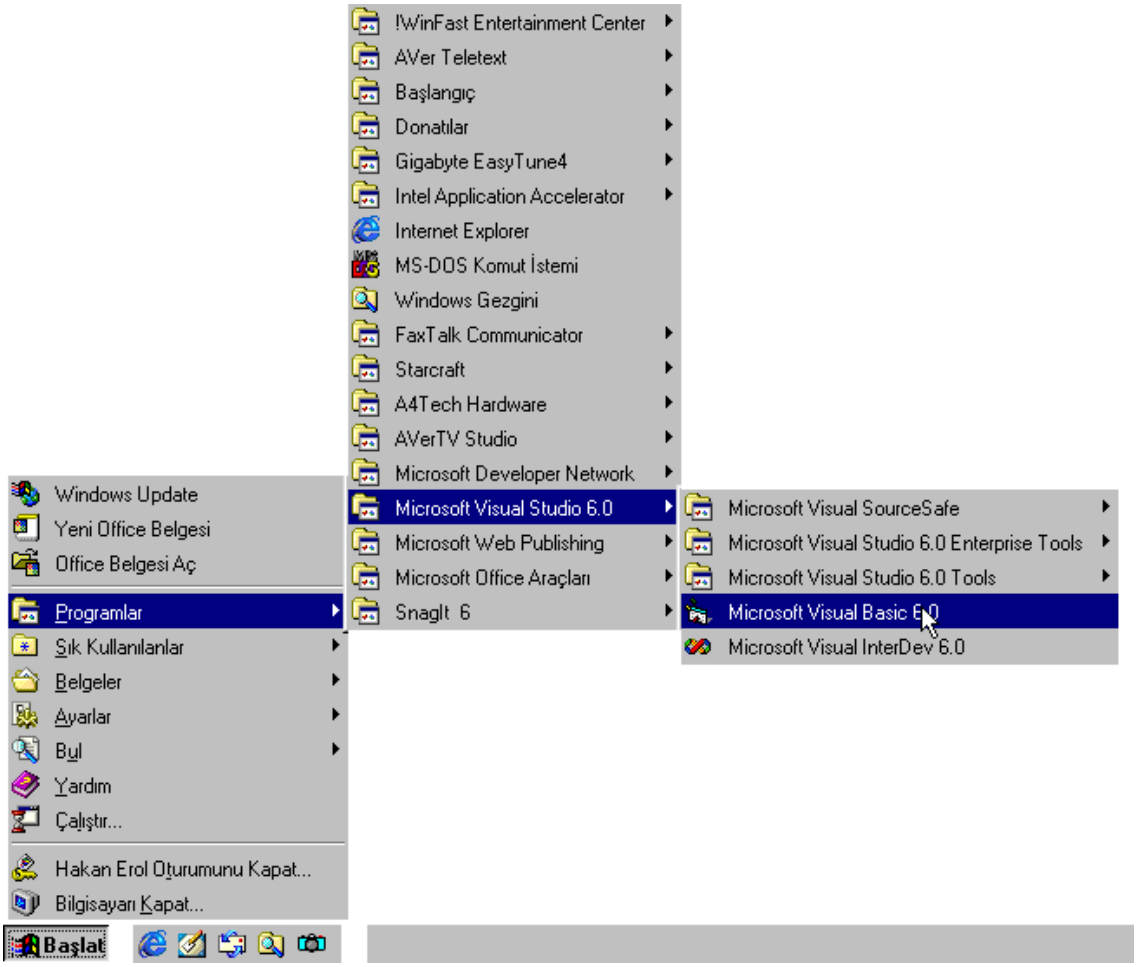
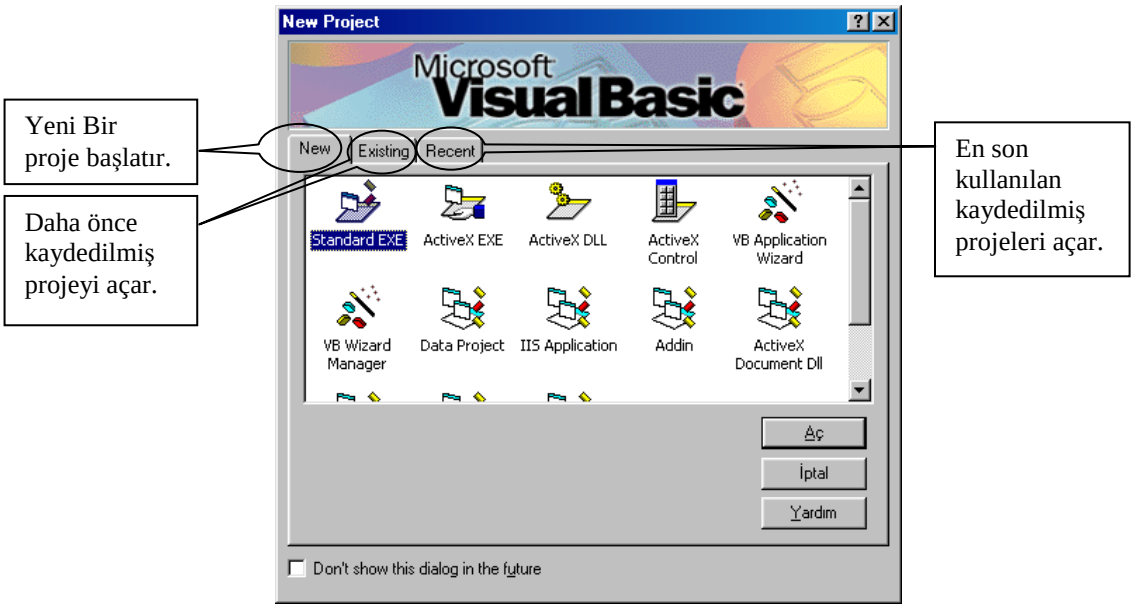
VISUAL BASIC 6.0

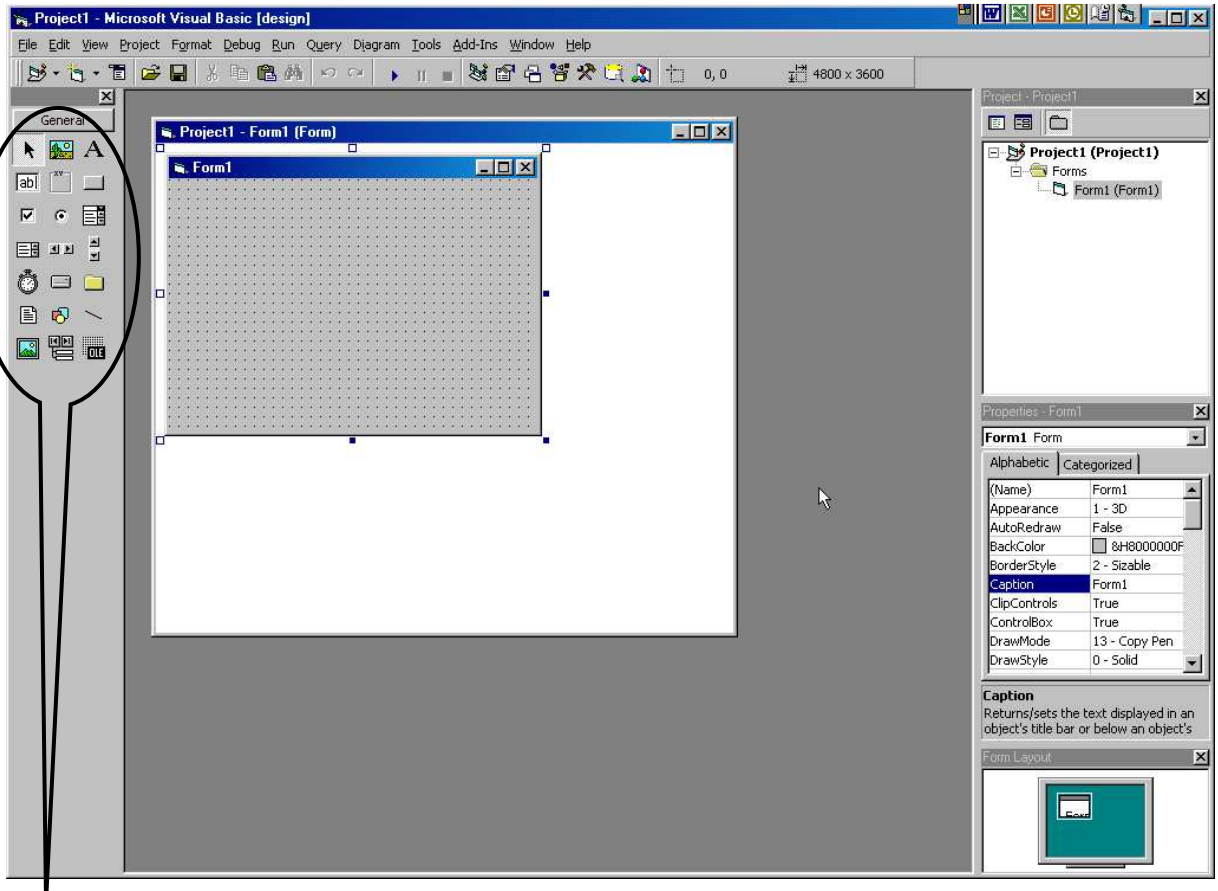
Visual Basic 6.0'la çalışmaya başlama için;



Microsoft Visual Basic 6.0 sekmesine bir kere tıklanarak program açılır. Program ilk açılışta

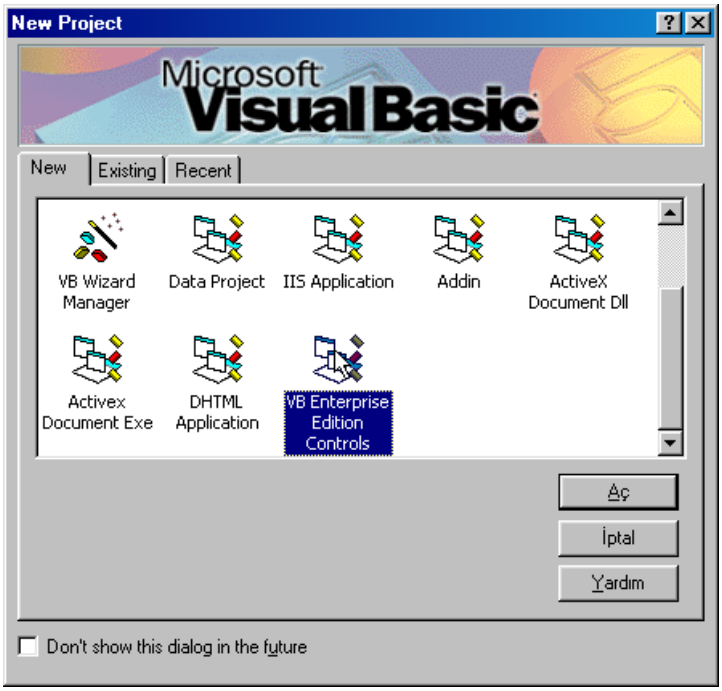


penceresi karşımıza çıkar. Bu pencere sayesinde ne tür bir program hazırlamak istiyorsak onun için gerekli bileşenlerin (kontrollerin) bulunduğu yeni bir proje başlatılır. Ayrıca bu pencere sayesinde daha önce çalışmaya başladığımız ve kaydettiğimiz projelere de ulaşabiliriz. Yeni bir proje başlatmak için **STANDART EXE** seçeneği seçili iken Aç butonuna bir kere tıklanırsa

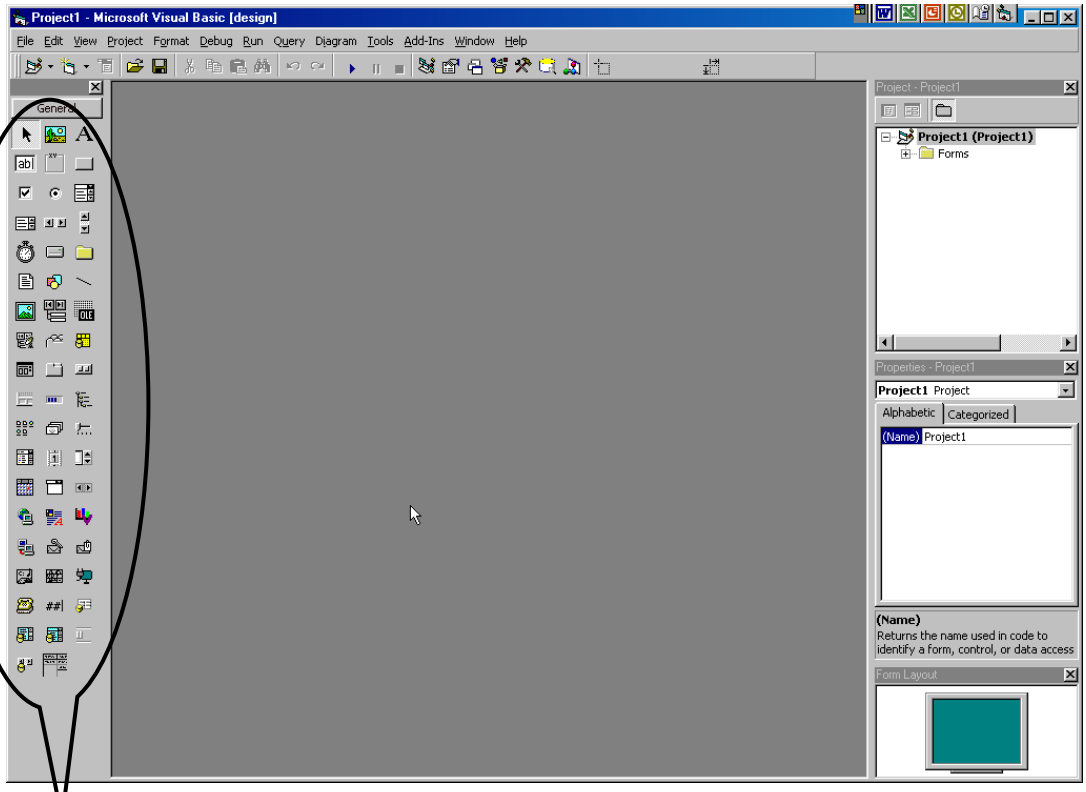


STANDART EXE seçeneği ile açılan V.B. 6.0'da bulunan bileşenler.

penceresi açılır. Açılışta daha fazla bileşene ihtiyaç duyuluyorsa;



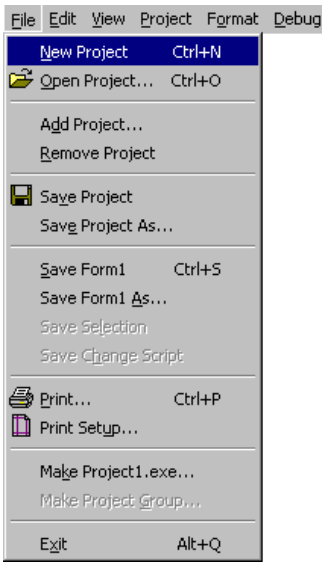
yine aynı açılış penceresinden **VB Enterprise Edition Controls** seçeneği işaretlenerek açılabilir. Böylece daha fazla bileşen kullanılabilir.



VB Enterprise Edition Controls seçeneği seçilerek açılan projede bulunan bileşenler.

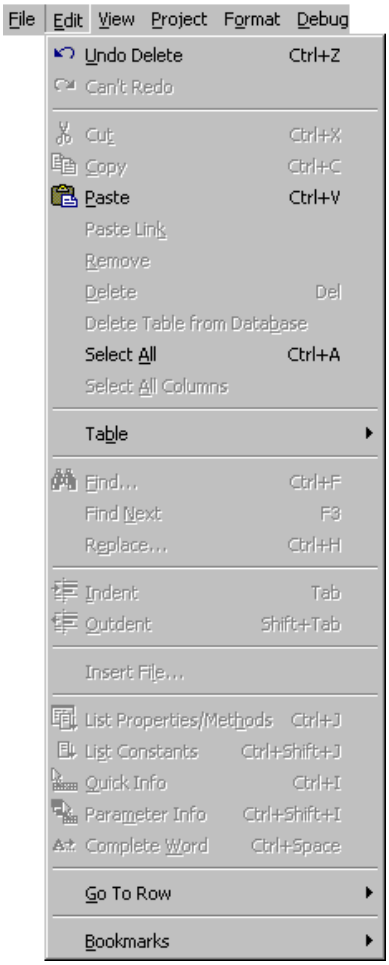
VB de diğer Windows altında çalışan programlarında olduğu gibi çeşitli menülerin bulunduğu bir programdır.

FİLE MENÜSÜ



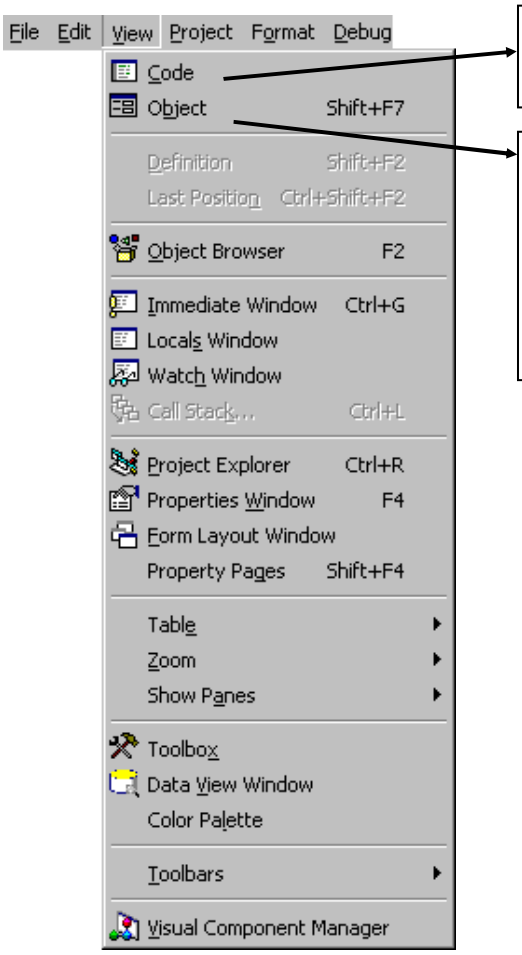
Diğer Microsoft programlarında olduğu gibi dosya açma, dosya kaydetme farklı isimde kaydetme, yazdırma seçeneklerinin bulunduğu menüdür. VB'de yaptığımız bir çalışmayı kaydederken program ilk olarak formları kaydetmemizi isteyecektir. Formlar **frm** uzantısı ile kaydedildikten sonra projeyi kaydetmemizi isteyecektir. Projeler de **vbp** uzantısı ile kaydedilir.

EDIT MENÜSÜ



Diğer Microsoft programlarında olduğu gibi kes, kopyala, yapıştır işlemlerinin yapıldığı menüdür. Ayrıca kod penceresinde herhangi bir metin parçasının bulunup değiştirilmesini sağlayan seçenekler vadır.

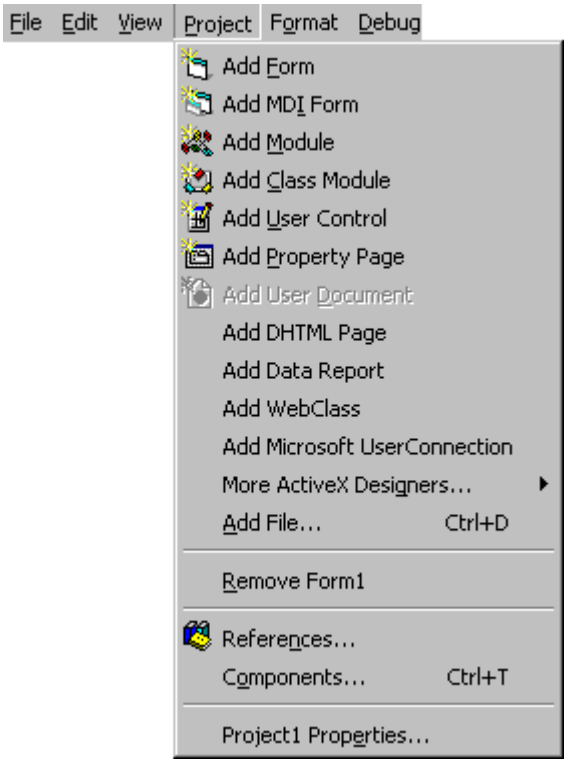
VIEW MENÜSÜ



Program kodunun yazıldığı **kod penceresini** öne getirir.

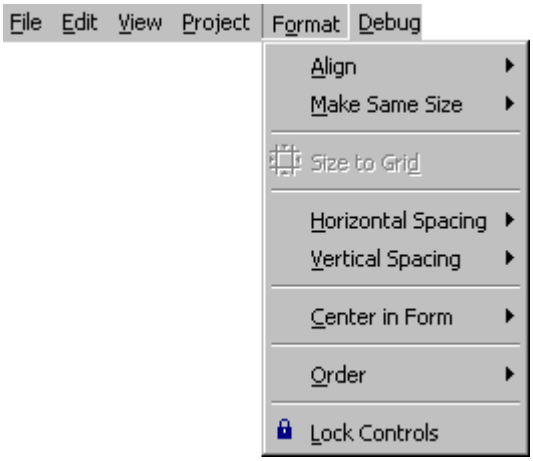
Program formunun bulunduğu pencereyi öne getirir. Eğer birden fazla form bulunuyorsa **Project Explorer**'da seçili olan formun bulunduğu pencereyi öne getirir.

PROJECT MENÜSÜ



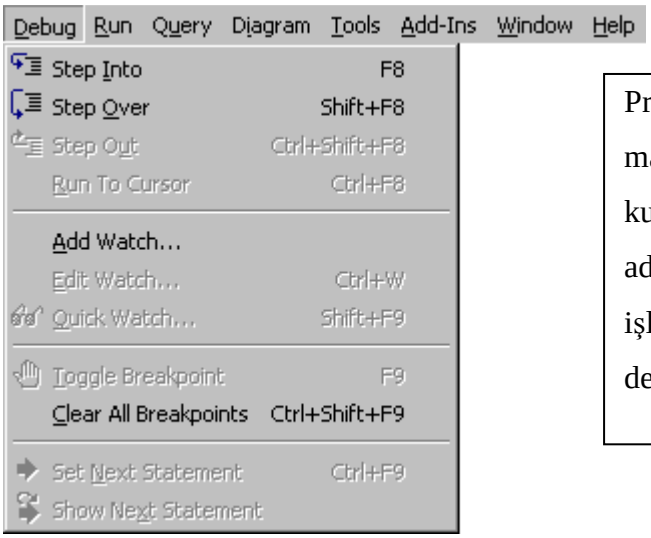
Projeye çeşitli yeni formlar eklenmek istendiğinde bu menü kullanılır. Ayrıca Toolbox çubuğuna yeni (components) bileşenler eklemek içinde kullanılır.

FORMAT MENÜSÜ



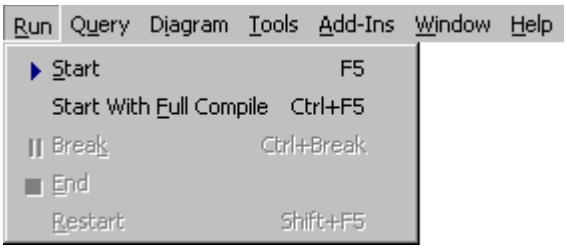
Formun tasarlanması sırasında, yerleştirilen bileşenlerin, birbirlerine göre ve/veya forma göre konumlarını düzenleyen seçenekler vardır.

DEBUG MENÜSÜ



Program çalıştırılmaya başlandıktan sonra mantıksal hataların denetlenip ayıklanması için kullanılan seçenekler mevcuttur. Program adım adım çalıştırılarak hangi aşamalarda hangi işlemlerin yapıldığı ve değişkenlerin hangi değerleri aldığı belirlenebilir.

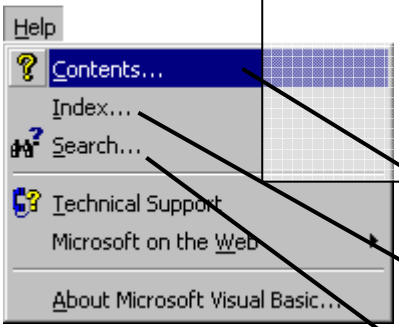
RUN MENÜSÜ



Program çalışmaya hazır hale getirildikten sonra çalıştırmak için kullanılan menüdür.

HELP MENÜSÜ

Program geliştirirken herhangi bir konuda yardıma ihtiyaç duyulduğunda kullanılan menüdür. Diğer programların aksine bilhassa programlama dillerinde yardımın büyük önemi vardır ve oldukça sık kullanılır.



Tüm Visual Studio 6.0 içeriğini klasörler halinde gösterir.

Tüm Visual Studio 6.0 içindeki komutları alfabetik sıra ile gösterir.

Yardım istenilen konu ile ilgili kelimeler girildiğinde sonuçlarını gösterir.

VB'de diğer Microsoft programlarında olduğu gibi çeşitli araç çubuklarının bulunduğu bir programdır. Standart araç çubuğunda diğer programlarda olduğu gibi dosyalama işlemlerinin yanı sıra VB için özel bazı butonlar da vardır.

STANDART ARAÇ ÇUBUĞU

: Çalışmakta olduğunuz projeye ilave **yeni** bir proje açar.

: Çalışmakta olduğunuz projeye **yeni** bir form ekler.

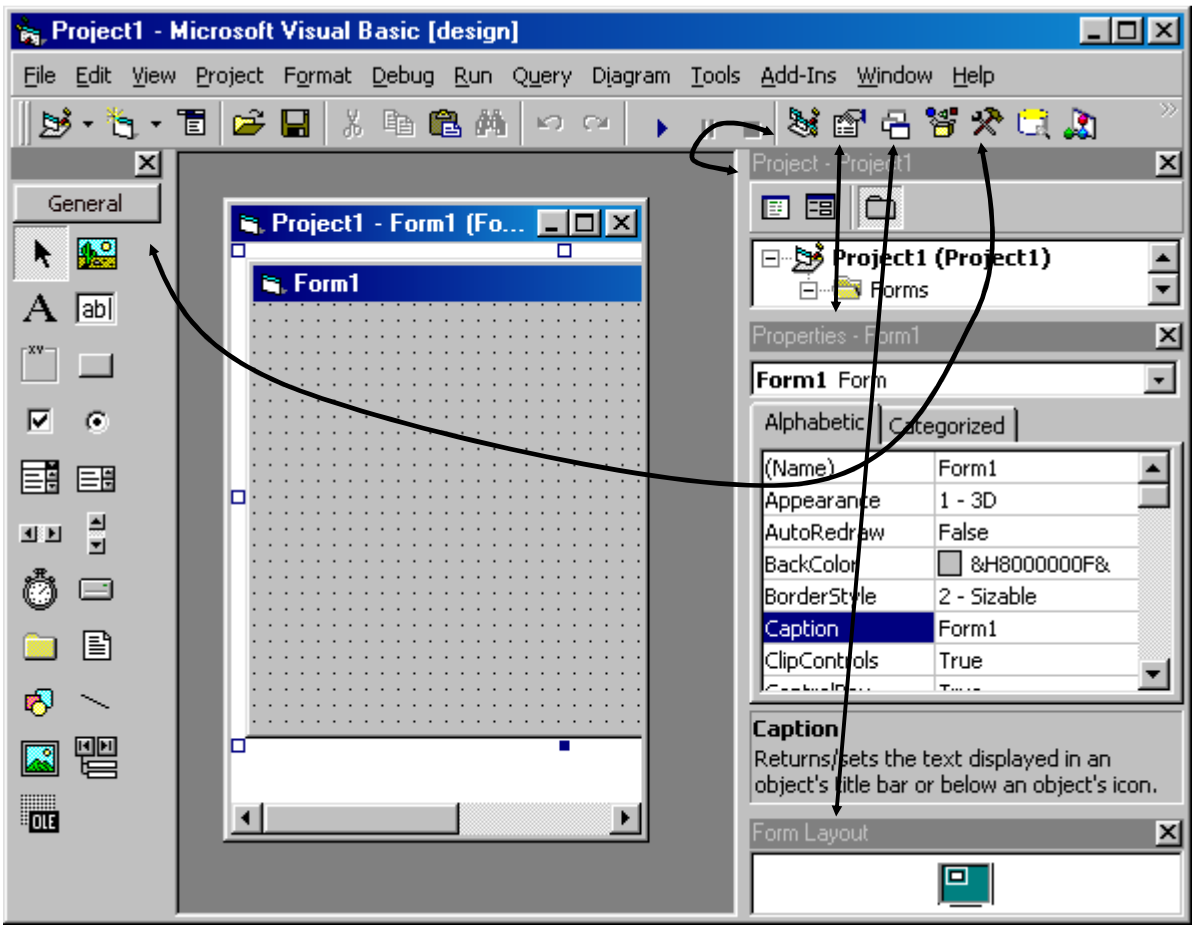
: Eğer kapanmışsa **Project Explorer** penceresini açar.

: Eğer kapanmışsa **Properties Window** (Özellikler Penceresini) açar.

: Eğer kapanmışsa **Form Layout Window** (Form Yeri Penceresini) açar.

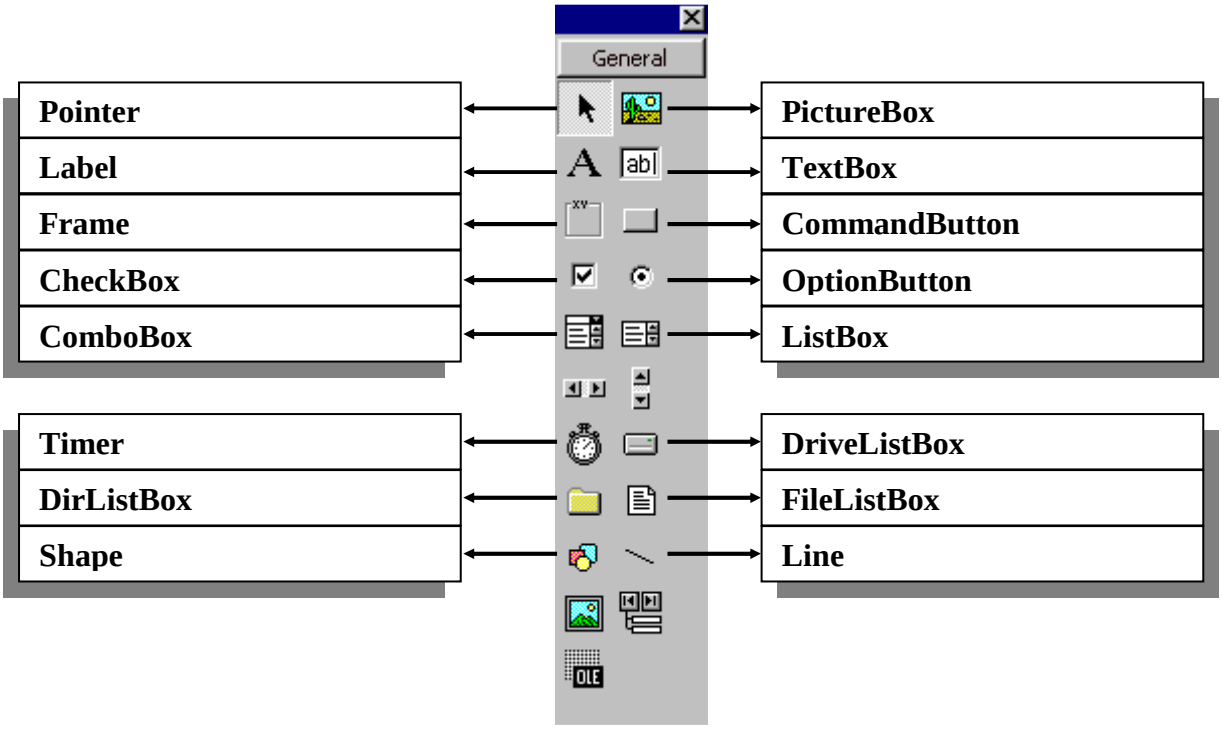
: Eğer kapanmışsa **Toolbox** (bileşenlerin bulunduğu araç çubuğunu) açar.

View Menüsü altında da aynı butonlar vardır.



TOOLBOX

İçerisinde form üzerine yerleştirilmek için bulunan farklı işlev ve amaçları olan bileşenlerin bulunduğu araç çubuğudur.



Pointer : Form üzerindeki nesneleri seçmek için kullanılır.

PictureBox : Çeşitli formatlardaki resimlerin görüntülenmesini ve çizim nesneleri aracılığı ile çizim yapılabilmesini sağlar.

Label : Daha çok kullanıcıya form üzerinde bilgi vermek için kullanılır. Bu nesneye kullanıcı tarafından bilgi girişi yapılamaz.

TextBox : Bilgi girişi için kullanılan bir kontroldür. Programın çalışması ve tasarımı esnasında metni kullanıcı tarafından değiştirilebilmesinin yanı sıra kesme, kopyalama, yapıştırma gibi işlemlerinde yapılmasına olanak sağlar.

Frame : Bazı kontrolleri gruplandırmak için kullanılır.

CommandButton : Bir olayın kullanıcı tarafından başlatılması için programlarda çok kullanılan bir bileşendir.

CheckBox : Kullanıcının belirli özellikleri aktif veya pasif hale getirmek için kullanılan bir kontroldür.

OptionBox : CheckBox'dan farklı olarak kullanıcının birkaç seçenekten sadece birini belirlemesini sağlayan bir kontroldür. Birkaç seçenekten birinin seçilmesini öngördüğü için tek bir tanesinin kullanılması anlamsız olacaktır. Aynı grupta en az iki adet bulunmalıdır.

ComboBox : Aşağı doğru açılabilen bir liste kontrolüdür. Genellikle değerleri daha önceden belli olan elemanların seçimi için kullanılır.

ListBox : Elemanları listelemek, sıralamak amacıyla kullanılan bir kontroldür.

Timer : Programda belirli aralıklarla aktif hale gelip belirli işleri yapabilmek amacıyla kullanılan bir kontroldür.

DriveListBox : Sistemde bulunan sürücüleri listelemeye yarayan ComboBox kontrolünden türemiş bir bileşendir.

DirListBox : Sistemde bulunan sürücülerdeki klasörleri listelemeye yarayan ListBox kontrolünden türemiş bir bileşendir.

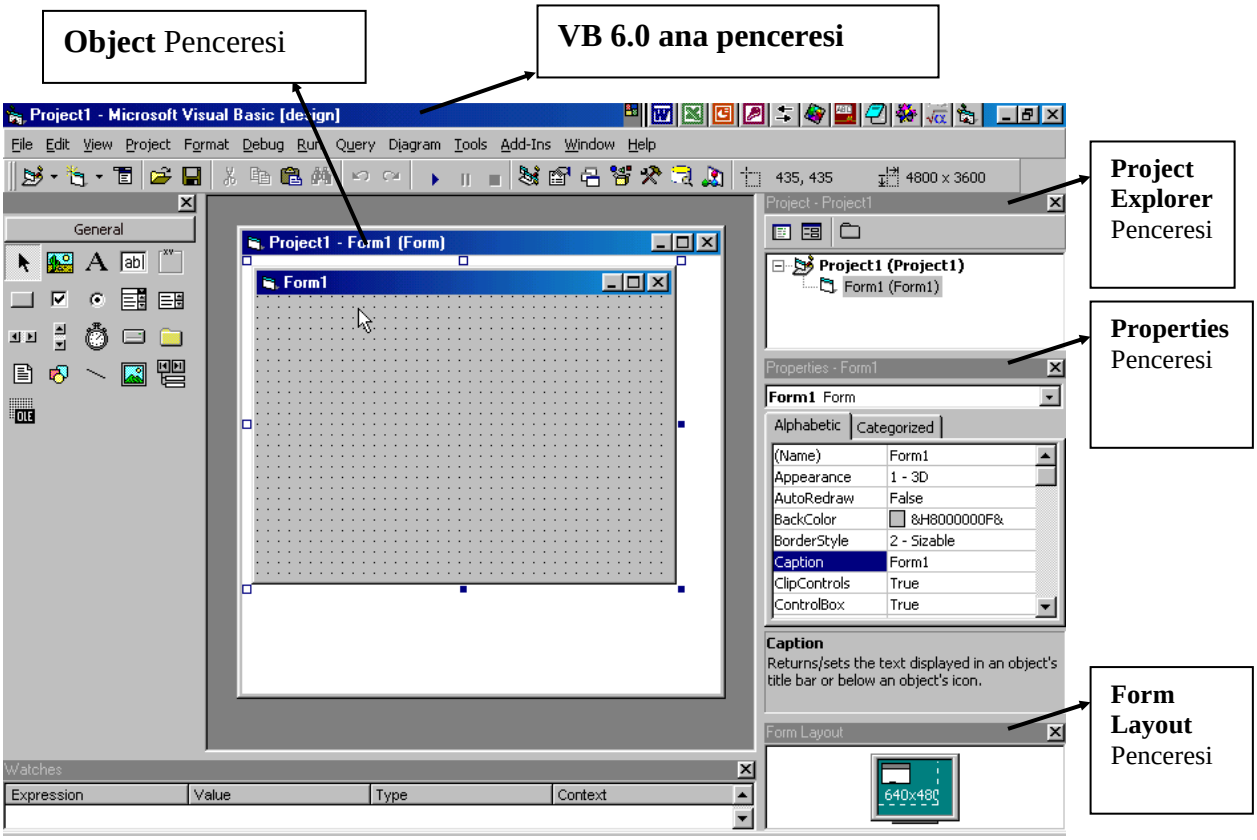
FileListBox : Herhangi bir klasörde bulunan dosyaları listelemeye yarayan ListBox kontrolünden türemiş bir bileşendir.

Shape : Grafiksel bir kontrol elemanı olup dikdörtgen, kare, elips, çember, oval kare ve oval dikdörtgen şekillerinin oluşturulmasını sağlar.

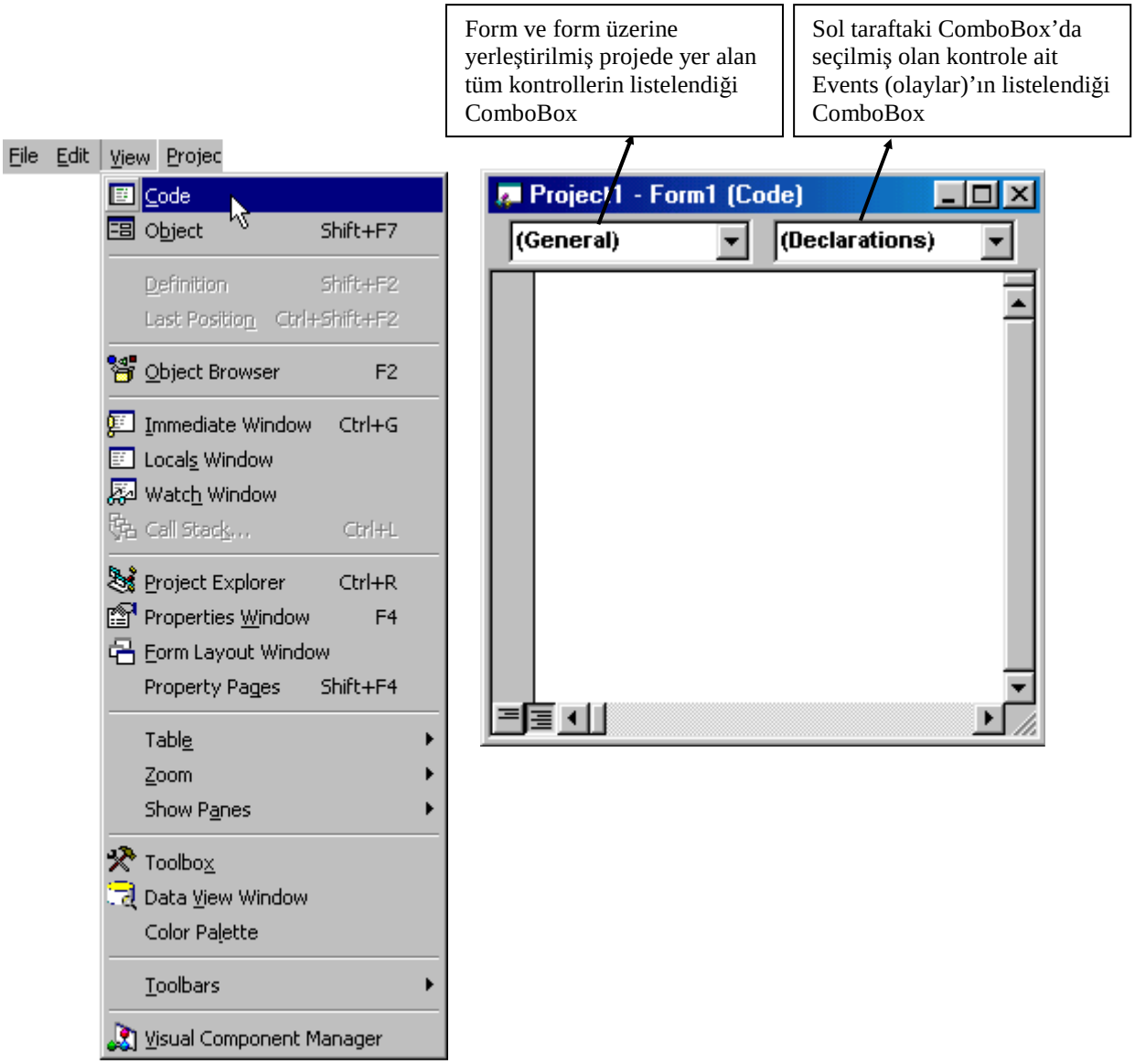
Line : Form üzerine çizgi çizmek amacıyla kullanılır.

PENCERELER

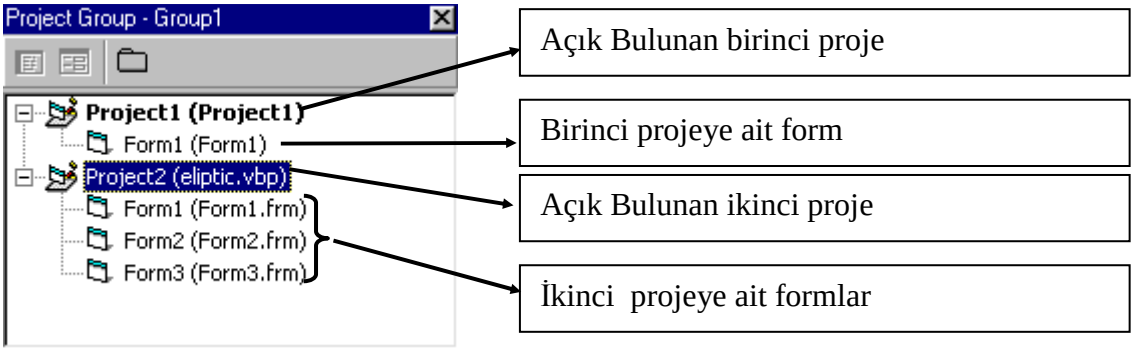
Program ilk açıldığında aşağıdaki gibi bir pencere karşımıza çıkar.



İlk açılışta görünmeyen kod penceresinin açılması için **VIEW** menüsünden **CODE** sekmesine tıklamak yeterli olacaktır.

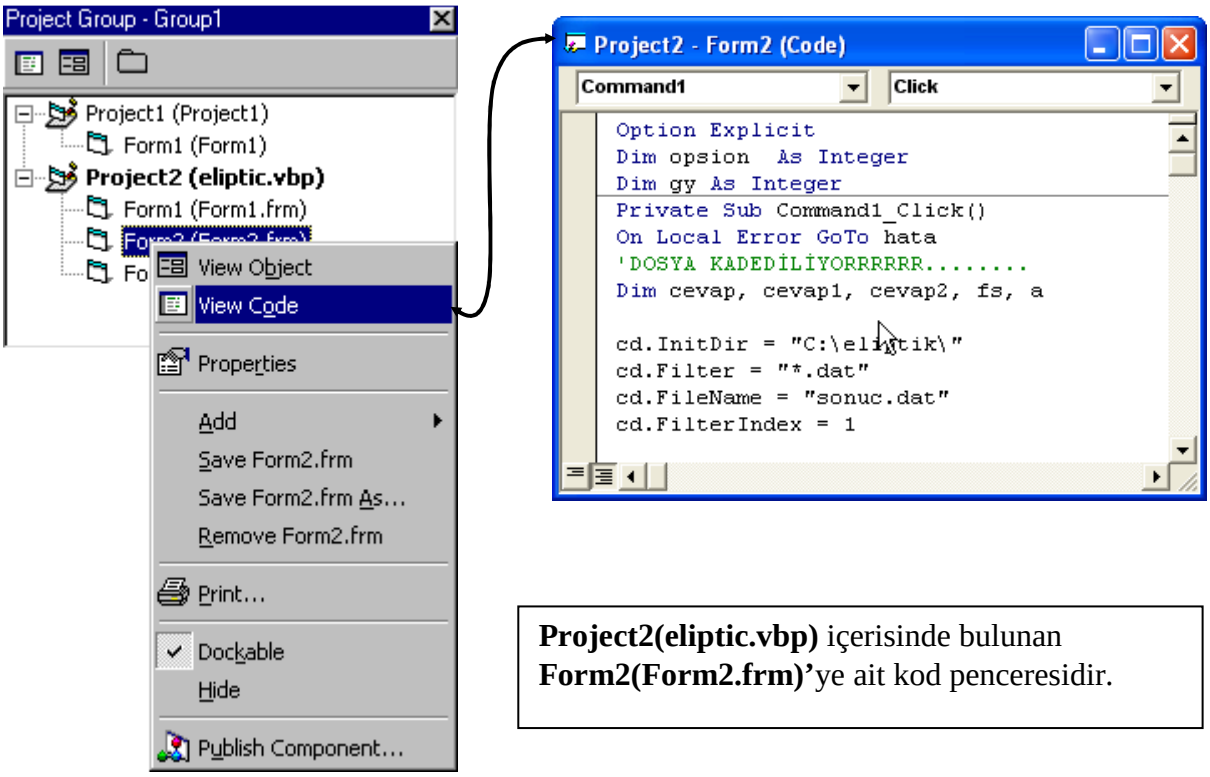


PROJECT EXPLORER PENCERESİ



Bu pencere, o anda açık bulunan projelerin ve bu projelere ait çeşitli formların, eklenmiş diğer bileşenlerinin listelendiği penceredir. Bu pencerede listelenen formlardan, görüntülemek istediğimiz üzerine **çift tıklayarak** Object Penceresinde aktif olarak görüntülenmesini

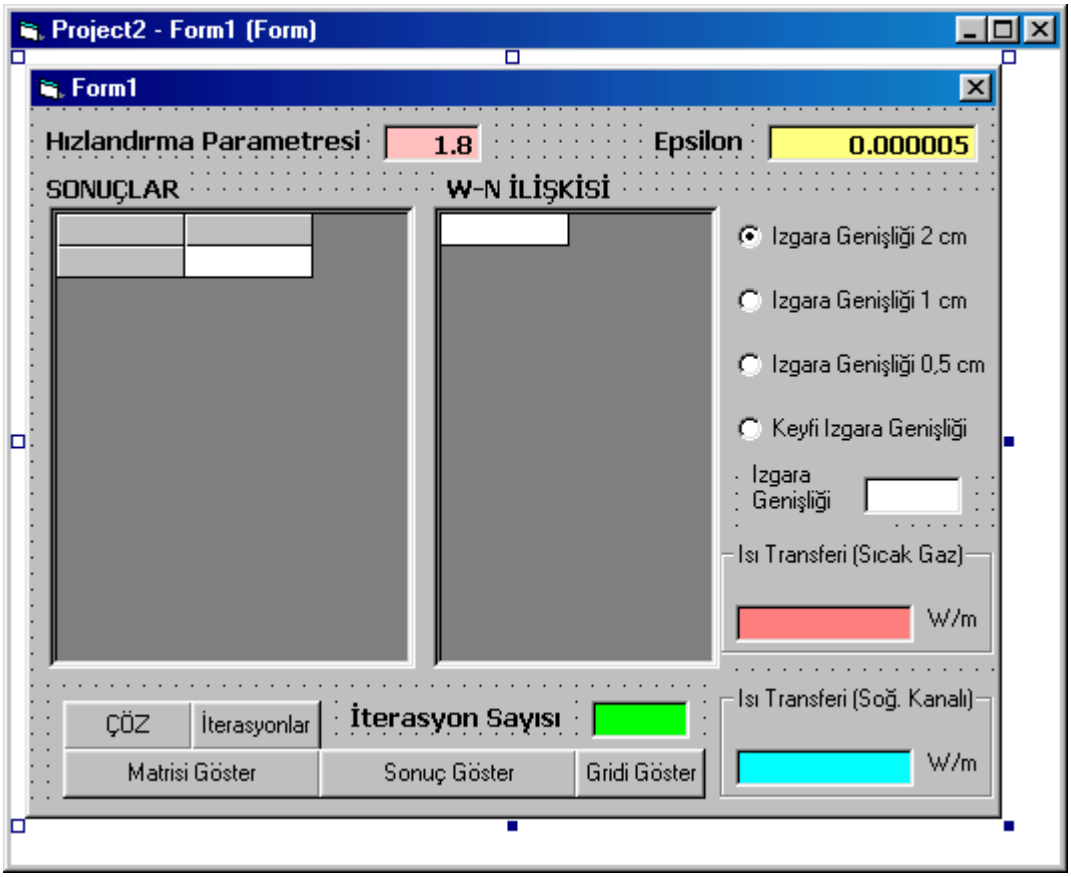
sağlayabiliriz. Bu formlara ait **kod penceresini** görüntülemek istersek, form üzerine sağ tuş ile tıklayıp çıkan pencereden **VIEW CODE** seçeneği seçilmelidir.



Project2(eliptic.vbp) içerisinde bulunan **Form2(Form2.frm)'**ye ait kod penceresidir.

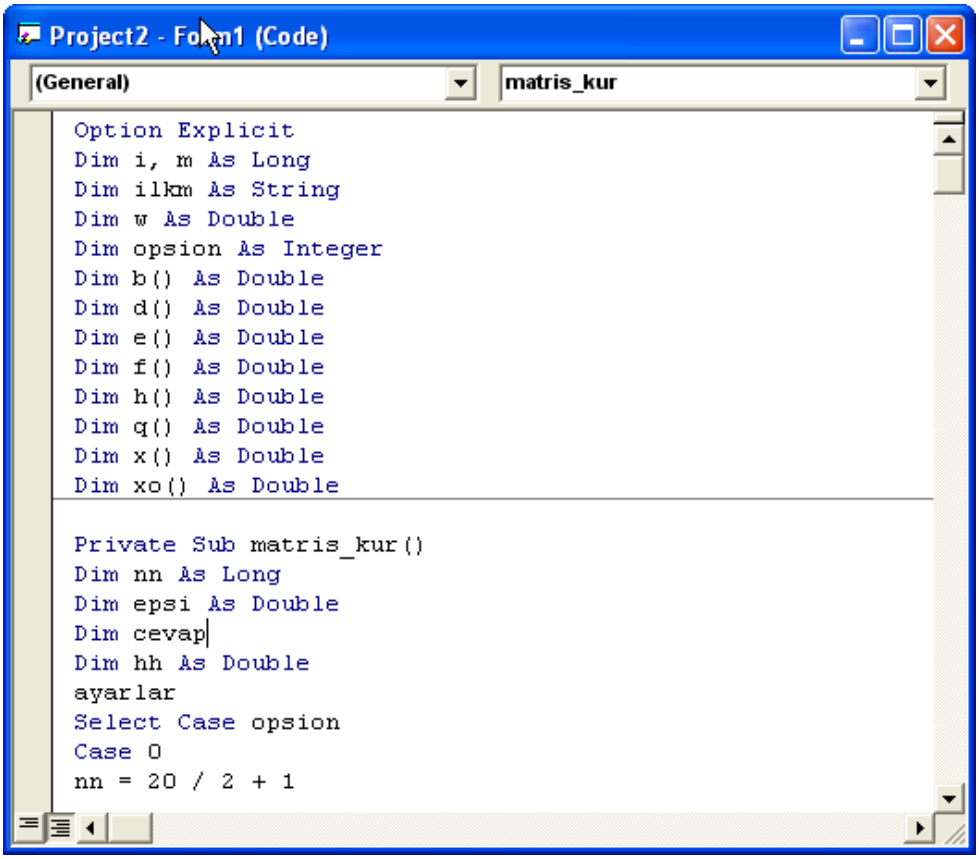
Eğer çalışmış olduğumuz proje daha önceden kaydedilmiş ise projenin adı uzantısıyla birlikte parantez içerisinde yazılacaktır. Aynı şekilde formlarda kaydedildikten sonra parantez içerisinde formun adı ve uzantısıyla birlikte görünecektir. Yukarıda gösterilen pencereden istediğimiz projeye yeni formlar ekleyebiliriz.

OBJECT PENCERESİ



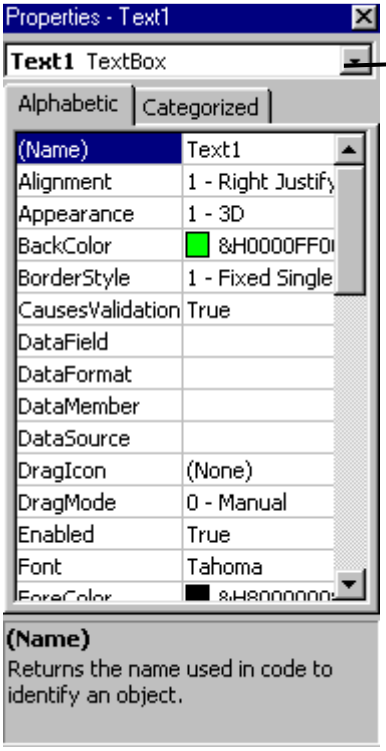
Bu pencere bize PROJECT EXPLORER PENCERESİ'nden seçtiğimiz formu ve bu form üzerine yerleştirilmiş diğer kontrolleri görmemizi sağlar. Formu tasarlarken, kontrolleri yerleştirirken, boyutlarını, renklerini ve diğer görünüm özelliklerini belirlerken yardımcı olacaktır. Program çalıştırıldığında formumuzun nasıl görüneceğini bu pencere sayesinde belirleriz.

CODE PENCERESİ



Bu pencere bize PROJECT EXPLORER PENCERESİ'nden seçtiğimiz forma ait programın işletilecek kodu yazmamızı sağlar. Yazılacak kod Prosedür adı verilen kısımlara ayrılırlar.

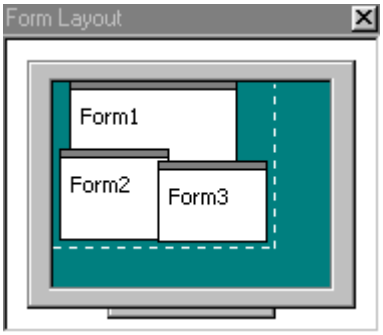
PROPERTIES PENCERESİ



Seçili olan formda yer alan tüm kontrollerin listelendiği ComboBox'dir. Buradan özelliklerini görmek ve gerekirse değiştirmek istediğimiz kontrol seçilebilir.

Bu pencere, yine PROJECT EXPLORER PENCERESİ' nden seçtiğimiz form ve bu form üzerinde yerleştirilmiş kontrollerin özelliklerini gösterir. Yan tarafta gösterilen örnekte adı **Text1** olan TextBox'a ait özelliklerin listesi görülmektedir. Bu pencerede özelliklerin iki farklı şekilde sıralanma imkanı vardır. Birincisi yan tarafta da görüldüğü gibi alfabetik sıra, diğeri de kategorilere ayrılmış listedir.

FORM LAYOUT PENCERESİ

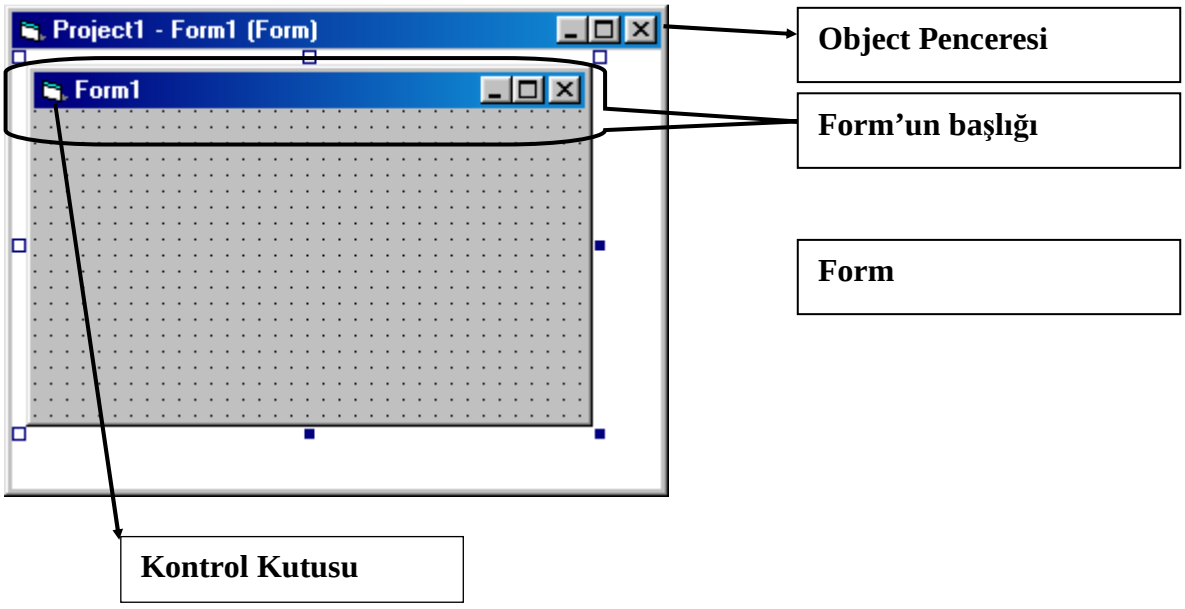


Bu pencere projemizde bulunan formların çalıştırıldıklarında ekran üzerinde hangi konumda görüneceklerini belirlememizi sağlar. Bu pencere sayesinde istediğimiz formu mouse yardımı ile başlangıç konumunu belirleyebiliriz.

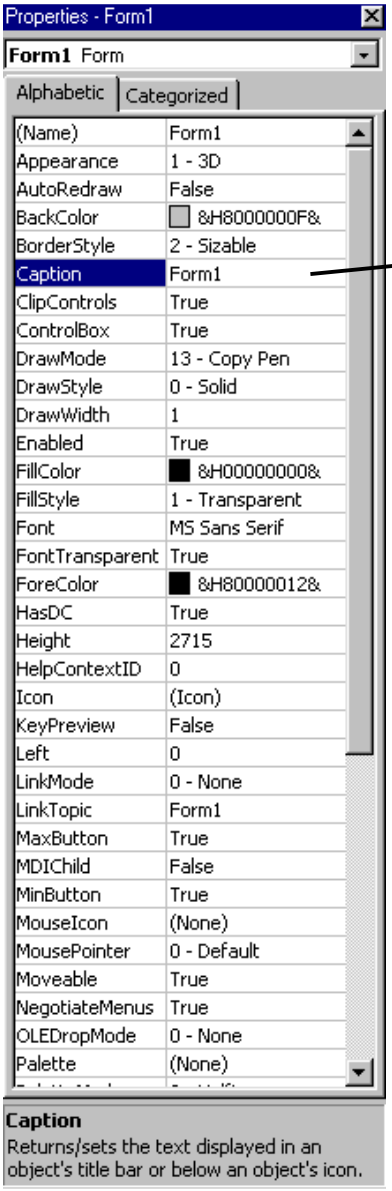
COMPONENTS (Kontroller, bileşenler)

FORM

Windows ara biriminin en temel kontrolü formlardır. Windows'ta hemen hemen her program formlar üzerinde çalışır. Diğer tüm kontroller başlangıçta standart olarak yaratılan veya daha sonra kullanıcı tarafından eklenen formlar üzerine yerleştirilir. Herhangi bir form üzerine kontrolleri yerleştirmek için **Object** penceresinin görünür olması gereklidir.



Properties (Özellikler)



Project Explorer penceresinden düzenlemek istediğimiz formu seçtiğimizde **Properties Penceresi** yandaki gibi görünecektir.

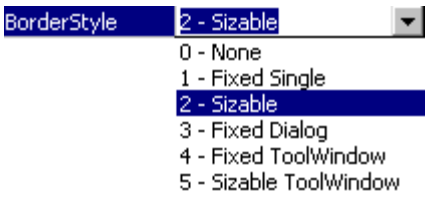
Caption (Başlık) Program çalıştırıldığında formun başlığında yazılacak yazıyı belirler. Standart olarak projede bulunan ilk form için **Form1** yazılı olarak gelir. Formu tasarlarken veya program çalışırken başlığı değiştirmek mümkündür.

Program çalışması esnasında bu değişiklik;

Form1.Caption = "Başlık"

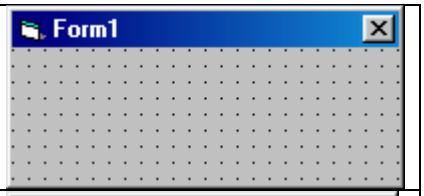
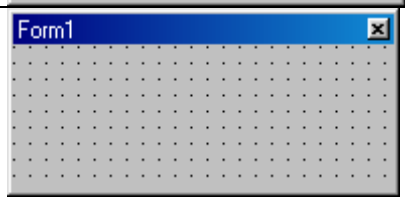
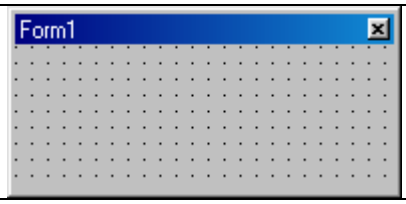
Olarak yazılır.

BorderStytle Formun çerçeve şeklini belirleyen özelliktir.



BorderStyle yukarıda listelene 6 özellikten bir tanesiyle belirlenebilir. Standart olarak 2. seçenek seçili olarak gelir.

0- None	Çerçevesi, başlığı, Kontrol kutusu Ekranı Kaplama (Max.), Simge durumuna küçült (min.), düğmeleri olmayan bir for oluşturur.	
1 – Fixed Single	Boyutları değiştirilemeyen fakat konumu kullanıcı tarafından değiştirilebilen bir form yaratır.	Form1
2 - Sizable	Formun tüm özelliklerinin kullanıcı tarafından kullanılmasına imkan verir.	Form1

3 – Fixed Dialog	1 seçeneğinden farkı min. Ve max. Butonlarının olmaz ve form görev çubuğunda görünmez.	
4 – Fixed ToolWindow	Formun başlığı normalden biraz daha küçük olur. Ayrıca kontrol kutusu bulunmaz. Daha çok ToolBox gibi araçların bulunduğu pencerelerdir.	
5 – Sizable ToolWindow	4 seçeneğinden farklı olarak kullanıcı pencerenin boyutu değiştirebilir.	

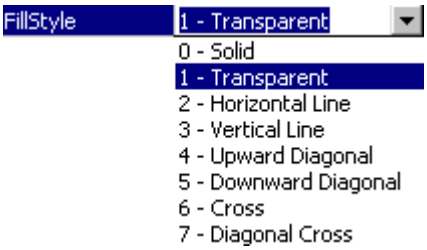
MaxButton, MinButton Formun sağ üst köşesindeki maximize ve minimize düğmelerinin görüntülenip görüntülenmemesini sağlar. Değeri *True* ise görüntülenecektir.

ControlBox Formun sol üst köşesindeki kontrol kutusunun görüntülenmesini sağlar. Eğer bu kutu görüntülenmiyorsa kullanıcı formu *Alt + F4* tuşları ile kapatamaz.

AutoRedraw False ise form üzerine başka form geldiğinde veya form boyutlandırıldığında form üzerine yapılan yazım ve çizimlerin yenilenmeyeceğini gösterir. Bu özellik form üzerine yerleştirilen kontrolleri kapsamaz. Sadece *Show* komutu kullanılarak form üzerine yaz yazıldığında veya *Line, Circle* gibi komutlarla çizilmiş şekilleri kapsar.

Name Formun kod yazımı sırasında tarif edilebilmesini sağlar. VB form isimlerini ekleniş sırasına göre *Form1, Form2, ...* şeklinde otomatik yapacaktır. Eğer istenirse bu isimler değiştirilebilir. *Caption* özelliği ile karıştırılmamasına dikkat edilmelidir. İsim olarak mümkün olduğu kadar kısa, türkçe karakter içermeyen tek kelimeden oluşan isimler tercih edilmelidir.

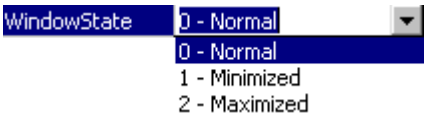
FillColor, FillStyle Circle ve line komutları kullanılarak çizilen şekillerin renkleri ve desenlerini belirler.



FillColor olarak farklı renkler seçildikten sonra FillStyle'den de 1. seçenekten farklı bir seçenek seçmek gereklidir.

0	Solid		Tam dolu
1	Transparent		Üzerinde bulunduğu yerin rengi
2	HorizontalLine		Yatay çizgili
3	VerticalLine		Dikey çizgi
4	UpwardDiagonal		Sola eğik çizgi
5	DownwardDiagonal		Sağa eğik çizgi
6	Cross		Kareli
7	DiagonalCross		Çapraz kareli

WindowState Formun çalışması üç durumda olur ve bu durumlar WindowState özelliği ile belirlenir.



Normal seçildiğinde form ekranın belirlenmiş yerinde, Minimized seçildiğinde form simge durumunda, maximized seçildiğinde ise form ekranın tamamını kaplayacaktır.

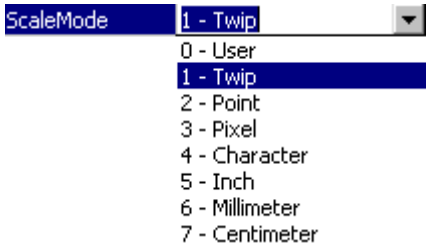
KeyPreview Form aktifken klavyeden basılan tuşlardan formun haberdar edilip edilmemesini belirleyen bir özelliktir. **False** ise herhangi bir kontrol üzerindeyken basılan tuşlar sadece o kontrole ait *key* olaylarını meydana getirir. **True** ise kontrolden önce forma ait *key* olaylarını meydana getirir.

Enable False ise form ve üzerindeki tüm kontroller kullanılmaz durumda kalacaktır. True olması halinde forma ve kontrollere ait olayların gerçekleşmesine izin verilmiş olur.

Font Form üzerine P*rint* komutu ile yazılacak yazının yazı tipini ve özelliklerinin belirlenmesi amacıyla kullanılır. Ayrıca forma yeni yerleştirilen kontrollerin yazı tipleri de seçilen yazı tipi ve özellikleri ile aynı olacaktır. Örneğin form üzerine bir çok label koymak istiyorsunuz ve her birinin yazı tipi özelliklerinin standartta kabul edilenlerin yerine farklı tipte olmasını istiyorsunuz. Bu durumda formun yazı tipini istediğiniz gibi değiştirdikten

sonra kontrolleri eklemek gerekecektir. Forma ait yazı tipini değiştirmek daha önceden yerleştirilmiş kontrollerin yazı tipinin değişmesine neden olmaz.

ScaleMode



Form üzerinde kullanılacak ölçek birimini belirlemeye yarar. Sekiz farklı alternatif vardır. İlki kullanıcı tarafından yaratılan ölçektir. Standart olarak **Twip** seçilidir.

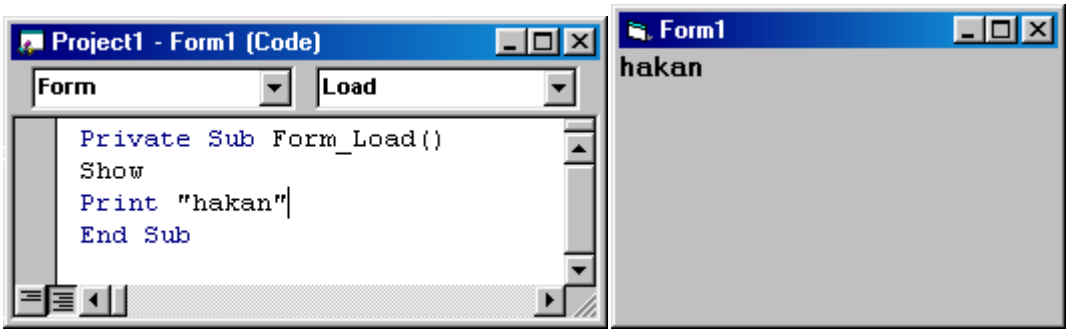
Twip : Ekran özerinde kullanılan ölçü birimlerinden bir tanesidir. Printerden çıktısı alınan 1 cm uzunluğunda bir çizginin 1/567'si veya 1 inch'in 1/1440'dır.

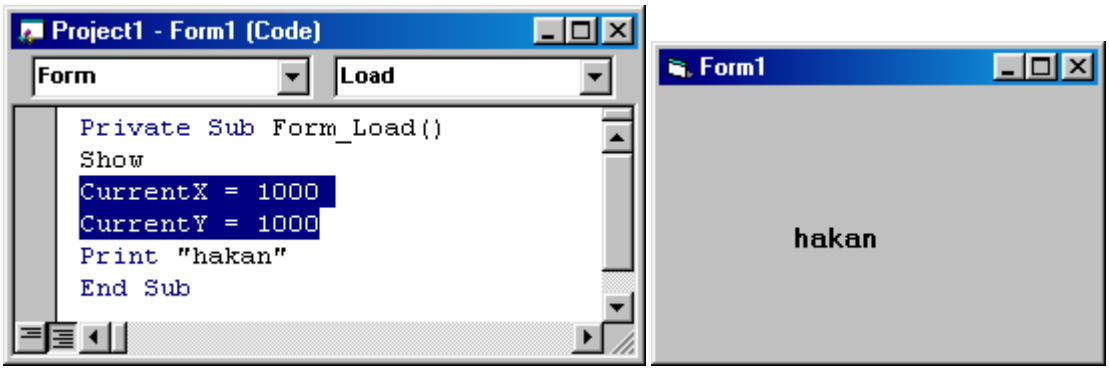
Point : Genellikle fontların boyutlarını belirtirken kullanılır. 1/72 inch'dir.

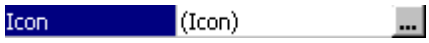
Pixel : Ekran çözünürlüğüne bağlı en küçük ölçü birimidir.

Character : Düşeyde 120 twip yatayda 240 twip'e eşit olan bir ölçü birimidir.

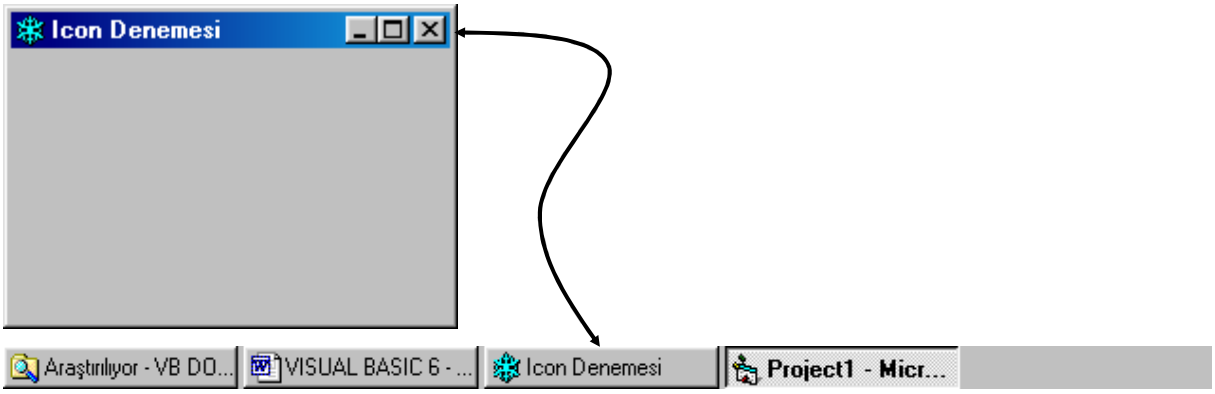
CurrentX, CurrentY Form üzerine yapılan başlangıç noktası olamayan yazım ve çizimler o anda aktif olan pikselden başlar. Bu aktif pikselin koordinatları CurrentX, CurrentY özellikleri belirler. Bu özellikler Properties penceresinde görünmezler, sadece kod yazarak kullanılırlar.





Icon Formu kullanıcıya tanıtacak iconun belirlenmesini sağlar. Form minimize edildiğinde ve *Alt + Tab* tuşları ile geçişlerde formu temsil eder. Form için uzantısı ***ico*** olan bir icon seçildiğinde forma ait properties penceresinde 

olarak görünecektir. Kontrol kutusu da seçilen icon olarak görünecektir.



MousePointer Form üzerinde mouse iler gezinirken işaretleyicinin nasıl görüneceğini belirler.

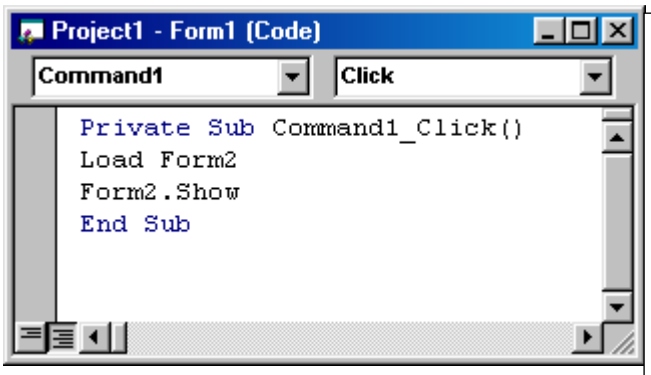
Visible Formun görünüp görünmeyeceğini belirler. False ise görünmez.

EVENTS (OLAYLAR)

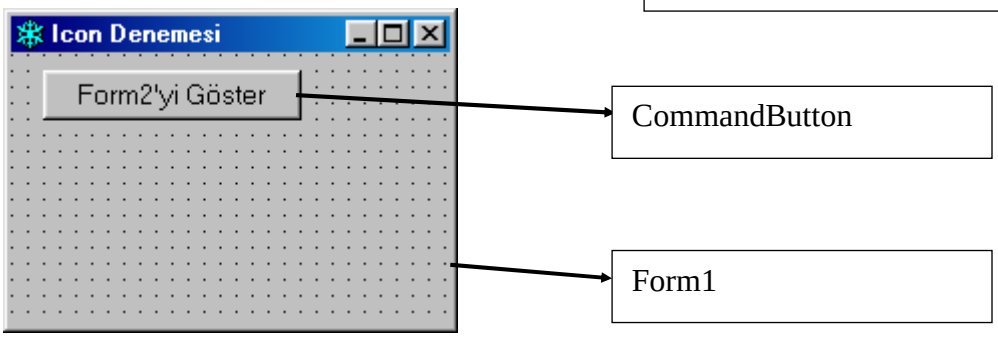
Forma ait bazı önemli olaylar kısaca anlatılacaktır.

Load()

Form ilk defa yüklendiğinde bu olay meydana gelir. Formun *visible* özelliği ile gizlenip sonradan gösterilmesi bu olayı meydana getirmez. Eğer projenizde birden fazla form varsa program çalıştığında ana form yüklenir ve görüntülenir diğer formlar yazılacak kod ile yüklenmeyi bekler. Program ilk çalıştırıldığında ana formun yüklenmesinin yanı sıra bu olay kod yazılarak da gerçekleştirilebilir. Örneğin, projenizde iki tane formunuz var. Birinci form ana form olması nedeniyle otomatik yüklenecektir. İkinci formun yüklenip görüntülenmesi için şu şekilde bir kod yazılabilir:

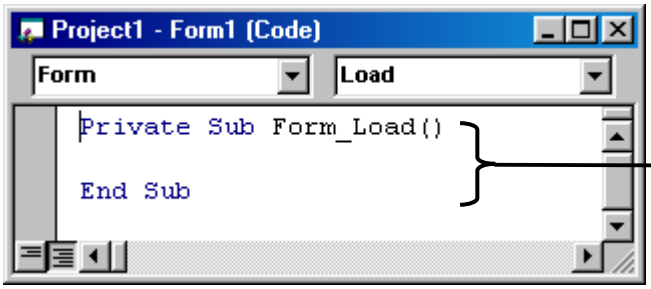


Form1 Üzerine yerleştirilen bir CommandButton'nun *Click* olayında gerçekleştirilmek üzere iki satırlık kod yazılmıştır.
Load Form2 : Form2'yi belleğe yükleyecek fakat bu görüntülenmesi için yeterli değildir.
Form2.Show : Form2'nin görüntülenmesini sağlayacaktır.



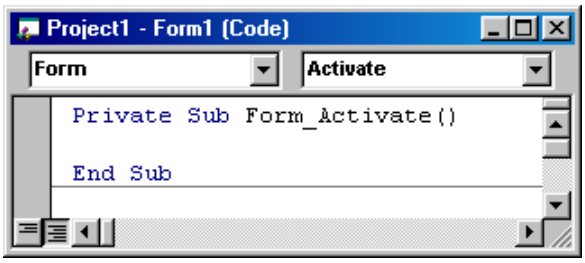
Yukarıdaki örnekte **Form2.Show** komutu da tek başına form2'nin yüklenmesi için yeterli olacaktır.

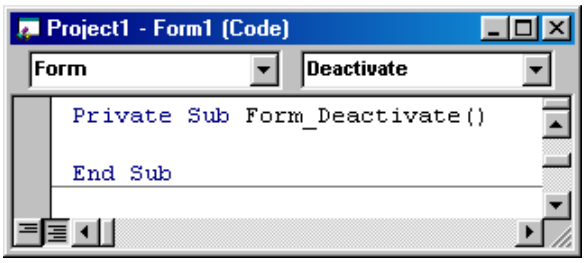
Bu prosedürde Form yüklenip ekranda görüntüleninceye kadar gerçekleşmesi gereken komutlar bilgisayara bildirilir. Örneğin form üzerine yerleştirilen kontrollerin boyutlarının, konumlarının değiştirilmesi gibi...



Form Load olayı gerçekleştiğinde yapılacak işleri kod penceresinde bu iki satır arasına yazılmak zorundadır. İlk satır **Load** prosedürünün başladığını son satır ise bittiğini gösterir.

Active(), Deactive() Hazırladığınız program birkaç formdan oluşuyor ise aynı anda bu formlardan sadece bir tanesi aktif diğerleri deaktif olacaktır. Formlar arasında geçiş yaptıkça aktivitede değişecektir. Bir form aktif olurken bir diğeri deaktif olacaktı. Bu olaylar meydana gelirken program tarafından çalıştırılacak komutlar;





yukarıda gösterilen prosedürlerde yazılması gereklidir.

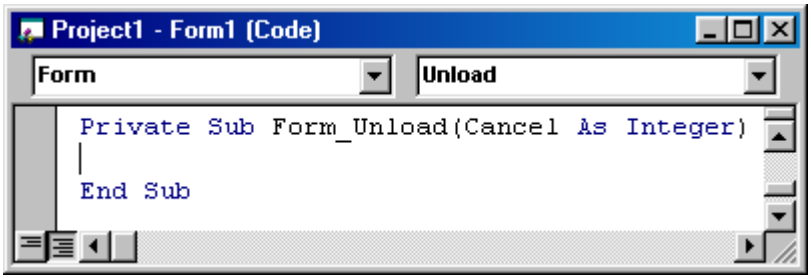
Unload (Cancel As Integer)

Form herhangi bir şekilde kapatılmaya çalışıldığında bu olay gerçekleşir. Bu olay altında yazılacak kod aşağıdaki satırlar arasında olmalıdır. Bu prosedür genellikle form ve dolayısıyla program sonlandırılmadan önce kullanıcının yapmayı unuttuğu işlemlerin hatırlatılması amacıyla kullanılır. Örneğin Word'de bir belge yazdınız ve kaydetmeden programı kapatmak istediniz. Bu durumda program sizi yapılan çalışmaların kaydedilip kaydedilmemesini soracaktır. Böylece yapılan çalışmaların yanlışlık sonucu kaybolması engellenmiş olacaktır.

Form **Unload** olayı şu durumlarda meydana gelir:

1. Formun sol üst köşesindeki kontrol kutusundan *Kapat* seçildiğinde, Sağ üst köşedeki çarpı tıklandığında veya *Alt+F4* tuşları kullanıldığında
2. Programın herhangi bir yerinde **Unload** komutu kullanıldığında
3. Windows görev yöneticinden "Göreve son ver" seçeneği seçildiğinde
4. Windows'tan çıkılmaya çalışıldığında

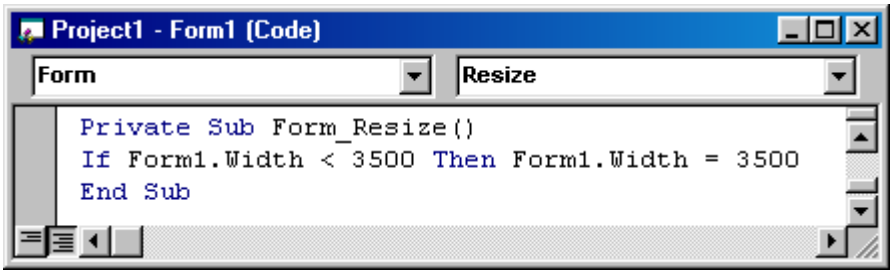
Programın herhangi bir yerinde **End** komutu kullanarak programı sonlandırırsanız bu olay gerçekleşmez.



Yanda gösterilen prosedürde;
Cancel =True
Komut satırı yazılırsa program sonlanmayacaktır.

Resize() Formun boyutlarının değişmesi (Genişlik, yükseklik) veya minimize ve maximize olması durumunda bu olay gerçekleşir. Bu prosedürde formun boyutu değiştikçe kontrollerinde boyutunun ve yerlerinin yeni oluşan boyutlara uyması için gerekli değişiklikler bulunabilir veya formun belirli ebatlardan daha küçük/büyük olması önlenebilir. Örneğin

aşağıdaki örnekte formun genişliğinin 3500 twip'ten küçük olması engellenmiştir. Kullanıcı formun boyutlarını istediği gibi değiştirebilir. Eğer formun genişliği 3500 twip'ten küçük olarak boyutlandırılırsa formun genişliği 3500 twip olarak otomatik olarak değiştirilir.



Methods (Metotlar)

Line Bu komut yardımı ile çizgi ve dikdörtgen çizilebilir. Çizgi çizmek için

Line (x1,y1)-(x2,y2), [renk]

Komut satırı yazılabilir. Burada x1,y1,x2,y2 formun *ScaleMode* özelliğinde belirtilen ölçü biriminde koordinatları belirtmektedir. X1,y1 koordinatından x2,y2 koordinatına köşeli parantezi içerisinde verilen renkte bir çizgi çizer. Eğer renk belirtilmez ise formun ***ForeColor*** özelliğinde seçili renkte bir çizgi çizer.

Dikdörtgen çizmek için ise

Line (x1,y1)-(x2,y2), [renk], B

Komut satırı yazılmalı, çizilen dikdörtgenin içinin doldurulması için ise;

Line (x1,y1)-(x2,y2), [renk], BF

Komut satırı kullanılmalıdır.

Eğer ***ForeColor*** özelliğinde seçili renkte dikdörtgen çizmek istiyorsanız renk kısmını boş bırakmalısınız.

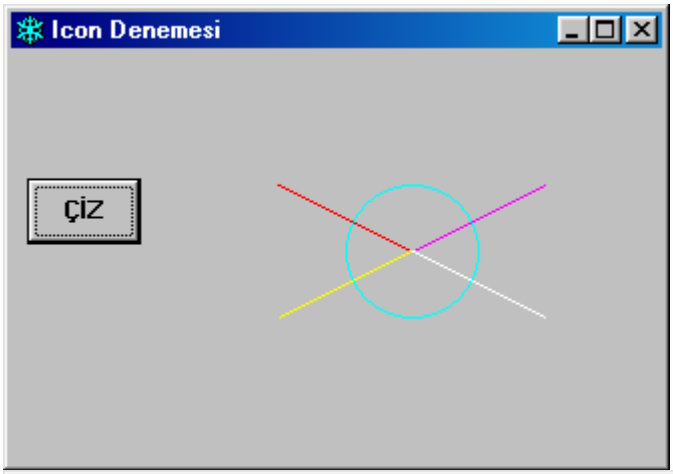
Line (x1,y1)-(x2,y2), , BF

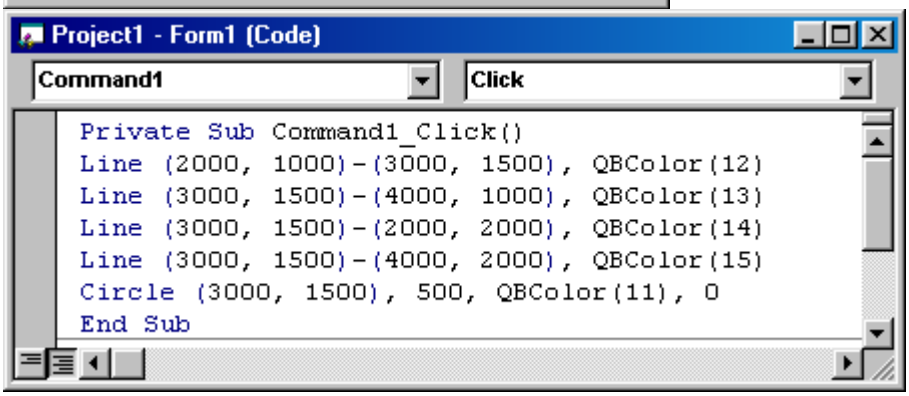
Circle Çember, elips veya yay parçası çizmek için kullanılır. Komut satırı;

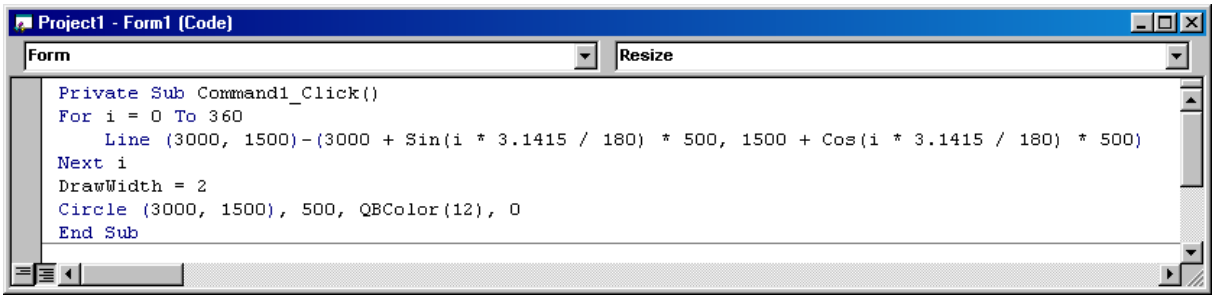
Circle (mx,my),r,[renk],a,b

Mx,my merkezli r yarıçaplı yayı a açısından başlayıp b açısına kadar verilen renkte çizer. Bu ifadedeki a ve b açıları radyan cinsindendir.

Bu konu ile ilgili aşağıdaki örneği yapınız. İlk olarak form üzerine bir adet *CommandButton* ekleyip formun boyutlarını Width = 4995 ve height = 3495 olacak şekilde ayarlayınız. CommandButton'nun *Click* olayına aşağıdaki satırları yazıp programı çalıştırınız.





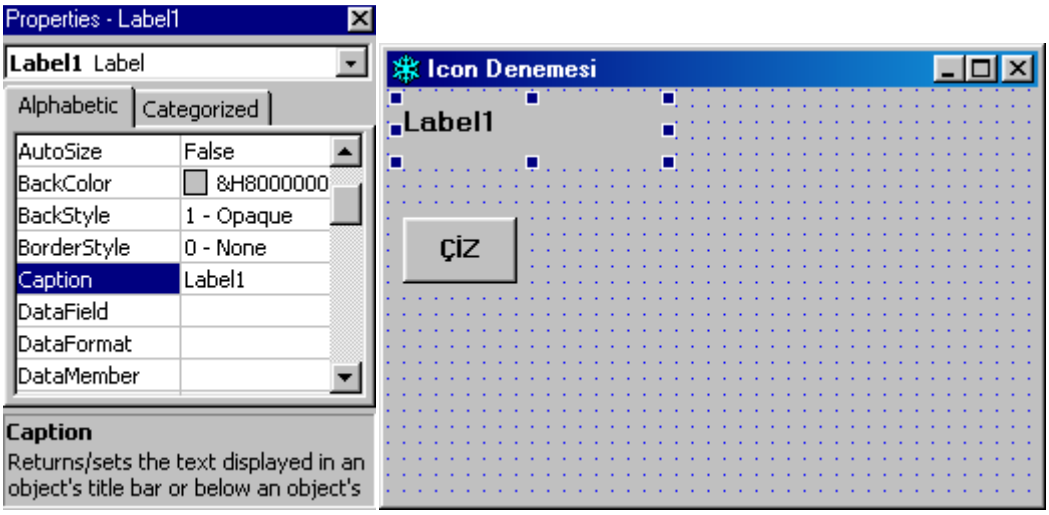


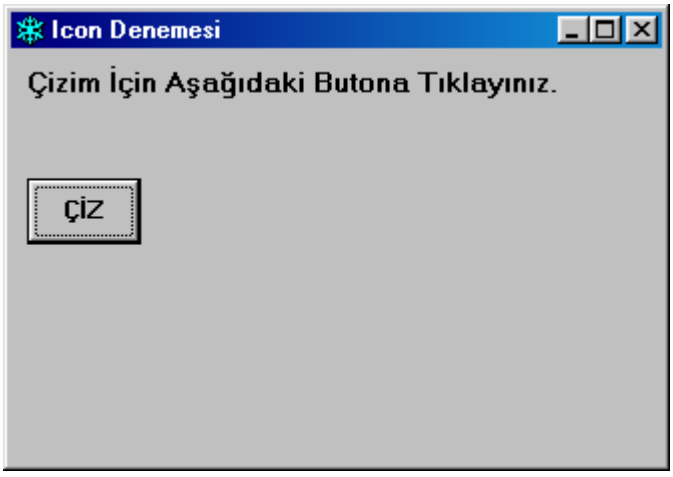
Label

Programın çalışması esnasında kullanıcıya bilgi vermek amacı ile kullanılan bir kontroldür. Giriş yapılması mümkün değildir fakat programın çalışması aşamasında özellikleri değiştirilebilir.

Properties (Özellikler)

Caption Nesnenin üzerindeki yazı bu özellik ile belirlenir. Aşağıdaki örnekte olduğu gibi nesne form üzerinde seçildiğinde Properties penceresinde *Label'a* ait özellikleri listelenmiş olacaktır. Bu özellikler arasında *Caption* bulunarak karşısındaki ifade istenildiği gibi değiştirilebilir. Bu kontrol, diğer kontrollere açıklama yazmak için kullanılır.





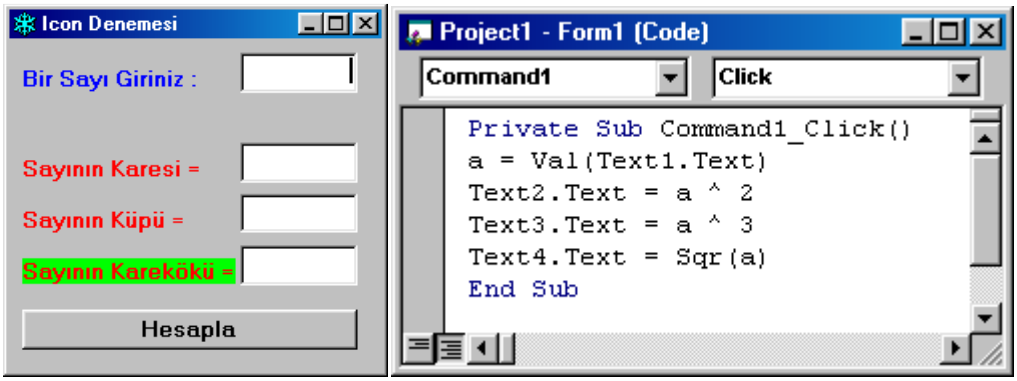
Autosize **True** ise kontrolün içeriği değiştiğinde nesnenin boyutları da otomatik olarak değişecektir.

BackStyle Bu özellik sayesinde nesnenin üzerinde bulunduğu konuma uyması sağlanır. Eğer bu özellik **0 – Transparent** ise sadece yazı görünün label'in kendisi görünmez. **1 – Opaque** ise label kendi rengini gösterir.

ForeColor Kontrol üzerine yazılacak yazının rengini belirler.

BackColor Label'in kendi renginin belirlenmesini sağlar.

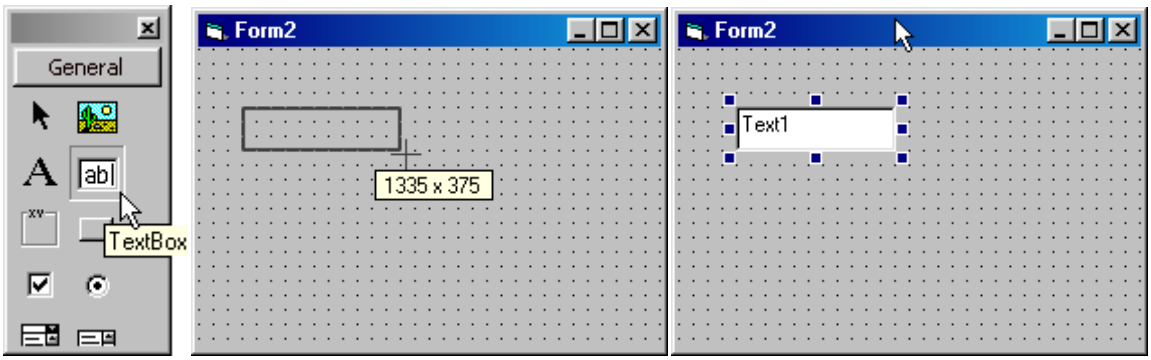
Aşağıda verilen örneği tasarlamaya çalışınız.



TextBox

Bilgi girişi için kullanılan en önemli kontrollerden biridir. Tasarlanması sırasında veya programın çalışması esnasında içeriği kullanıcı tarafından değiştirilebilir. Aynı zamanda kes, kopyala yapıştır özelliklerini kullanabilir, birden fazla satır yazılabilir.

TextBox nesnesini form üzerine yerleştirebilmek için ToolBox'dan **TextBox** seçilir ve form üzerine mouse'un sol tuşu basılı tutularak bir dikdörtgen alan çizilir.

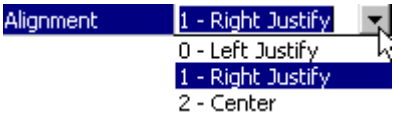


Properties (Özellikler)

Text Properties penceresinden **text** özelliği TextBox nesnesinde yazdırılacak yazıyı belirler. Label kontrolünde verilen örnekteki gibi textBox kullanıcının programı çeşitli değerler girmesine olanak sağlar. Böylece girilen bu değerler program tarafından işlenerek sonuçlar belki de yine yukarıda olduğu gibi textBox nesnesi ile kullanıcıya aktarılır.

ToolTipText Kullanıcının mouse ile üzerine geldiği nesne hakkında bilgilendirilmesini sağlayan bir özelliktir. Mouse nesnenin üzerinde bir süre bekletildiğinde bu özellikte yazılmış olan yazı kullanıcıya yol gösterir.

Alignment Nesne içindeki yazının sağa, sola veya ortaya yazılmasını sağlar.



0 – Left Justify	Sola dayalı	
1 – Right Justify	Sağa Dayalı	
2 - Center	Ortalı	

Locked Text kutusunun *Locked* özelliği **true** yapıldığında kullanıcı programın çalışması esnasında text içeriğini değiştiremez. Bu özellik genellikle program tarafından üretilen sonuçların kullanıcıya aktarıldığı, sonuçların gösterildiği durumlarda kullanılır. Fakat text kutusu içerisindeki yazı seçilip kopyalanabilir.

Enabled Nesnenin aktif veya pasif olmasını sağlar. Eğer **False** ise kullanıcı içeriği değiştiremediği gibi metin de seçemez ve metin soluk renkte görünür.

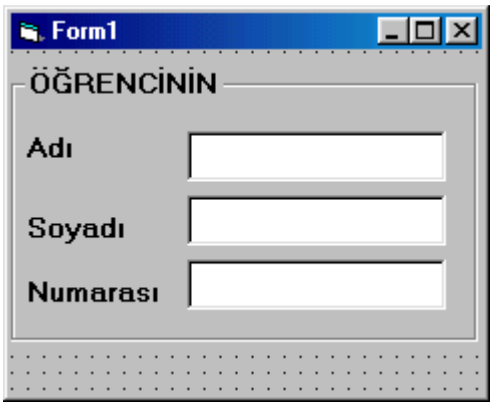
Visible True ise nesne görünün değilse görünmez.

MaxLenqth Text kutusuna girilebilecek max karakter sayısını belirler. Örneğin öğrenci numarasının girileceği bir text kutusunda bu değer 12 olmalıdır. Böylece kullanıcı daha fazla karakter girmek istese de text kutusu buna müsaade etmeyecektir.

MultiLine Eğer **True** ise kullanıcı text kutusuna istediği zaman alt satıra geçme imkanı sağlar yada text kutusuna girilen yazı bir satıra sığmıyorsa otomatik olarak yazı alt satıra devam edecektir.

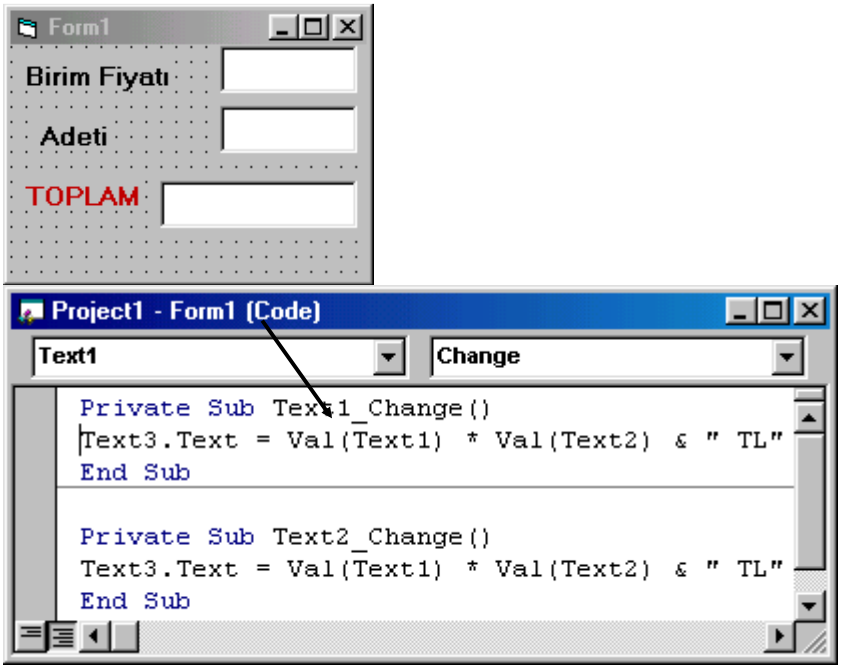
ScrollBars Eğer text kutusunun **MultiLine** özelliği True ise bu özellikte de text kutusuna ait ScrollBars seçeneklerinden uygun olan seçilmelidir.

TabIndex Form üzerinde kullanıcının ulaşabildiği her nesnenin bir tabindex'i vardır. Böylece kullanıcı nesneler arasında ***Tab*** tuşuna basarak bu nesneler arasında hareket edebilir. VB form üzerine yerleştirilen her nesne için otomatik olarak yerleştirilme sırasına göre bir tabindex oluşturur. Ama kullanıcı formu tasarladıktan sonra bu tabindex'leri istediği gibi değiştirebilir. Program çalıştırıldığında imleç Tabindex değeri 0 olan kontrol üzerinde aktif olarak başlar. Aşağıdaki formu oluşturup tabindex özelliğini inceleyiniz. Programı çalıştırdıktan sonra klavyeden ***Tab*** tuşuna basarak nesneler arasında dolaşınız.



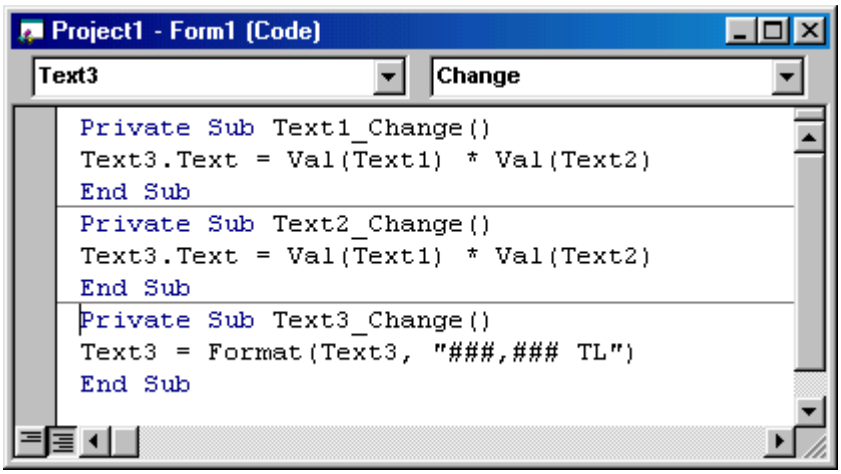
Events (Olaylar)

Hemen her nesneye ait VB tarafından başlangıçta seçilmiş bir olay vardır. Text kutusu için bu olay **Change()** olayıdır. Bu olay text kutusunun içeriği değiştiğinde meydana gelir. Bu değişiklik gerek kullanıcı bilgi girişi yapsın gerekse programın ürettiği sonuçların text kutusuna aktarılması olsun her iki durumda da oluşur. Eğer form üzerine yerleştirmiş olduğunuz bir nesnenin üzerine iki kez tıklarsanız o nesne ile ilgili kod penceresi açılarak varsayılan olay prosedürü oluşturulur. Bu olayı kavramak için şöyle bir örnek yapalım. Aşağıdaki formu tasarlayıp kodları yazınız.



Text3

Kodları aşağıdaki gibi değiştirip sonucu karşılaştırınız.



Click() Kullanıcı programın çalışması esnasında mouse imleci ile nesne üzeride iken mouse'un butonlarından birine tıklaması sonucu oluşur.

DblClick() Yukarıdaki olayın mouse'un iki kere tıklanması şeklindedir.

KeyDown(KeyCode** As Integer**, Shift** As Integer**) Klavyeden tuşa basmayla ilgili üç farklı olay vardır. Bu üç olay arasındaki en büyük fark zaman ve sıradır. Bir tuşa basıldığında ilk olarak meydana gelen olay **KeyDown** olayıdır ve tuşa basıldığı anda başlar be basılı tutulduğu müddetçe devam eder. Bu olay gerçekleşirken bilgisayar iki farklı değer üretir. Birincisi **KeyCode** diğer ise **Shift** değerleridir.

KeyCode Basılan tuşun Scan kodudur. Bu keyCode'lara göre uygun komutlar verilerek programın belirli tuşlara basıldığında belirle işleri yapması sağlanır.

Shift Shift, Ctrl ve Alt tuşlarının durumunu belirler. Şu değerleri alabilir;

1: Sadece Shift tuşu basılı

2: Sadece Ctrl tuşu basılı

4: Sadece Alt tuşu basılı

3: shift + Ctrl tuşları basılı

5: Shift + Alt tuşları basılı

6: Ctrl + Alt tuşları basılı

7: Üç tuşta basılı

KeyUp (KeyCode** As Integer**, Shift** As Integer**)

Bu olay basılan tuş bırakıldığında meydana gelir. Her klavye olayında oluşmaz. Yani Eğer klavyeden **A** tuşuna basılı tutulursa sürekli **A** harfi girişi yapılır ama sadece son harften sonta **KeyUp** olayı gerçekleşir.

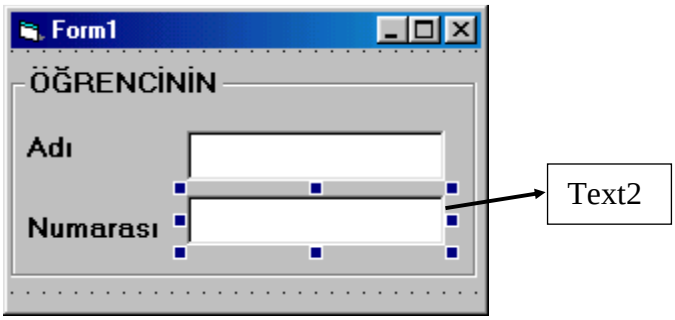
***KeyPress (KeyAscii** As Integer)*

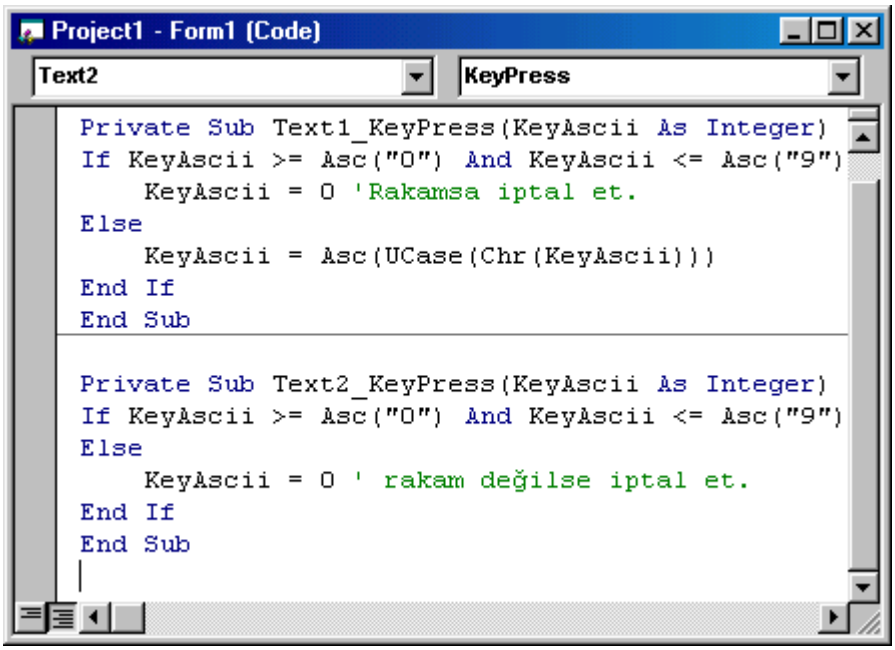
Bu olay klavyedeki tuşların bir kısmında meydana gelir. Basılan tuşun **AscII** Kodu varsa oluşur.

Klavyeden bir tuşa basıldığında sırasıyla şu olaylar oluşur: **KeyDown, KeyPress, Change, KeyUp.**

KeyPress olayı ile text kutusuna girilen bilgileri değiştirme düzeltme veya kabul etmeme gibi eylemler yapılabilir.

Bu olayla ilgili aşağıdaki uygulamayı yapınız. Text2'nin *maxlength* özelliğini 12 yapınız.





***MouseDown, MouseUp (Button** As Integer, **Shift** As Integer, **X** As Single, **Y** As Single)*

Bu olaylar; herhangi bir kontrol elemanı üzerinde iken mouse'un tuşlarına basılması ve bırakılması ile oluşur. Bu olay oluşurken yine bilgisayar otomatik olarak bazı değerleri oluşturur. Bu değerler sayesinde programcı dolayısıyla program mouse'un hangi tuşunun basıldığı hatta bu esnada klavyeden **Ctrl, Shift, Alt** tuşlarının basılı olup olmadığını da kontrol edebilir.

Button : Farenin hangi tuşuna basıldığını belirler.

1: vbLeftButton : Farenin sol tuşu

2: vbRightButton : Farenin sağ tuşu

4: vbMiddleButton : Farenin orta tuşu

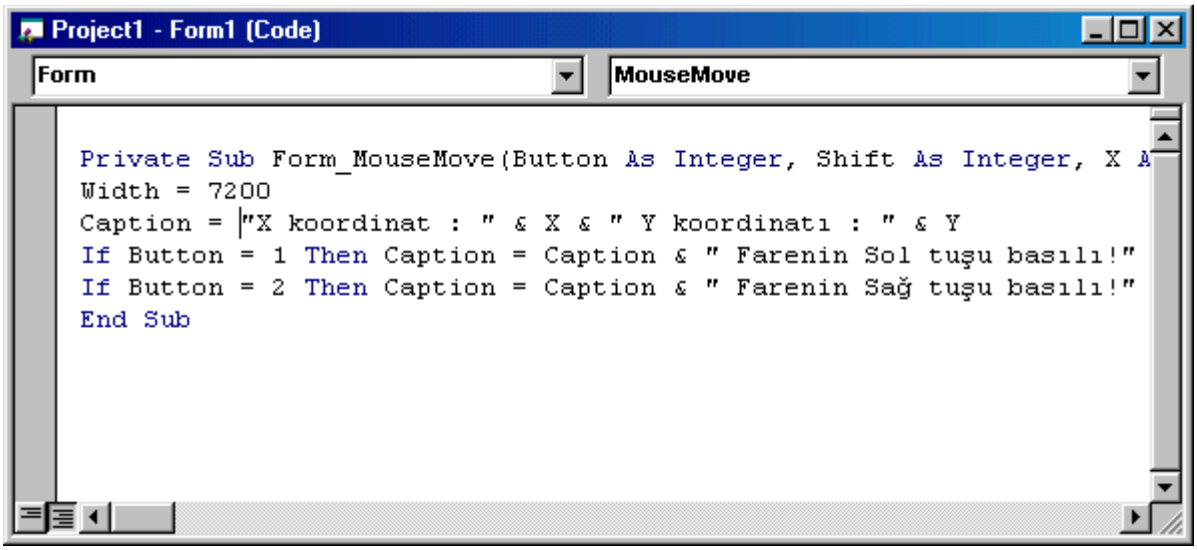
Shift : MouseDown veya MouseUp olayları oluştuğu esnada klavyeden **Ctrl, Shift, Alt** tuşlarından hangilerinin basıldığını belirler. *KeyDown* olayında açıklanan değerleri alabilir.

X, Y : Fare imlecinin konumunun koordinatlarını verir.

***MouseMove(Button** As Integer, **Shift** As Integer, **X** As Single, **Y** As Single)*

Bu olaylar; herhangi bir kontrol elemanı üzerinde iken mouse'un dolaştırılması ile oluşur. Bu olay oluşurken yine bilgisayar otomatik olarak bazı değerleri oluşturur. Bu değerler sayesinde programcı dolayısıyla program mouse'un hangi tuşunun basıldığı hatta bu esnada klavyeden **Ctrl, Shift, Alt** tuşlarının basılı olup olmadığını da kontrol edebilir.

Boş bir form yaratıp forma ait *MouseMove* olay prosedürü altında şu ifadeleri yazınız. Programı çalıştırıp inceleyiniz.

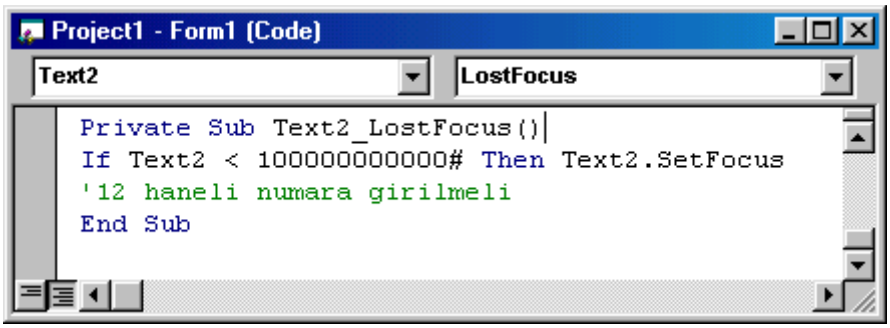


GotFocus() Klavye kontrolünün bir nesneye geçmesi ile o nesneye ait *GotFocus* olayı meydana gelir. Bu olayın oluşabilmesi için nesnenin V*isible* ve *Enabled* özelliğinin **True** olması gereklidir.

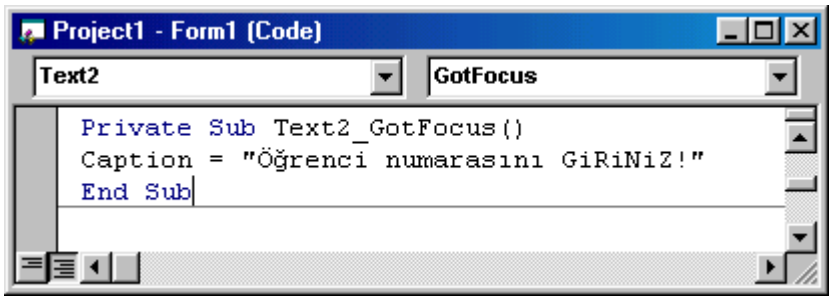
LostFocus() Bir nesnenin klavye kontrolü kaybetmesi o nesneye ait *LostFocus* olayı meydana gelir. Bu olayın oluşabilmesi için nesnenin V*isible* ve *Enabled* özelliğinin **True** olması gereklidir.

Bu olaylarla ilgili şu örnekleri inceleyelim.

KeyPress örneğindeki öğrencinin numarasını gireceği **text2** nesnesine ait *LostFocus()* olayı prosedürü altında aşağıdaki kodu yazıp çalıştırınız. Sonuçları inceleyiniz.

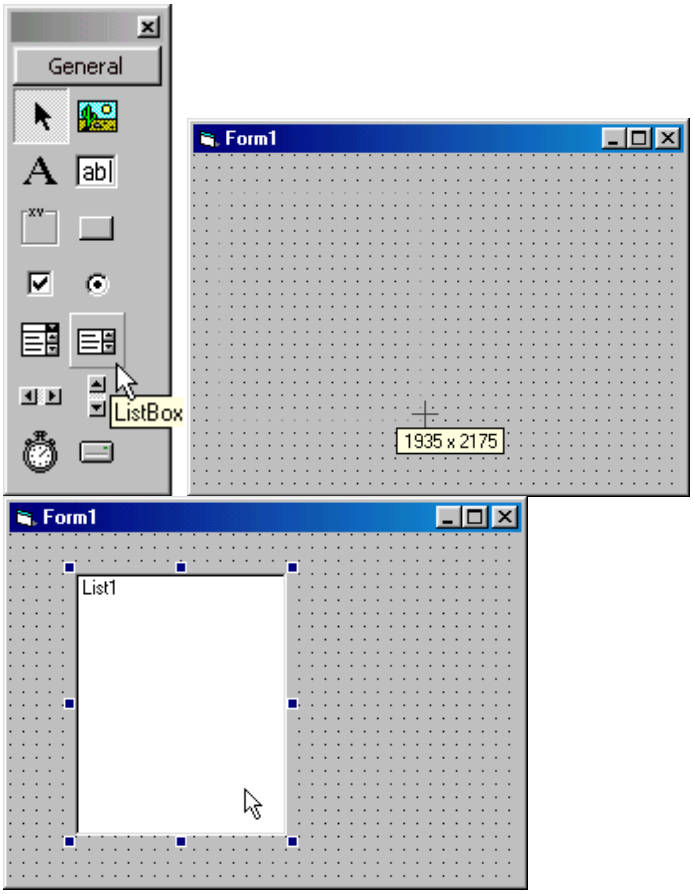


yine aynı örnekte text2 nesnesine ait *GotFocus* olayı prosedürü altında aşağıdaki kodu yazarak çalıştırınız. Sonuçları tartışınız.



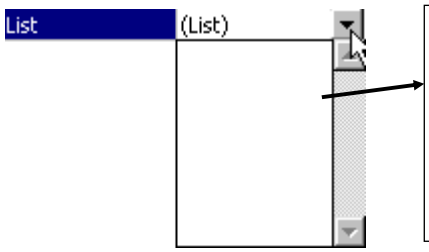
ListBox

Program hazırlanırken kullanıcıya çeşitli elemanların listesini sunmak için kullanılan bir kontroldür.



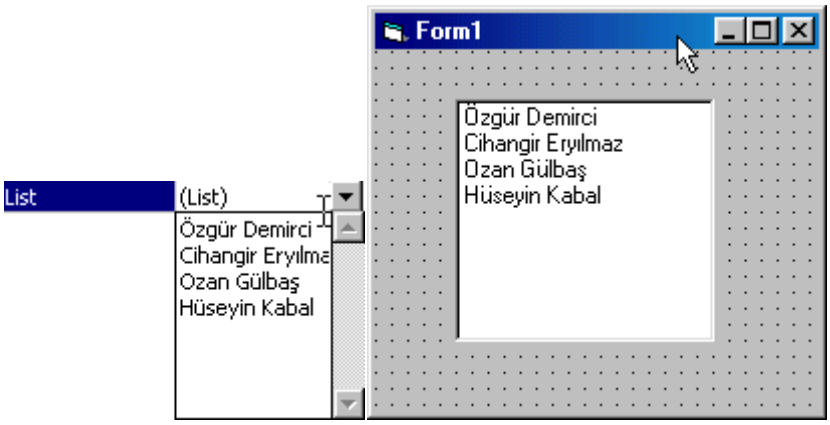
Properties (Özellikler)

List(Index) Program çalıştığında listelenecek elemanların girilmesine olanak sağlayan özelliktir.



Aşağı indirme butonuna tıklandığında boş bir liste çıkacaktır. Bu listeye istenildiği gibi veri eklenebilir. Her satırdan sonra ***Ctrl*** + ***Enter*** tuşlarına beraber basıldığında bir sonraki eklenecek eleman için alt satıra geçer.

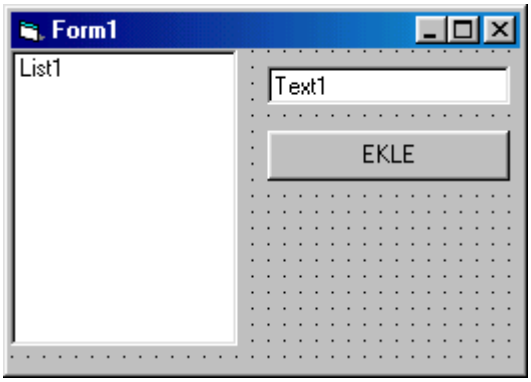
Listeye elemanlar eklendikten sonra form üzerine yerleştirilmiş ListBox'da eklenen elemanlar görüntülenecektir.



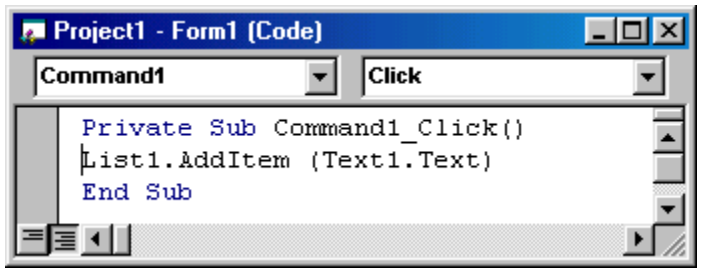
Kullanıcı listeye programın çalışması esnasında da yeni elemanlar ekleyebilir. Bunun için kod satırı şu şekilde elmalıdır;

List1.AddItem (Text1.Text)

Bu satırı inceleyecek olursak, **List1=> ListBox** nesnesinin adıdır. **Text1.Text**=> listeye eklenecek metindir. Bu konunun daha iyi anlaşılabilmesi için aşağıdaki örneği yapınız.



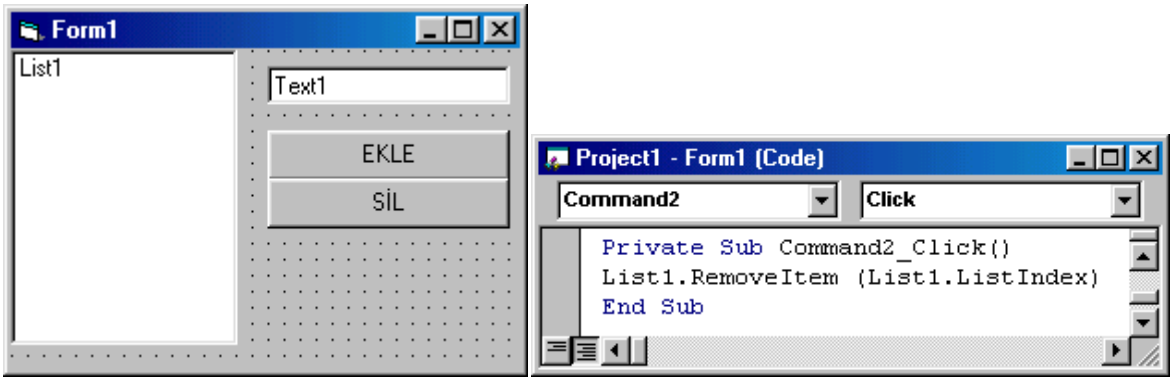
Yandaki gibi formunuza bir adet ListBox, bir adet TextBox ve bir adet de CommandButton ekledikten sonra CommandButton'un Caption özelliğini "EKLE" olarak değiştirin. Daha Sonra Ekle butonunun üzerine iki kere tıklayarak code penceresinin açılmasını sağlayın ve aşağıdaki kodu yazınız.



Programı çalıştırdıktan sonra Text1 kutusuna yazılan ifadelerin EKLE butonuna tıklandığında List1'de listelendiğini göreceksiniz.

Listeye eklenen her eleman için program otomatik olarak bir index numarası oluşturur. Bu numara sayesinde programın kod aşamasında listedeki elemanlar hakkında bilgi alabiliriz, elemanların silinmesi ve değiştirilmesi sırasında yardımcı olur.

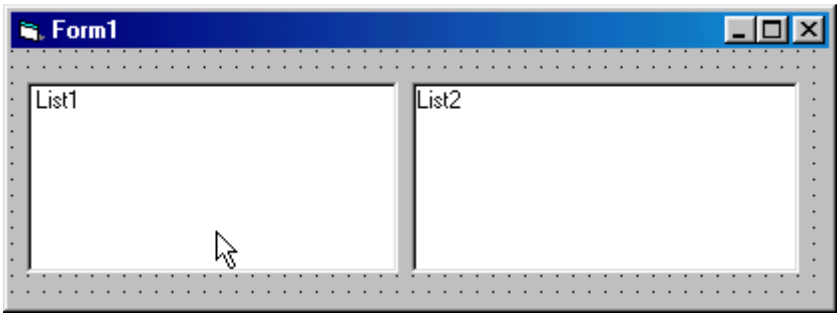
RemoveItem Index Programın çalışması sırasında listeden eleman silmek için kullanılır.Properties penceresinde görüntülenmeyen özelliklerdendir. Kod yazımı kısmında programcı tarafından yazılması gereken bir özelliktir. Bu özellik için yukarıdaki örneğe şu ilaveleri yapalım.



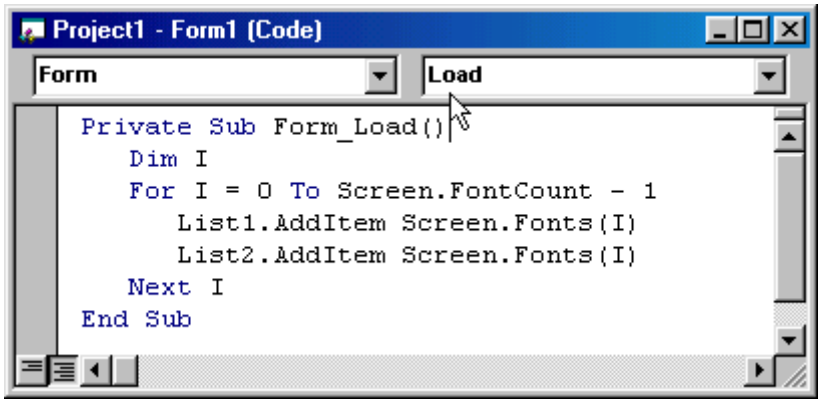
ListCount Listede yer alan elemanların sayısını verir.

Columns ListBox (liste kutusu)'da görüntülenecek kolon sayısını belirler. Bu özelliği daha iyi anlamak için aşağıdaki örneği hazırlayıp çalıştırınız.

Form üzerine aynı boyutlarda iki adet ListBox yerleştirip 2. liste kutusunun ***Columns*** özelliğini 2 olarak değiştirin.

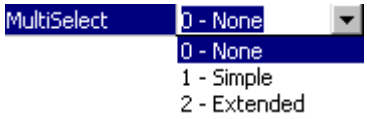


Kod penceresine geçerek Form load prosedürü altında aşağıdaki kodu yazınız.



Program çalıştırdığınızda liste kutuların arasındaki farkı inceleyiniz.

MultiSelect Bu özellik bir liste kutusu içinde birden fazla eleman seçme imkanı verir.

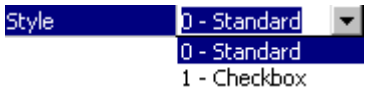


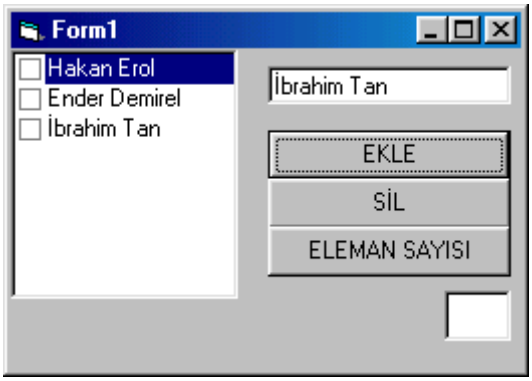
Standart olarak 0- None seçilidir. Listedeki elemanlardan sadece bir tane seçilebilir. 1-Simple olması halinde mouse ile tıklanan her öğe seçilecek, seçili öğe tıklandığında seçilmişliği kaldırılacaktır. 2-Extended seçilmesi halinde **Ctrl** ve **Shift** tuşları yardımıyla çoklu seçim yapmak mümkün olacaktır.

Text Liste içerisinde seçili elemanını içeriğini verir. TextBox nesnesindeki gibi değiştirilebilir bir özellik değildir. Sadece okunabilir bir özelliktir.

Sorted Eğer **True** seçilirse Liste alfabetik olarak sıralanarak gösterilir. **False** seçili ise Liste elemanların eklenme sırasına göre listelenir.

Style Eğer 1 – CheckBox seçilirse liste kutusundaki elemanlar birer CheckBox gibi görüntülenir ve bu kutucuklar işaretlenerek seçilebilir.



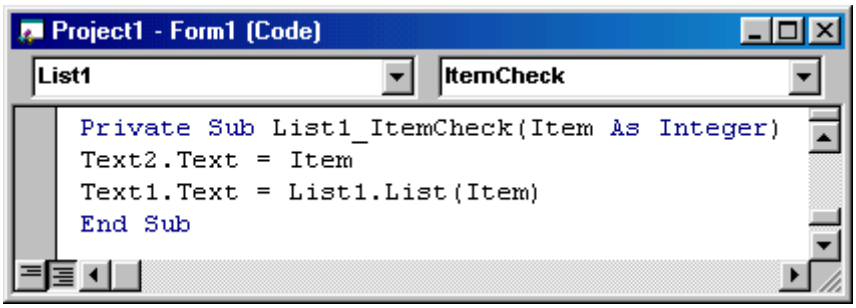


Events (Olaylar)

Click() Listeden bir eleman seçmek için listeye tıklandığında bu olay meydana gelir. Diğer kontrollerdeki gibi çalışır.

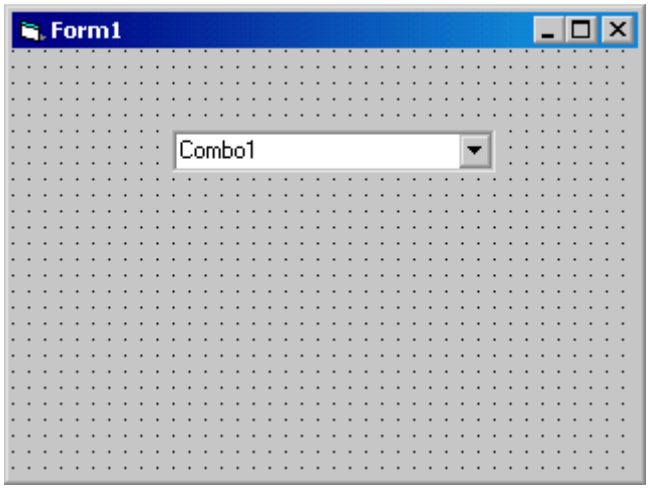
***ItemCheck(Item** As Integer)*

ListBox kontrolünün Style özelliği 1 – CheckBox olarak değiştirilmiş ise listeden bir eleman seçmek için check kutusu işaretlendiğinde bu olay meydana gelir. Bu prosedürde program tarafından **Item** sayısı seçilen elemanın index numarasını verir.



ComboBox (Açılan Liste)

Aşağı doğru açılan bir liste kontrolüdür. Değerleri daha önceden belli olan elemanların görüntülenmesi için kullanılır. Bu liste kontrolünde ekranda sadece seçili olan eleman görünür. Form üzerine ComboBox eklemek için bileşen paletinden ComboBox nesnesi seçili duruma getirilir ve formun üzerinde istenilen yere yerleştirilir. Bu işlemin ekran görüntüsü aşağıda verilmiştir.



Properties (Özellikler)

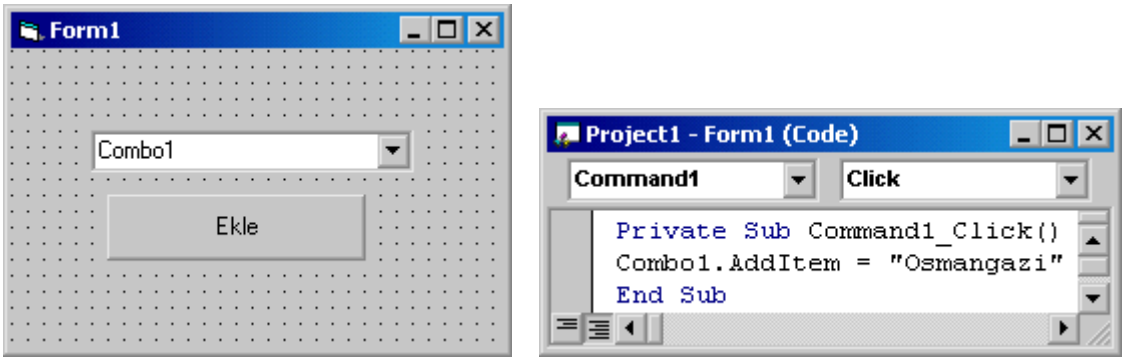
Style 0,1,2 değerlerini alan bu özellik ComboBox' ın tipini belirler.

2: Aşağı doğru açılabilen ve kullanıcı tarafından giriş yapılamayan tipi oluşturur. Kullanıcı listedeki değerlerden birini seçebilir fakat içeriğini değiştiremez.

1: Aşağı doğru açılmayan ve içeriği kullanıcı tarafından değiştirilebilen tipi oluşturur. Listedeki elemanlardan herhangi birini değiştirebilmek için klavyeden aşağı ve yukarı tuşları kullanılır.

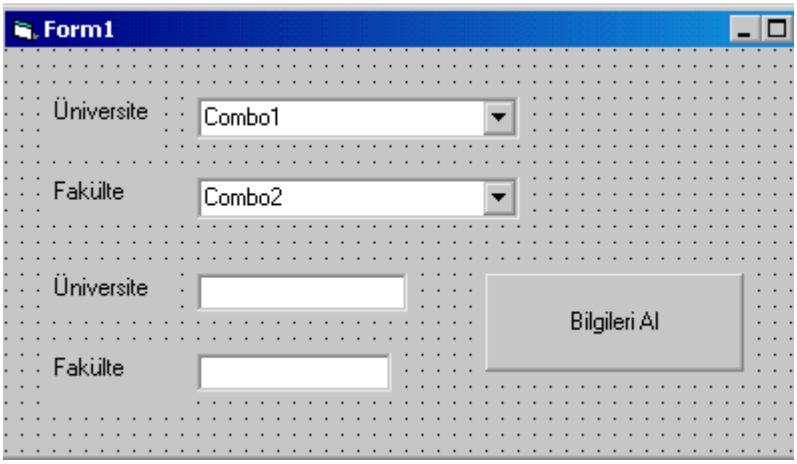
0: Aşağı doğru açılan ve içeriği kullanıcı tarafından değiştirilebilen tipi oluşturur.

İçeriği kullanıcı tarafından değiştirilebilen ComboBox' larda kullanıcının girdiği değerler listede görüntülenmez. Bu değerlerin görülmesi için aşağıdaki gibi kod yazılması gerekir. Kod penceresinde de görüldüğü gibi bu iş için AddItem metodu kullanılmıştır. Aşağıda verilen örneği hazırlayınız ve çalıştırınız.

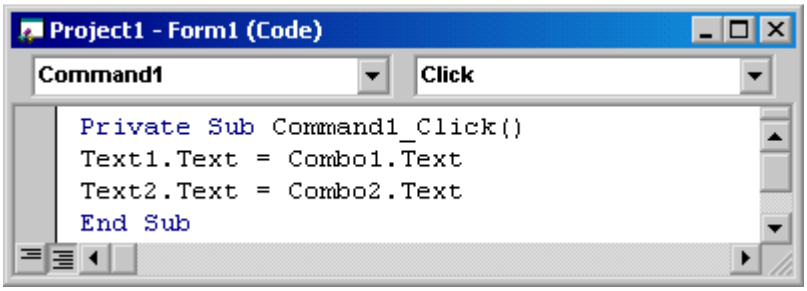


Text

ComboBox' ın kutusundaki aktif olan yazıyı belirtir. Eğer ComboBox stili 0 veya 2 ise bu yazı listedeki elemanlardan biri olmayabilir. Bu özellik kullanıcının kutuya girdiği metni öğrenmek için kullanılır. Örnek olarak öğrencinin kişisel bilgilerinin alındığı bir Form penceresini aşağıdaki gibi tasarlayalım.



Üniversite ve Fakülte isimlerinin olduğu ComboBox' lara bildiğiniz üniversite ve fakülte adlarını yazınız. Bilgileri Al düğmesinin Click() olayına da aşağıdaki kodu yazıp çalıştırınız. Burada Combobox' lardan öğrencinin, hangi üniversiteyi ve fakülteyi seçtiği aşağıdaki Text kutularında gösterilecek. Ancak bunun için program çalıştırıldığında kullanıcının ComboBox' lardan herhangi bir seçeneği tıklaması gerekir.

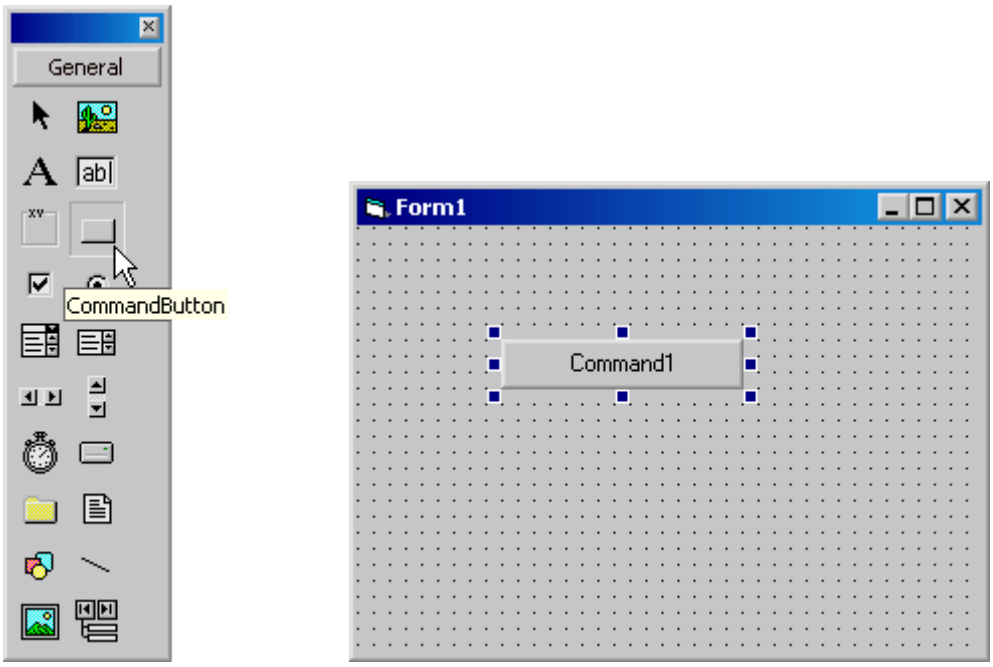


CommandButton

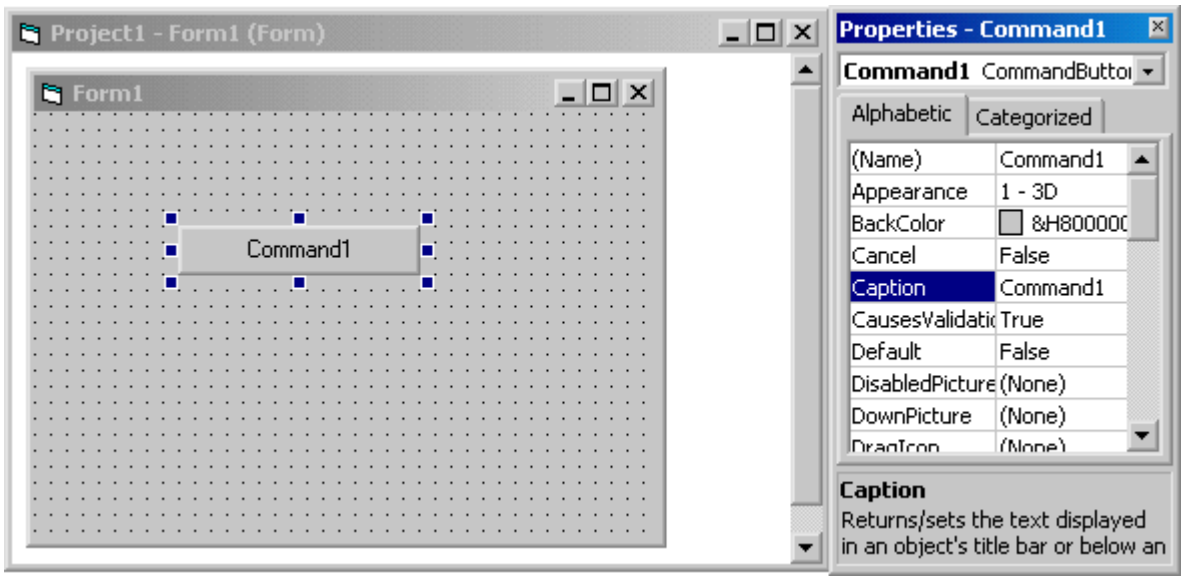
Component (bileşen) araç çubuğundan seçilerek form üzerine yerleştirilen CommandButton nesnesini Windows uyumlu pek çok uygulamada görmek mümkündür ve bir Windows uygulaması için vazgeçilmez bir nesnedir. CommandButton nesnesinin bir çok özelliği daha önce anlatılmış olan Label ve TextBox gibi nesnelerin özelliklerine benzediği için burada bazı özellikler anlatılmamıştır.

Form üzerine CommandButton nesnesi eklemek için Component araç çubuğunda CommandButton nesnesini gösteren düğme seçili duruma getirilir. Aşağıdaki ekran görüntüsü ilgili düğme seçili duruma getirildikten sonra alınmıştır.

Yanda gösterilen araç çubuğunda CommandButton düğmesi seçili duruma getirildikten sonra fare işareti form üzerinde düğme eklenmek istenen yerine tıklanır ve düğmenin büyüklüğü ayarlanarak form üzerine yerleştirilir. Aşağıdaki ekran görüntüsü düğme eklendikten sonra alınmıştır.



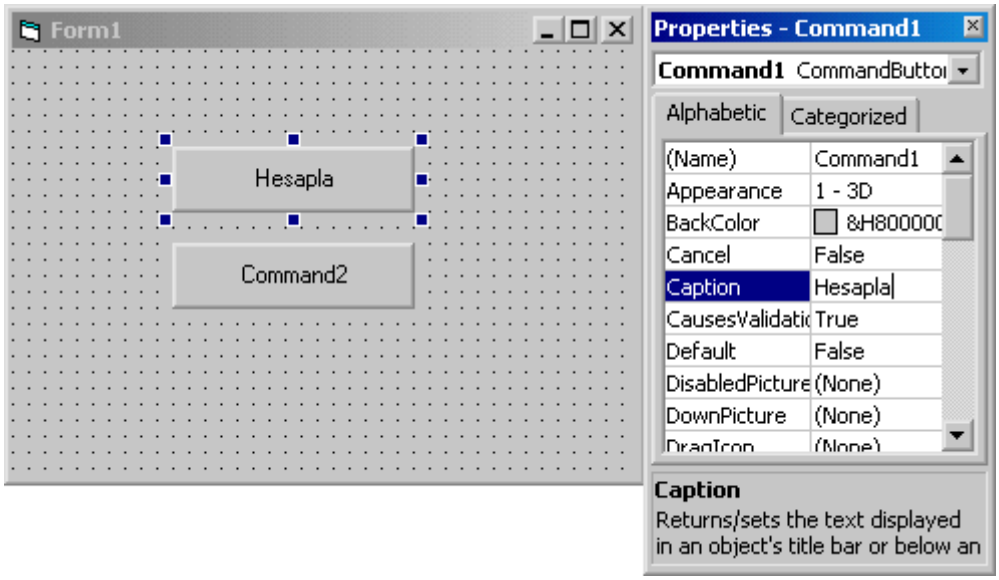
CommandButton nesnesi form üzerine yerleştirildikten sonra sıra bu nesnenin özelliklerini değiştirmeye gelir. Nesnenin özelliklerini değiştirmek için CommandButton nesnesi fare ile bir kere tıklanır ve aktif hale getirilir. Aşağıda verilen ekran görüntüsünde görüldüğü gibi ilgili nesnenin özellikleri Properties paletinde gösterilmiştir.



Properties (Özellikler)

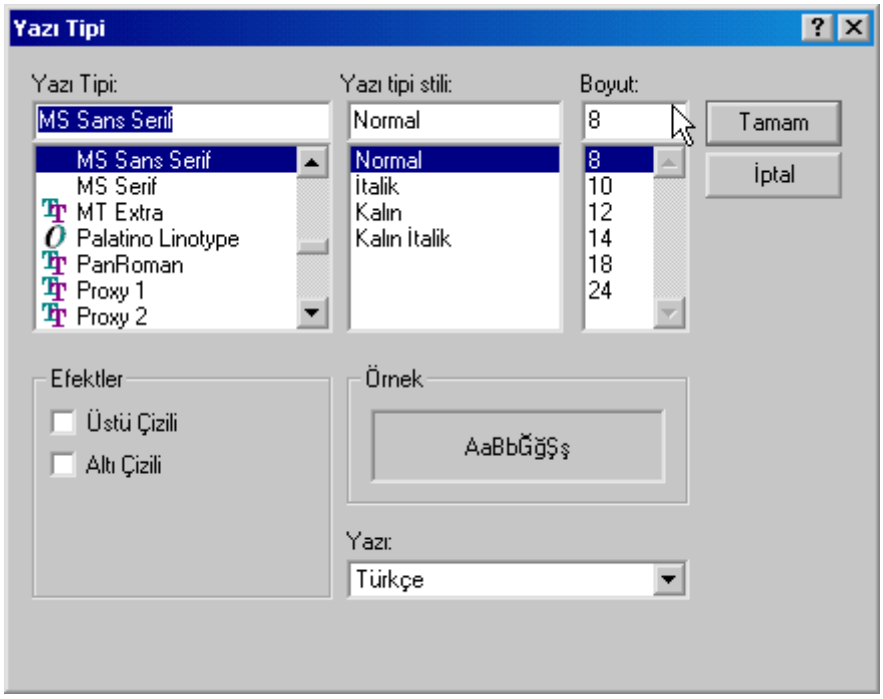
Caption

Düğmenin başlığını değiştirmek için Properties paletinden Caption özelliği değiştirilir. Aşağıda bu işlemin ekran görüntüsü verilmiştir.



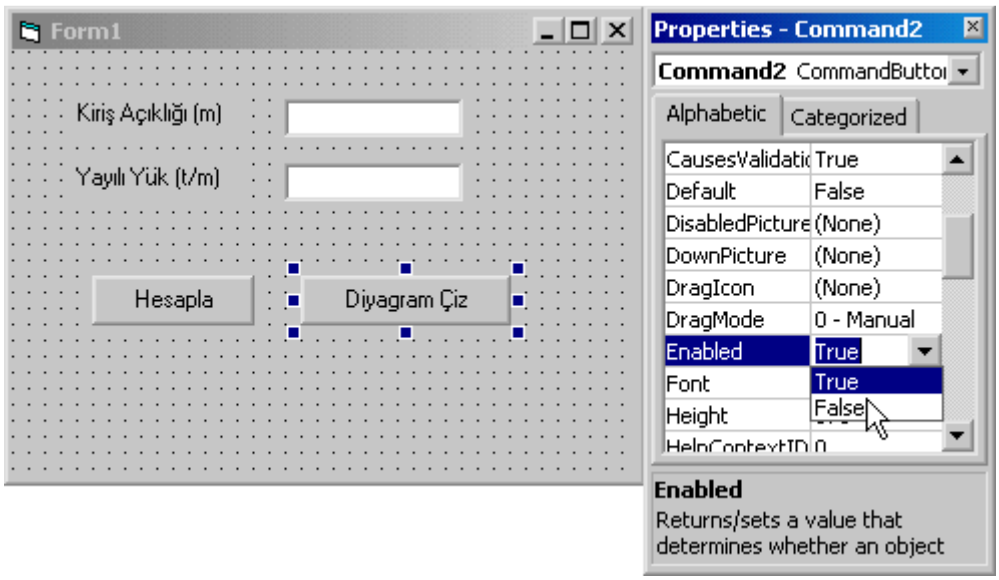
Font

Düğme üzerine yazılan metnin fontunu değiştirmek için yine nesne aktif hale getirilerek Properties paletinden Font özelliği seçilir. Bu özellik seçildikten sonra ekrana yazı tipi ile ilgili bir pencere gelir ve bu pencereden yazı ile ilgili özellikler değiştirilir. Yazı tipi için ekrana gelen pencerenin ekran görüntüsü aşağıda verilmiştir.



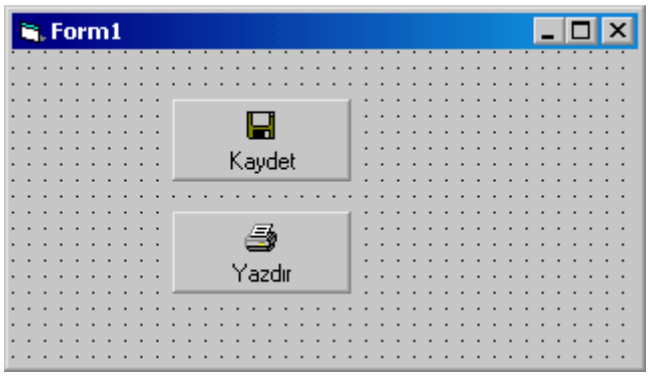
Enabled

Hazırlanan programın amacına göre programcı isterse bazı nesneleri programın belli aşamalarında kullanılamaz hale getirebilir. Örnek olarak bir kirişin üzerindeki moment değerleri hesaplanmak isteniyor ve hesaplama sonucunda kiriş üzerindeki moment diyagramı çizilmek isteniyor. Hesaplama yapılmadan diyagram çizilemeyeceği için başlangıçta Diyagram Çiz düğmesi kullanılamaz olmalıdır. Bu işlem düğmenin Enabled özelliği False seçilerek yapılır. Düğmenin Enabled özelliği True ise düğme kullanılabilir, False ise düğme kullanılamaz olmaktadır. Bu işlemin ekran görüntüsü aşağıda verilmiştir.



Picture

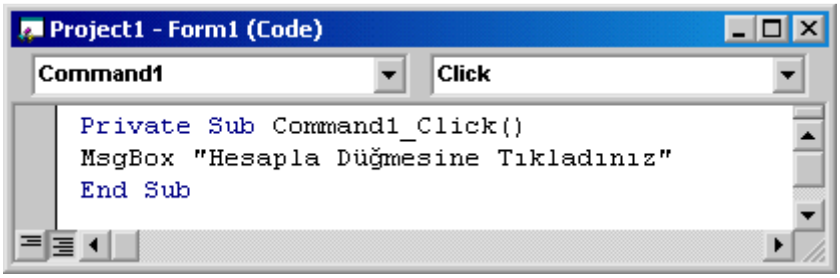
CommandButton üzerinde resim görüntülemek için kullanılan bir özelliktir. Örnek olarak kayıt işlemini yapan bir düğme üzerinde disket resmi ya da yazdırma işlemini yapan bir düğme üzerinde yazıcı resmi görüntülenebilir. Bu özelliği kullanmak için CommandButton nesnesinin özelliklerinden Picture özelliği seçilir ve ekrana gelen pencerede görüntülenmek istenen resmin bilgisayardaki yeri belirlenir. Ardından CommandButton nesnesinin Style özelliği Graphical olarak seçilir. Aşağıdaki ekran görüntüsü bu işlemler yapıldıktan sonra alınmıştır. Aşağıdaki gibi üzerinde resim bulunan düğmeler hazırlayınız.



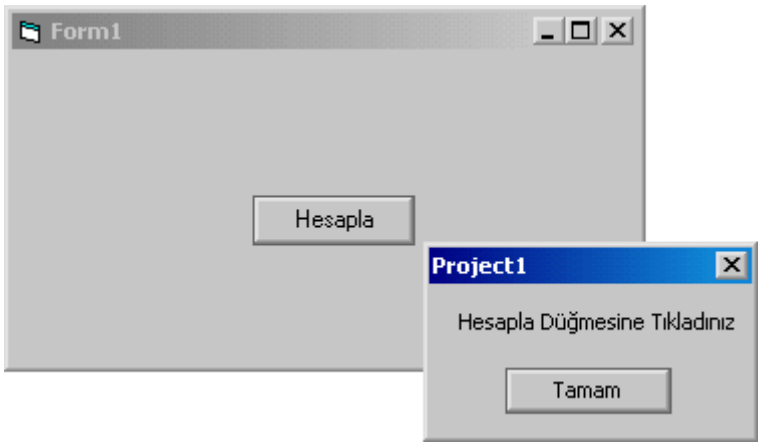
Events (Olaylar)

Click ()

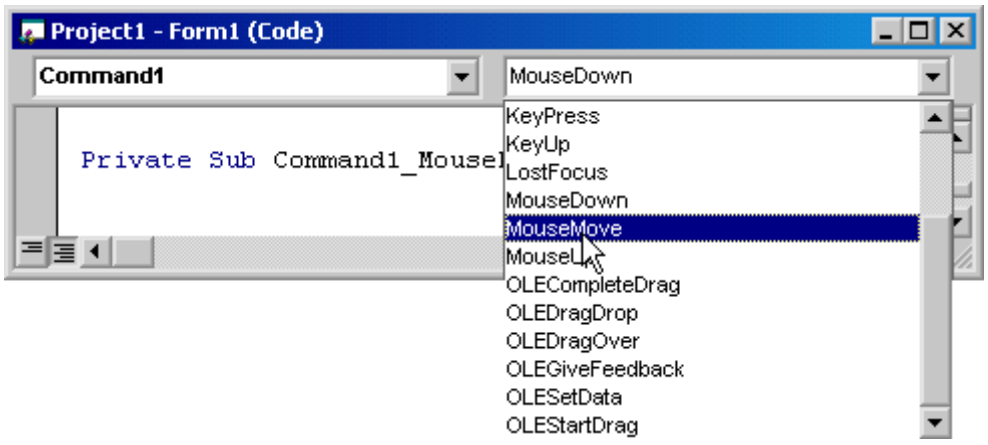
CommandButton nesnesi için en çok kullanılan olaylardan biri Click() olayıdır. Kullanıcı CommandButton üzerine tıkladığında ya da düğme aktifken Enter tuşuna bastığında Click() olayı meydana gelir. Tasarım aşamasındayken CommandButton nesnesi üzerine çift tıklanarak ekrana gelen kod penceresinde bu olay ile ilgili kod yazılır. Örnek olarak kullanıcı CommandButton üzerine tıkladıktan sonra ekrana bir mesaj yazılsın. Olayın kod penceresinin ekran görüntüsü aşağıda verilmiştir. Aşağıdaki programı hazırlayınız.



Program çalıştırıldığında kullanıcı Hesapla düğmesine tıkladığı zaman ekrana “Hesapla Düğmesine Tıkladınız” yazılı bir mesaj penceresi gelir. Aşağıdaki ekran görüntüsü bu olaydan sonra alınmıştır.

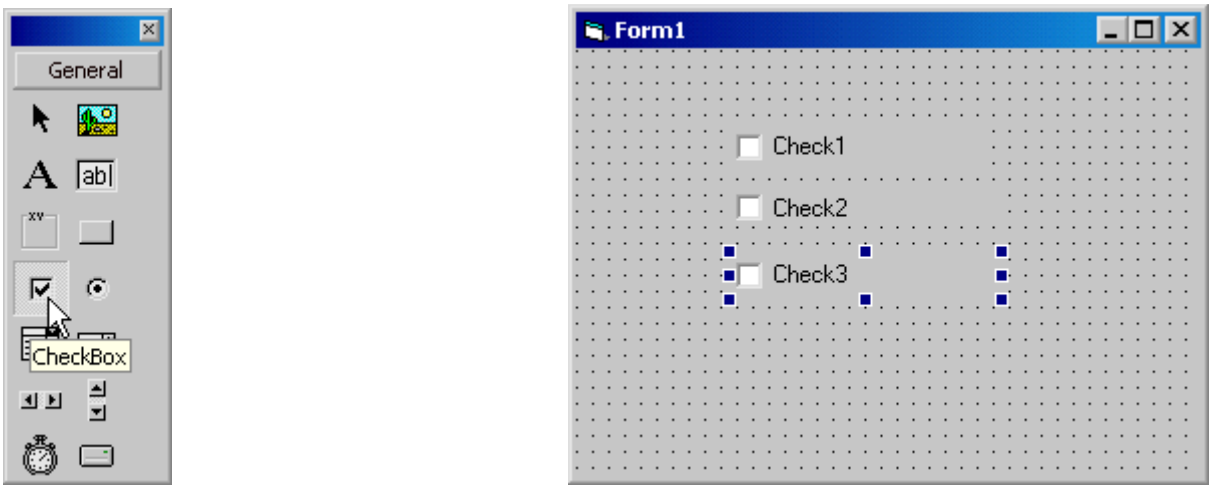


CommandButton nesnesi ile ilgili bütün olaylar kod penceresinin sağ üst köşesindeki menüden görülebilir. Buradan istenen olay seçilerek olayla ilgili kod yukarıda anlatıldığı gibi yazılabilir.

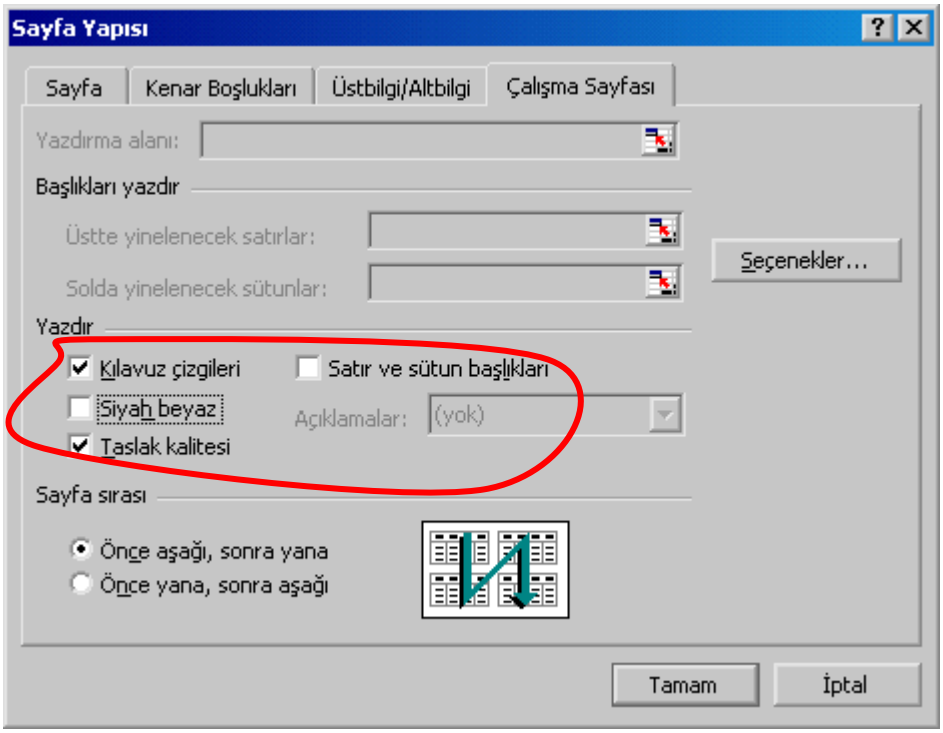


CheckBox (Onay Kutusu)

Bir Windows uygulamasında en çok kullanılan bileşenlerden biridir. Kullanıcının belirli özellikleri aktif ya da pasif hale getirmesi için kullanılan bir kontroldür. Form üzerine CheckBox eklemek için Component araç çubuğundan CheckBox nesnesi seçili duruma getirilir ve Form üzerinde istenilen yere boyutları da ayarlanarak yerleştirilir. Aşağıda bu işlemlerin ekran görüntüleri verilmiştir.



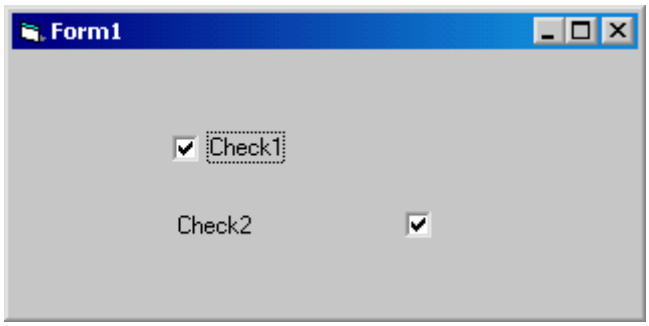
CheckBox kontrolünün kullanılması ile ilgili örnek olarak Excel programında çıktı sayfasının düzenlenmesi için aşağıda ekran görüntüsü verilebilir. Bu pencerede kullanıcı CheckBox kontrolünü kullanarak aynı anda birden fazla özelliği seçebilir.



Properties

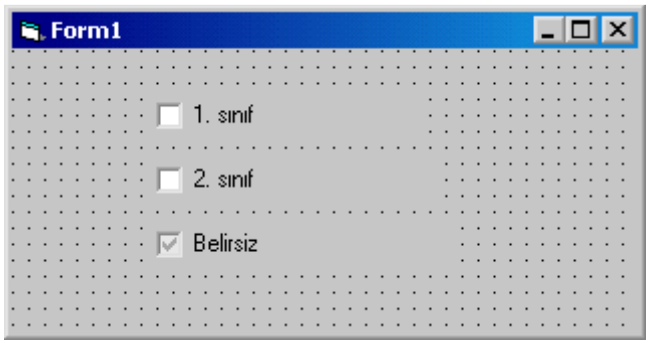
Alignment

İşaretleme kutusunun sağda ya da solda olması özelliğini değiştirir. Aşağıda bu özellik ile ilgili bir ekran görüntüsü verilmiştir. Check1 nesnesinin onay kutusu solda, Check2 nesnesinin onay kutusu sağdadır. Genelde Windows uygulamalarında onay kutusu solda seçilmektedir.



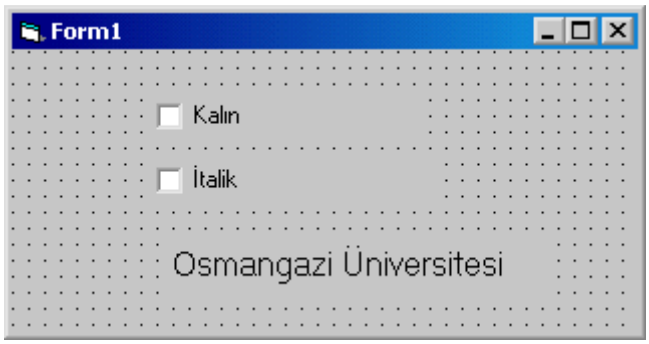
Value

Bu özellik üç değer alabilir. İşaretsiz (0), İşaretli (1) ve Belirsiz (2). İşaretsiz ve İşaretli değerleri kullanıcı tarafından üzerine tıklanarak değiştirilebilir. Belirsiz değeri sadece program tarafından atanabilir ve kullanıcı bu özelliğe tıklamayla müdahale edemez. Örnek olarak CheckBox kontrolünün bir öğrencinin aldığı derslerin 1. sınıf ya da 2. sınıf olduğunu gösterdiğini düşünelim. Öğrenci 1. ve 2. sınıftan da ders alıyorsa bu özellik belirsiz olur. Program öğrencinin aldığı dersleri yorumlar ve iki sınıftan da ders alıyorsa bu özelliği belirsiz olarak atar.

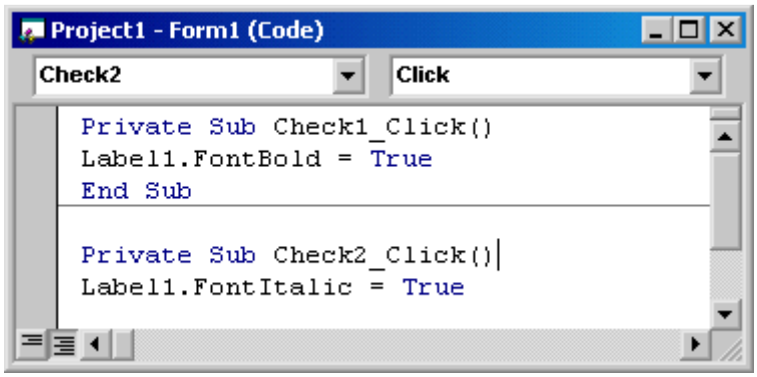


Events (Olaylar)

Diğer kontroller için kullanılan bir çok olay CheckBox kontrolü için de kullanılabilir. Ancak en çok kullanılanlardan biri Click () olayıdır. Örnek olarak Form üzerindeki bir Label' ın yazı tipini CheckBox ile değiştirelim. Bunun için tasarlanan Formun ekran görüntüsü aşağıda verilmiştir.



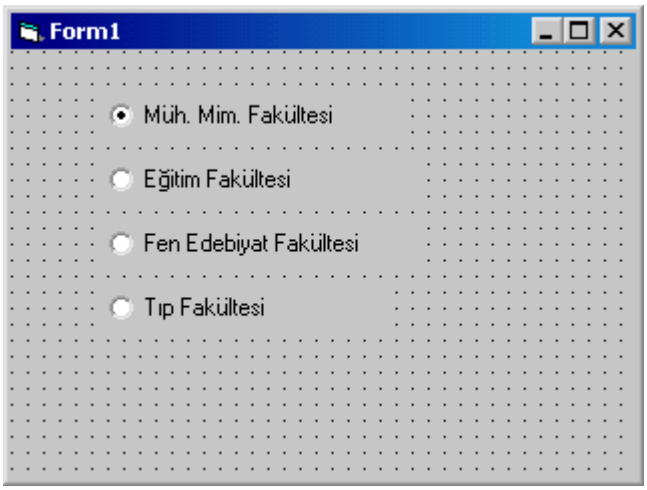
Osmangazi Üniversitesi yazılı Label' ın yazı tipi Kalın ve İtalik onay kutularına tıklama yapılarak değiştirilebilir. Olayın kod penceresi aşağıda verilmiştir. Aşağıda verilen programı hazırlayınız.



OptionButton

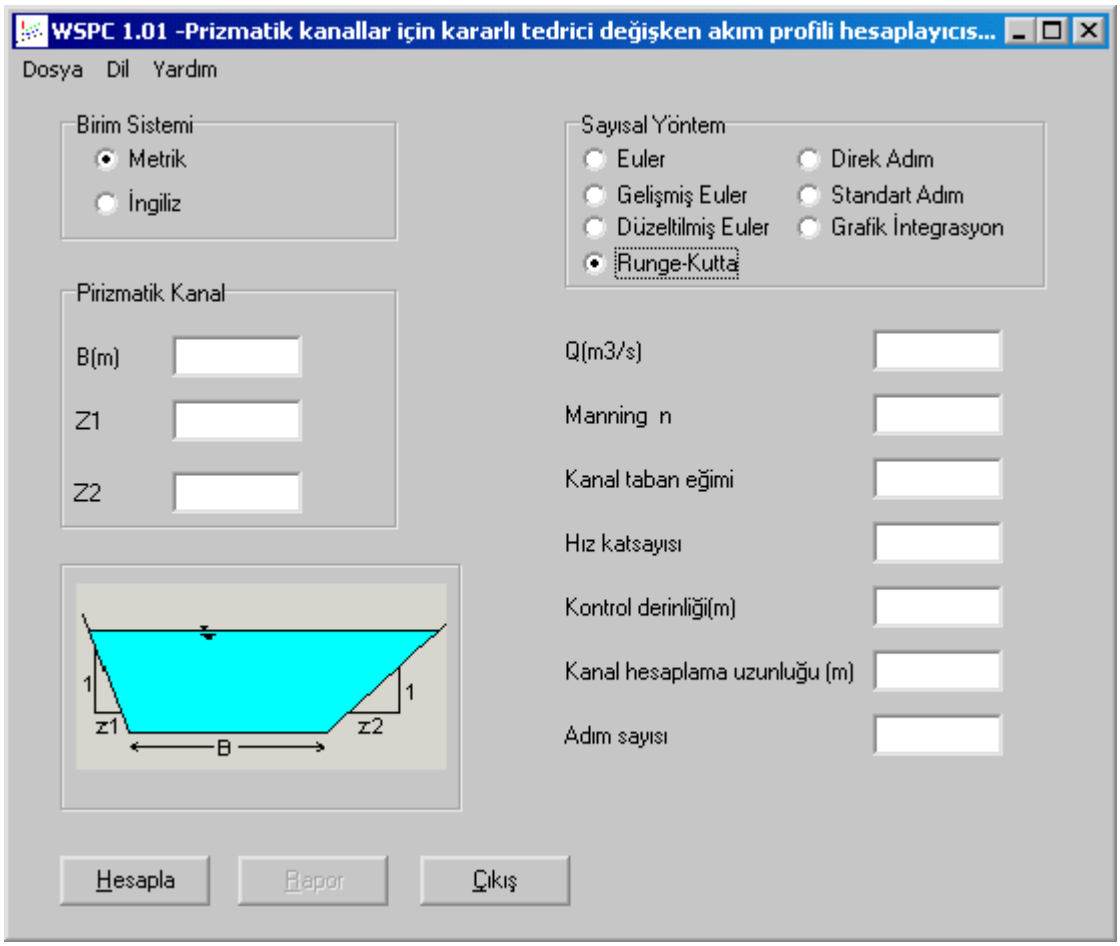
Kullanımı CheckBox kontrolüne benzer fakat farklı olarak kullanıcının birkaç seçenekten sadece birinin seçmesini sağlayan bir kontroldür. Birkaç seçenekten birinin seçilmesini öngördüğü için tek bir tanesinin kullanılması anlamsızdır. Aynı grupta en az iki adet bulunmalıdır. OptionButton kontrolünü kullanmak için Components paletinden OptionButton nesnesini gösteren düğme seçili duruma getirilir. Aşağıdaki ekran görüntüsü ilgili düğme seçili duruma getirildikten sonra alınmıştır.

Yanda gösterilen araç çubuğunda OptionButton düğmesi seçili duruma getirildikten sonra fare işareti form üzerinde düğme eklenmek istenen yerine tıklanır ve düğmenin büyüklüğü ayarlanarak form üzerine yerleştirilir. Aşağıdaki ekran görüntüsü düğme eklendikten sonra alınmıştır. Örnek olarak öğrencinin fakültesini seçmesi istenmektedir.



OptionButton kontrolünün bir çok özelliği CheckBox kontrolüne benzemektedir. Aralarındaki en önemli fark kontrolün Value özelliği True ya da False olabilir. Value özelliği True ise düğme seçilidir, False ise düğme seçili değildir. Bir Form üzerindeki OptionButton' ların sadece birinin Value özelliği True değerini alabilir. Kullanıcı tıklama yaptığı sürece Visual Basic OptionButton' ların Value özelliğini True ya da False yapar.

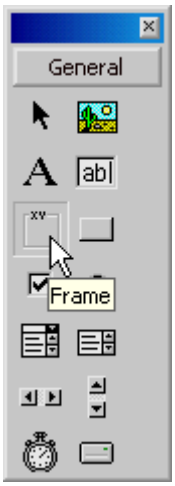
Aşağıda ekran görüntüsü verilen Windows uygulamasında program kullanıcıdan çözüm metodunu seçmesini beklemektedir. Kullanıcı OptionButton' lara tıklama yaparak çözüm metodunu seçmekte, program da kullanıcının seçtiği çözüm yöntemine göre problemi çözmektedir.



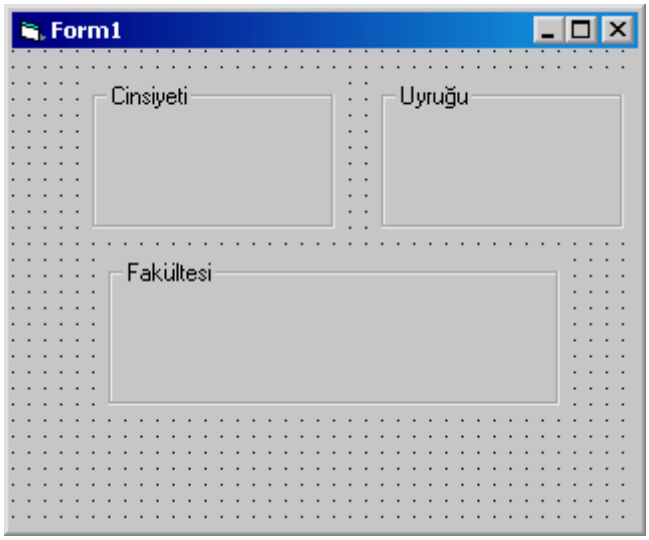
OptionButton kontrolünün bir çok özelliği CheckBox kontrolüne benzediği için bileşenin özelliklerine değinilmemiş, bu kısım okuyucuya bırakılmıştır.

Bazı Windows uygulamalarında OptionButton' lar Frame' ler ile birlikte kullanılır. Örnek olarak programınızda öğrencinin kişisel bilgilerinizi öğrenmek istiyorsunuz ve bu özellikler içinde öğrencinin cinsiyeti, uyruğu ve öğrencisi olduğu Fakülte var. Bunların hepsini tek bir Form üzerine yerleştirirsek kullanıcı bunlardan sadece birini seçebilir. Bunu önlemek için OptionButton' ları gruplandırmak gerekir ve bunun için Frame' lerden yararlanılır.

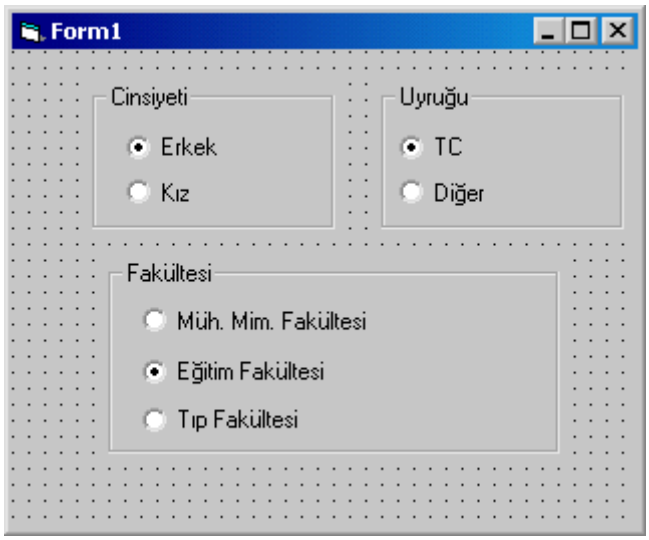
Frame kontrolünü kullanmak için Components paletinden Frame nesnesini gösteren düğme seçili duruma getirilir. Aşağıdaki ekran görüntüsü ilgili düğme seçili duruma getirildikten sonra alınmıştır.



Yukarıda diğer kontrollerde bahsedildiği gibi kontrol formun üzerine istenildiği gibi yerleştirilebilir. Aşağıda bir öğrencinin kişisel bilgilerinin alındığı bir Form gösterilmiştir.



Form üzerine üç tane Frame (çerçeve) yerleştirilmiş ve çerçevelerin boyutları ve başlıkları Properties çubuğundan amaca uygun şekilde ayarlanmıştır. Şimdi sıra OptionButton kontrollerinin Frame' ler üzerine yerleştirilmesine geldi. Yukarıda anlatıldığı gibi Form üzerine yerleştirme sırası takip edilerek Frame' ler üzerine yerleştirilir. Aşağıdaki ekran görüntüsü bu Frame' ler üzerine OptionButton' lar yerleştirildikten sonra alınmıştır. Aşağıdaki formu hazırlayınız.



Artık kullanıcı her bir Frame üzerinde sadece bir seçeneği işaretleyebilmekte ve OptionButton' ları istediğimiz gibi gruplandırmış olduk. Frame' ler genel olarak nesneleri gruplandırmak için kullanılır.

PictureBox

Form üzerinde resim görüntülemek için kullanılır. Ayrıca bazı metotlar kullanılarak içine çizimlerde yapılabilmektedir.

Proporties

Picture

Bu özellik kontrol içerisinde görüntülenecek resmi belirlemek için kullanılır. Nesne içerisinde resimler şu şekilde yüklenebilir.

LoadPicture ile dosyadan resim yüklenebilir. Bunun için dosyanın bilgisayardaki yerinin belirtilmesi gerekir. Örnek olarak bilgisayarın Windows klasörü içerisindeki okul.bmp dosyasını yüklemek için,

Picture1.Picture=LoadPicture("\windows\okul.bmp")

Form üzerindeki başka bir PictureBox nesnesinin Picture özelliği kullanılarak resim yüklenebilir. Bunun için Form üzerindeki Picture2 nesnesinin içindeki resmi Picture1 nesnesine yükleyelim. Bunun için aşağıdaki kodu yazmamız gerekir.

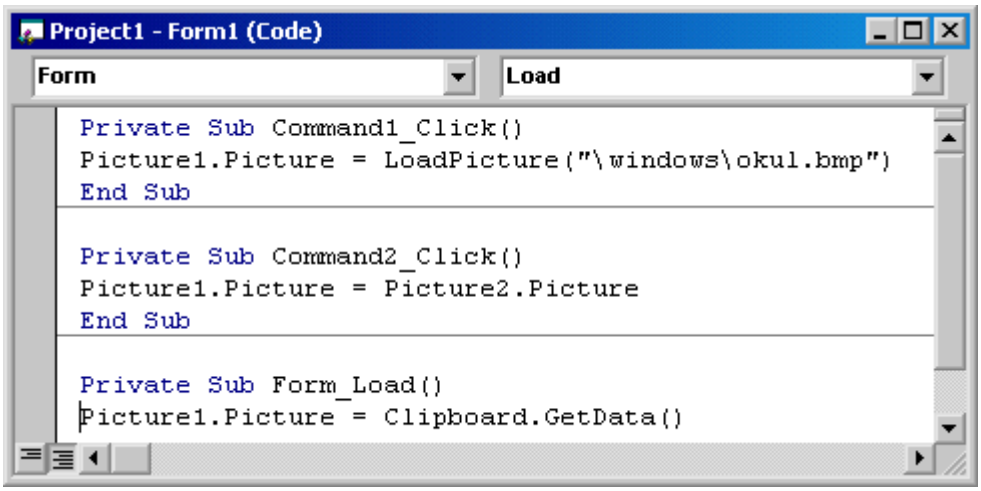
Picture1.Picture=Picture2.Picture

PictureBox içerisine resim yüklemek için kullanılan diğer bir metot da; herhangi bir Windows uygulamasından kesilerek ya da kopyalanarak Clipboard' a yüklenen resmi PictureBox içerisinde görüntülemektir.

Picture1.Picture=Clipboard.Getdata();

Bu işlemlerin yapılması için aşağıda bir Form ve kod penceresi hazırlanmıştır. Form üzerinde iki adet PictureBox nesnesi ve üç adet CommandButton nesnesi yerleştirilmiştir.

Aşağıdaki resim kutusunda kullandığınız bilgisayardaki herhangi bir resmi yükleyin ve sağdaki düğmeler için aşağıdaki kod penceresinde verilen kodları yazın. Clipboard' dan yükle düğmesinin çalışması için Windows' un Paint uygulamasından herhangi bir resmi kopyalayın ve sonra programınızı çalıştırın.



AutoRedraw

Bu özelliğin aldığı True ya da False değerleri ile nesnenin kendini otomatik olarak yenilemesi ya da yenilememesi sağlanır. Bu özellik sayesinde arka plana düşmüş ya da minimize olmuş bir form veya üzeri kapatılmış bir nesnenin üzerinin açılması ile nesnenin kendisini yeniden çizdirmesi sağlanır. Picture kutusunda metotlar kullanılarak yapılmış yazım ve çizimler varsa PictureBox' un kendini yenileyebilmesi için bu özellik True olmalıdır.

Shape & Line

Bu kontrol elemanları Form üzerinde dikdörtgen, kare, elips, çember, oval kare, oval dikdörtgen ve çizgi çizmek için kullanılır.

Proporties

BorderStyle

Bu özellik nesnenin çerçeve biçimini belirler. Çerçeve biçimleri ve BorderStyle özelliğinin aldığı değerler aşağıdaki tabloda gösterilmiştir.

BorderStyle	Çerçeve Biçimi
0	Zeminin rengiyle uyumlu, görülmez
1	Solid (Tam çerçeve)
2	Dash (Çizgi)
3	Dot (Nokta)
4	Dash dot (Çizgi, nokta)
5	Dash dot dot (Çizgi, nokta, nokta)
6	Inside solid (Şekil ile çerçeve kenarları çakışık şekilde)

BorderWidth

Bu özellik kontrol elemanlarının çerçeve kalınlığını gösterir. 1 ile 8192 arasında değerler alabilir. Borderstyle' ın sadece solid ve inside solid biçimlerindeyken etkisi görülür.

Shape

Shape kontrol elemanının çizeceği şekli belirler ve 0 ile 5 arasında değerler alır.

Shape	Şekil	Shape	Şekil
0	Dikdörtgen	3	Çember
1	Kare	4	Oval Dikdörtgen
2	Elips	5	Oval Kare

FillColor

Şeklin içinin boyanacağı rengi belirler.

Line

Bu kontrol elemanı form üzerine bir çizgi şeklinde alınır. Alınan bu çizgiyi yatay, dikey ve eğik bir biçimde göstermek, istenilen boyuta getirmek mümkündür.

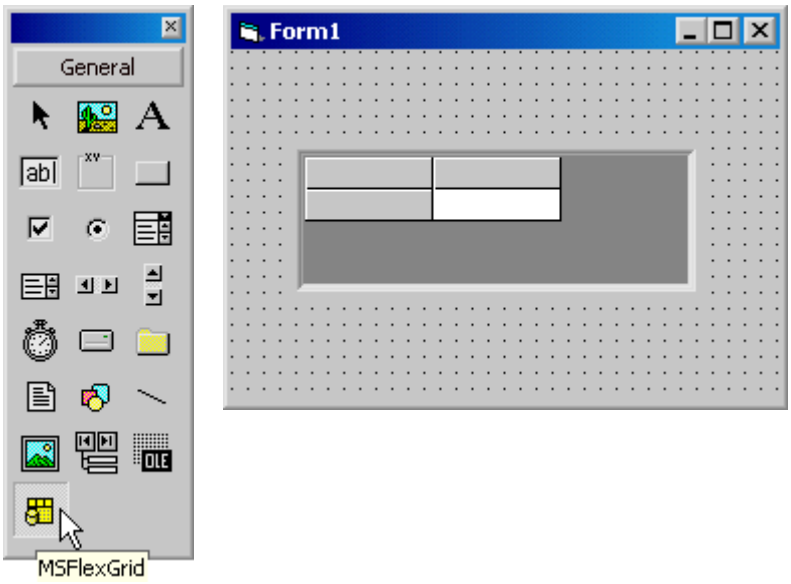
Properties

X1,X2,Y1,Y2

X1 ve Y1 noktaları çizgi kontrol elemanının başlangıç koordinatlarını, X2 ve Y2 ise bitiş koordinatlarını belirler. Yatay kontrolleri X1 ve X2, Dikey kontrolleri ise Y1 ve Y2 belirler.

MSFlexGrid

StandartExe uygulamasında bulunmayan bir kontroldür. Bu kontrolü kontrol penceresinde görebilmek için EnterpriseEdition seçeneği seçilir ve diğer uygulamalarda olduğu gibi Form üzerine yerleştirilir. Aşağıda bu işlemin ekran görüntüleri verilmiştir.

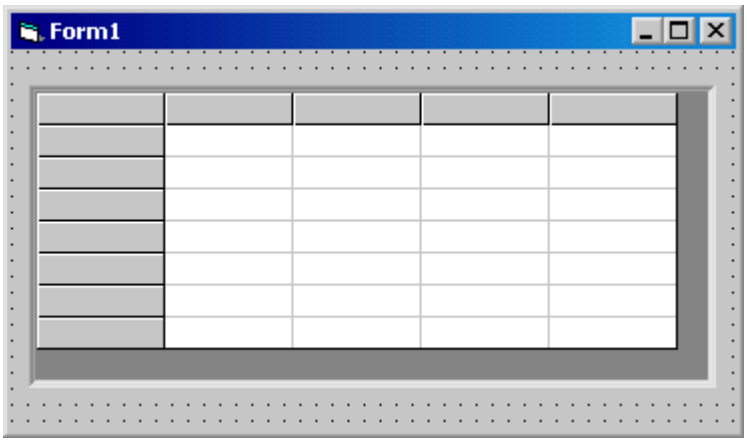


Bu ızgara kontrolü hücrelerden meydana gelir ve Excel' deki sayfalara benzer bir görüntüsü vardır.

Properties

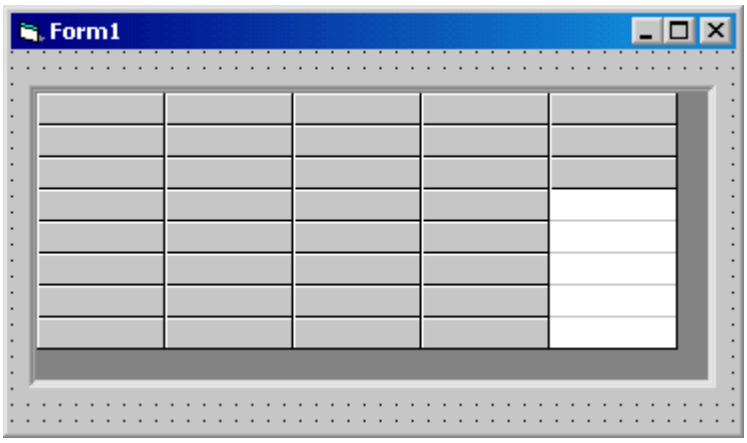
Cols, Rows

Bu özellik ızgarada bulunması gereken satır ve sütun sayılarını gösterir. Bunun minimum değeri 1 dir. Bir ızgarada en fazla 2000 satır ve 400 sütun oluşturulabilir. Aşağıda 5 sütundan ve 8 satırdan oluşan bir ızgara elemanı verilmiştir.



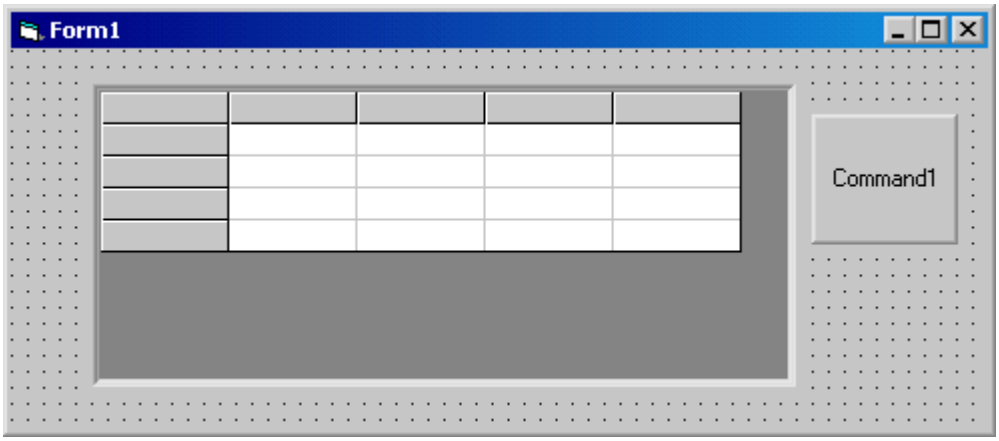
FixedCols, FixedRows

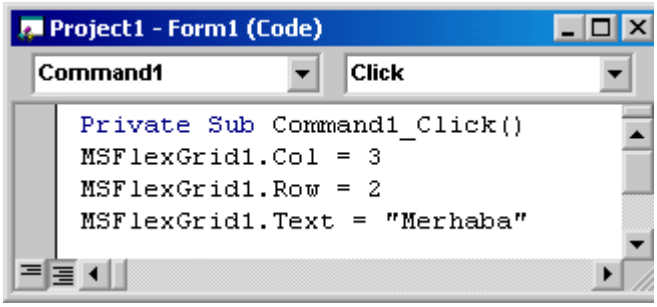
Tablo üzerinde kullanıcının erişemediği ve sütun ve satırların başlığı olarak kullanılan sütun ve satır sayısını belirler. Normalde her ikisinin de değeri 1 dir. Bu tablo üzerinde gördüğümüz gri renkli kısımdır. Aşağıda FixedRows değeri 3, FixedCols değeri 4 olan bir grid elemanı verilmiştir.



Col,Row

Bir ızgarada üzerinde işlem yapılacak hücre adreslerini belirtir. İlk satır için Row=0, ilk sütun için Col=0' dır. Bu iki değer aktif hücreyi belirler. Aşağıdaki örnekte 5 satır, 5 sütundan oluşan bir ızgara hazırlanmış ve sağdaki düğmeye tıklandığında 2. satır, 3. sütuna "Merhaba" yazılmaktadır. Olayın Form ve kod penceresi aşağıda verilmiştir. Sizde aynı örneği hazırlayıp çalıştırınız.





Text

Aktif hücrenin içindeki metni temsil eder. Yukarıda verilen örnekte Text özelliği kullanılmıştır.

CellSelected

Bu özellik Col ve Row özellikleri ile belirlenen hücrenin seçili olup olmadığını belirtir.

ColAlignment

Numarası verilen kolondaki hücrelerin içeriğini sola, sağa ve ortaya yerleştirir. 0 ile 2 arasında değer alabilir.

0: Sola dayalı

1: Sağa dayalı

2: Ortalanmış

ColWidth

Numarası verilen kolonun genişliğini ayarlamak için kullanılır. Atanan değerin birimi twip cinsinden olmalıdır.

RowHeight

Numarası verilen satırın yüksekliğini ayarlamak için kullanılır. Atanan değerin birimi twip cinsinden olmalıdır.

GridLines

False değeri verilerek hücreler arasındaki ayırma çizgileri görünmez yapılabilir.

Picture

Bu özellik diğer nesnelerin Picture özelliği gibidir. Grid nesnesinin her bir hücresine bu özellikle resim eklenebilir.