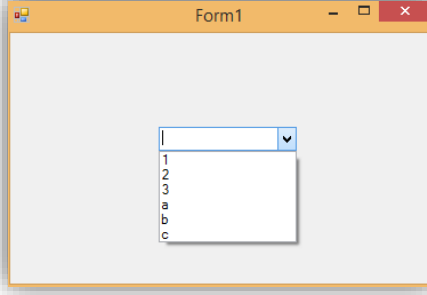
ComboBox kontrolü, içinde bulunan elemanları açılabilen liste halinde görüntülemeyi sağlamaktadır. Bu kontrol içerisindeki elemanlara da ListBox ve CheckedListBox kontrollerinde olduğu gibi index numaraları ile ulaşılabilir.

ComboBox içinde bulunan elemanları görüntülemek için, çalışma anında ComboBox üzerine tıklamak gerekmektedir. Bu tıklama işlemi yapılmadan ComboBox içinde herhangi bir metin görüntülemek istiyorsak, özellikler panelinde bulunan **Text** alanını doldurmamız gerekmektedir.



ComboBox kontrolü, içinde bulunan elemanları açılabilen liste halinde görüntülemeyi sağlamaktadır.

Uygulama çalışırken kontrol içinde bulunan öğelerin sıralı bir şekilde görüntülenmesini sağlayabiliriz. Özellikler panelinde bulunan **Sorted** seçeneğini **True** olarak ayarlasak bu işlemimiz gerçekleşmiş olacaktır. Kontrol içine rasgele girilen sayısal ve sözel ifadeler, uygulama çalıştırıldığında Şekil 8.12.'de gösterildiği gibi önce sayıları küçükten büyüğe, sonra metinleri A'dan Z'ye doğru sıralı bir şekilde görüntüleyecektir.



Şekil 8.12. Controlbox Kontrolü, Sorted = True

**Clear** metodu CheckedListBox ve ListBox kontrollerinde olduğu gibi ComboBox kontrolünde de bütün öğelerin silinmesine yardımcı olmaktadır. Tüm öğeler yerine tek bir öğeyi silmek için iki farklı metot bulunmaktadır. Bunlardan biri **Remove** metodudur. Bu metoda göre silme işlemi yapılırken silinecek öğenin **index** numarası metot içinde belirtilmelidir. Diğer olan **RemoveAt** metodunun içine **seçim** (SelectedIndex) yapılarak silinmesi gerektiğini belirterek kullanabiliriz.

## KONTROLLERİN ARA YÜZ TASARIMI

Bir uygulamanın işlevsel olması kadar göze hitap etmesi de büyük önem taşımaktadır. Formların tasarımı gerçekleştirilirken; üzerine yerleştirilen kontrollerin birbirleriyle olan bağlantısı, şekil-zemin ilişkisi, kontrollere ait metinlerin okunabilirliği gibi birçok özellik dikkate alınarak yapılmalıdır. Bu özelliklerin başlı başına bir konu olarak ele alınması gerekse de kontroller üzerinde toplu olarak işlemlerin nasıl yapılacağını bilmek faydalı olacaktır. Kontrolleri grup halinde taşıma, hizalama veya kontrollerin benzer özelliklerini toplu olarak değiştirme ihtiyacı olabilir.

Kontroller üzerinde toplu olarak işlemlerimizi gerçekleştirmeden önce birden fazla kontrolün seçimini yapabilmemiz gerekmektedir. Bu işlem için farklı

yöntemler bulunmaktadır. Kişisel tercihe ve kullanım kolaylığına bağlı olarak istediğiniz bir yöntemi kullanabilirsiniz. Form üzerinde birbirine yakın olan kontroller *fare yardımıyla* çerçeve içine alınarak seçim işlemi gerçekleştirilebilir. *Ctrl* veya *Shift* tuşlarından herhangi birine basılı tutulup seçilmek istenen kontroller tek tek tıklanarak seçim işlemi gerçekleştirilebilir. Aynı zamanda bu tuşlar toplu seçimlerden istenmeyen bir kontrolü çıkarmak için de kullanılabilir. Form üzerinde bulunan kontrollerin tamamını seçmek istiyorsak *Ctrl + A* tuş kombinasyonunu kullanabiliriz. Bu yöntemleri çoğaltmak mümkündür. Tercih tamamen programcının alışkanlıklarına ve bilgi birikimine bağlıdır. Toplu olarak yapılan seçim işlemi iptal etmek için form üzerinde boş bir alana tıklamanız yeterli olacaktır.

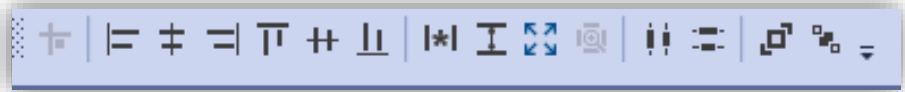


Düzen araç çubuğunu, Görünüm menüsü altında bulunan Araç Çubukları açılır listesindeki Düzen seçeneğini işaretleyerek ekrana getirebiliriz.

Seçim işlemi gerçekleştirilen kontrolleri grup halinde hareket ettirmek için herhangi bir kontrolün üzerine gelerek sürükleyip bırak yöntemi kullanılabilir. Daha kısa hareketler gerçekleştirmek için klavye üzerinde bulunan ok tuşları kontrollerin sağa, sola, yukarı ve aşağı olmak üzere ilerlemesini sağlayacaktır. Kontrolleri grup halinde hareket ettirebileceğiniz gibi özellikler panelinde bulunan bir takım seçenekler üzerinde de değişiklik yapabilirsiniz. Örneğin seçilen kontrollerin arkaplan rengi (BackColor) veya yazı tipi (font) özellikleri toplu olarak değiştirilebilir. Ancak özellikler panelinde bulunan her seçenek bu değişiklik için uygun olmayabilir. Her kontrol için özel olan *name* özelliği gibi bir seçenek toplu olarak değişikliğe uygun değildir. Burada nesnelerin ortak özellikleri kullanılmalıdır.

Birden fazla seçimi yapılmış olan kontrollerden herhangi biri üzerine fare ile gelerek kontrolün yanında bulunan tutamaçlar yardımıyla boyutu değiştirildiğinde, yapılan bu değişiklik diğer kontroller üzerinde de aynı oranda gerçekleşecektir. Böyle bir değişiklik yapılırken bir kontrolün başka bir kontrolün üstünde veya altında kalmamasına dikkat edilmelidir. Eğer yapılan değişikliklerin uygun olmadığını düşünürseniz *Ctrl + Z* tuş kombinasyonunu kullanarak yaptığınız işlemi hızlı bir şekilde geri alabilirsiniz.

Form üzerine yerleştirdiğimiz kontrolleri birbirlerine göre hizalarken ufak farkları gözden kaçırabiliriz. Benzer şekilde kontrollerin aynı boyutta olmasını istiyorsak fark edemeyeceğimiz düzeyde eşitsizlikler olabilir. Bu yüzden Şekil 8.13.'de gösterilen *Düzen Araç Çubuğunu* kullanmak işimizi kolaylaştıracaktır. Düzen araç çubuğu ekranda yoksa *Görünüm* menüsü altında bulunan *Araç Çubukları* açılır listesindeki *Düzen* seçeneğini işaretleyerek ekrana getirebiliriz. Grup halindeki kontrolleri bu araç çubuğunda bulunan butonlar yardımıyla sağa, sola, yukarı veya aşağı hizalayabiliriz. Aynı zamanda kontrollerin boyutlarının da aynı yükseklik veya genişliğe gelmesini sağlayabiliriz.



Şekil 8.13. Düzen Araç Çubuğu



Düzen araç çubuğu ekranda görünmesine rağmen üzerinde bulunan butonlar pasif konumdaysa ya bir kontrol seçilmiş ya da hiçbir kontrol nesnesi seçilmemiş demektir. Yani bu araç çubuğunu kullanabilmek için birden fazla kontrolün seçilmesi gerekmektedir. Ayrıca düzen araç çubuğunu kullanırken dikkat etmemiz gereken bir ölçüt bulunmaktadır. Hizalama veya boyutlandırma işleminin referans noktasını doğru bir şekilde seçmeliyiz. Yani toplu seçim yaptığımız kontrollerden seçim işlemine dahil ettiğimiz ilk kontrol diğer kontrollerin referans noktası olacaktır. Bütün kontroller hizalanırken veya boyutlandırılırken ilk kontrole göre pozisyonlarında değişikliğe uğrayacaktır.



## Özet

- Windows form projelerinde gerek programcının gerek kullanıcının ihtiyaç duyduğu bilgilendirme, etkileşim, bilgi girişi gibi birçok işlem bulunmaktadır. Bu işlemleri gerçekleştirmek üzere Visual Studio .Net içerisinde kullanıma hazır bir şekilde geliştirilen pek çok kontrol mevcuttur.
- Bazı durumlarda form üzerinde bulunan nesnelerin ne işe yaradıklarını ifade etmek veya belirli konularda kullanıcıya bilgi vermek için etiketleme işleminin yapılmasına ihtiyaç duyulmaktadır. Böyle durumlarda label kontrolünün kullanımı ihtiyaca cevap verebilmektedir. Genellikle içeriği sabit kalıp üzerinde değişiklik yapılmayacak bilgilerin gösteriminde kullanılmaktadır.
- Projelerimizde bazı durumlar için web sitesine bağlantı yapılması gerekebilir. Böyle durumlarda LinkLabel kontrolü kullanılmaktadır. Form üzerinde görüntülenmesi istenilen bağlantı metni, linklabel kontrolünün Text özelliğine yazılmalıdır.
- TextBox kontrolü, programcı tarafından görüntülenmesi istenen bilgilerin yer alabileceği ve kullanıcı tarafından bu bilgilerin değiştirilerek bilgi girişi yapabileceği metin kutularıdır. Özellikler panelinde bulunan Text özelliği sayesinde 255 karakteri aşmamak üzere görüntülenecek metin girişi yapılabilir.
- TextBox kontrolünün özellikleri
- Textbox kontrolünün içerisindeki metnin kullanıcı tarafından değiştirilmesini istemediğimiz durumlar olabilir ya da projemizde bulunan şartlara göre bilgi girişinin açık veya kapalı olmasını sağlayabiliriz. Bu durumda kontrolün ReadOnly özelliğini ayarlamamız gerekmektedir.
- Her bir karakterin özel bir simge ile gösterilmesi için PasswordChar özelliği kullanılır.
- CharacterCasing özelliği ile TextBox içine yazılan yazıların büyük ya da küçük harflerle yazılması mümkün olmaktadır.
- TextBox içerisine girilen metinlerin yazı tipini ayarlamak için özellikler panelinde bulunan Font seçeneğini kullanabiliriz.
- TextBox kontrolünün çok satırlı olmasını Multiline özelliği ile ayarlayabiliriz.
- Lines özelliği sayesinde Textbox kontrolü bir dizi değişkeni gibi işlev görebilmektedir.
- Windows form uygulamalarında kullanıcının veri girişi için belirli sınırlamalar getirmek istenebilir. Böyle bir durumda TextBox kontrolü için girebileceğimiz azami karakter sayısı MaxLengt özelliği ile ayarlanabilmektedir.
- TextBox kontrolünün metodları
- TextBox kontrolü kes, kopyala, yapıştır vb. pano işlemlerini metotlar yardımıyla gerçekleştirebilmektedir. Sık kullanılan metotlar: SelectAll, cUt, Copy, Paste, Undo, Clear.
- TextBox kontrolünün olayları
- Çalışma anında TextBox kontrolü üzerinde yapılan değişiklikleri izleme ihtiyacımız olabilir. Bu durumda TextBox kontrolüne ait olaylardan istifade etmemiz gerekmektedir. TextBox kontrolüne ait birçok olay mevcuttur. Bu olaylardan sık kullanılanlar: TextChange, Enter, Leave, Validated.



## Özet (devamı)

- Windows form uygulamalarının hemen hemen tamamında yer alan Button kontrolü herhangi bir işlemi veya komutu çalıştırmak amacıyla kullanılmaktadır.
- RadioButton kontrolü birden fazla seçenek arasından sadece birini seçmeniz gereken bir durum olduğu zaman kullanılmaktadır. Form üzerine beş adet RadioButton eklediğimizde bunlardan biri seçili konumdayken başka birini işaretlediğinizde önceden seçili olan RadioButton kontrolüne ait seçim işleminin kalktığını gözlemleyebilirsiniz.
- CheckBox kontrolü kullanıcıya belirli seçenekler arasından bir veya daha fazla seçim yapabilmesini sağlamaktadır. RadioButton kontrolünde kullanıcının yaptığı bir seçim diğer kontrolü etkilerken, CheckBox kontrolünde yapılan işlemler birbirinden bağımsız bir şekilde çalışmaktadır.
- CheckedListBox kontrolü, CheckBox elemanlarını ListBox içerisinde görüntülenmesini sağlamaktadır. Yani her bir öğenin sol kısmında onay kutusunun bulunduğu bir listeleme şeklidir.
- ListBox kontrolü içerisindeki elemanları liste şeklinde görüntülemeyi sağlamaktadır. Bu kontrolde bulunan elemanlara index numaraları ile ulaşılabilmektedir. Kontrole girilen ilk elemanın index numarası sıfır değerini alır. ListBox içerisine değer girebilmek için özellikler panelinde bulunan Items seçeneği kullanılabilir.
- ComboBox kontrolü, içinde bulunan elemanları açılabilen liste halinde görüntülemeyi sağlamaktadır. Bu kontrol içerisindeki elemanlara da ListBox ve CheckedListBox kontrollerinde olduğu gibi index numaraları ile ulaşılabilmektedir.
- Form üzerine yerleştirdiğimiz kontrolleri birbirlerine göre hizalarken ufak farkları gözden kaçırabiliriz. Benzer şekilde kontrollerin aynı boyutta olmasını istiyorsak fark edemeyeceğimiz düzeyde eşitsizlikler olabilir. Bu yüzden Düzen Araç Çubuğunu kullanmak işimizi kolaylaştıracaktır. Düzen araç çubuğu ekranda yoksa Görünüm menüsü altında bulunan Araç Çubukları açılır listesindeki Düzen seçeneğini işaretleyerek ekrana getirebiliriz. Grup halindeki kontrolleri bu araç çubuğunda bulunan butonlar yardımıyla sağa, sola, yukarı veya aşağı hizalayabiliriz. Aynı zamanda kontrollerin boyutlarının da aynı yükseklik veya genişliğe gelmesini sağlayabiliriz.

## DEĞERLENDİRME SORULARI

1. Aşağıdakilerden hangisi araç kutusunu ekranda göstermek için kullanılacak olan tuş kombinasyonudur?
  - a) Ctrl+Shift+A
  - b) Ctrl+Shift+X
  - c) Ctrl+Alt+X
  - d) Ctrl+X
  - e) Ctrl+Alt+A
2. Aşağıdakilerden hangisi genellikle içeriği sabit olan ve üzerinde değişiklik yapılmayacak bilgilerin gösteriminde kullanılan kontroldür?
  - a) Label
  - b) Listbox
  - c) ComboBox
  - d) RadioButton
  - e) CheckBox
3. Kod editöründe TextBox kontrolünün Text özelliğine değer aktarmak için hangi işaret kullanılmalıdır?
  - a) \*\*
  - b) //
  - c) <<
  - d) ""
  - e) ++
4. Aşağıdakilerden hangi TextBox kontrolünün çok satırlı olmasını sağlayan özelliktir?
  - a) ReadOnly
  - b) PasswordChar
  - c) CharacterCasing
  - d) Font
  - e) MultiLine
5. Aşağıdaki metotlardan hangisi TextBox kontrolünün içindeki yazıları siler?
  - a) SelectAll
  - b) Clear
  - c) Undo
  - d) Paste
  - e) Copy

6. Aşağıdakilerden hangisi birden fazla seçenek arasından sadece birini seçmeniz gereken durumlarda kullanılan kontroldür?
  - a) Label
  - b) TextBox
  - c) ListBox
  - d) RadioButton
  - e) Button
7. Bir veya daha fazla alternatifi seçebilmemizi sağlayan kontrol hangisidir?
  - a) Button
  - b) ListBox
  - c) CheckBox
  - d) Label
  - e) LinkLabel
8. CheckedListBox kontrolünde seçili olan öğe ile çalışmak için, kontrolün aşağıda verilen özelliklerinden hangisini kullanmamız gerekir?
  - a) CheckedItems
  - b) SelectedIndex
  - c) CheckedListBox
  - d) MessageBox
  - e) MultiExtended
9. Aşağıdakilerden hangisi, içinde bulunan elemanları açılabilen liste halinde görüntülemeyi sağlayan kontroldür?
  - a) ListBox
  - b) Label
  - c) TextBox
  - d) Button
  - e) ComboBox
10. Düzen araç çubuğunu ekranda görüntülemek için kullanabileceğimiz menü yolu aşağıdakilerden hangisidir?
  - a) Düzen -> Git -> Sembol Git...
  - b) Proje -> Sınıf ekle
  - c) Görünüm -> Araç çubukları -> Düzen
  - d) Görünüm -> Terminal
  - e) Git -> Kopyala

**Cevap Anahtarı**

1.c, 2.a, 3.d, 4.e, 5.b, 6.d, 7.c, 8.a, 9.e, 10.c

## YARARLANILAN KAYNAKLAR

- Aktaş, V. (2010). *Visual studio 2010 ile her yönüyle C# 4.0* (2. Baskı). İstanbul: Kodlab Yayın.
- Aktaş, V. (2020). *Visual studio 2019* (1.Baskı). İstanbul: 01 Yayınları.
- Atasever, V. (2017). *C# 7 Uygulamalarla C# programlama dilini keşfedin* (2. Baskı). Kocaeli: Level Kitap.
- Demirli, N. & İnan, Y. (2005). *Visual C# .Net* (3. Baskı). Ankara: Palme Yayıncılık.
- Sharp, J. (2011). *Adım adım Microsoft visual C# 2010*. (Çev. T. Buldu). Ankara: Arkadaş Yayınevi.
- Uzunköprü, S. (2017). *Projeler İle C# 7.0 ve SQL Server 2016* (7. Baskı). İstanbul: Kodlab Yayın.
- Yanık, M. (2008). *Visual studio 2008 ile Microsoft visual C# 3.0 for .Net framework 3.5* (1. Baskı). Ankara: Seçkin Yayıncılık.
- Yücedağ, M. (2020). *C# Eğitim kitabı* (3. Baskı). İstanbul: Dikeyeksen.