

Herkes İçin
Visual Basic

Visual Basic'in Tüm Sürümleri İçin Komple
Kendi Kendine Öğrenim ve Başvuru Kaynağı

Herkes İçin
Visual Basic

Bu kitap hakkında:

Bu kitap, Herkes için Visual Basic, okuyucu ile kitap arasında karşılıklı etkileşim yapısı içinde, kendi kendine öğrenim türünden, bir seri öğretici derslerden oluşmaktadır. Tam ve eksiksiz bir Visual Basic Programlama Dili ve Ortamı imkanı sağlamaktadır.

Bu öğretmen kitap, 10 ayrı dersten (kısmıdan) oluşmuştur (basit, takip etmesi kolay ifade ve terimler kullanılarak). Visual Basic Uygulama- larınızı nasıl yapılandıracağınızı size öğretecektir. Visual Basic Uygulama- larınızı yapılandırırken takip edeceğiniz yol ve yöntemlerin her adımını size göstermek için, kitap içinde çok sayıda örnek kullanılmıştır.

Bu kitap, aynı zamanda, okuyucularının yapılandırması ve denemesi için detaylı bilgisayar projelerini de (uygulama programları) sunmaktadır. Herkes için Visual Basic, yani ders notlarından oluşan bu kitapla birlikte, bir çok Visual Basic örnekleri ve uygulama projeleri (gerek bu kitapta işlenen, gerekse de diğer başka bazı örnekler) bu kitapla birlikte verilen bir CD ile size sunulmaktadır.

Kurs Ön Şartları:

Herkes için Visual Basic'i kullanmak için, Windows ortamını az çok biliyor olmalısınız. Yani, dosyaları nasıl bulacağınızı, pencereleri nasıl hareket ettireceğinizi, pencereleri nasıl yeniden boyutlandıracağınızı v.s. biliyor olmalısınız.

Hiçbir programlama tecrübesi gerekmemektedir. Derslerde işlenen malzeme herkes tarafından anlaşılabilir.

Son olarak, bir Microsoft Visual Basic Programı'na ihtiyacınız var. Bu öğretmen kitap içindeki konular; Visual Basic 4.0 (32 bit sürümü), Visual Basic 5.0, veya Visual Basic 6.0 sürümlerinin tümü için uygundur. Zira, kitap içerisinde işlenen konu ve örneklerin, bu sürümlerin her birisinde nasıl yapıldığı, teker teker ve ayrı ayrı anlatılmıştır.

Bu arada, Visual Basic 4.0 (32 bit sürümü), Visual Basic 5.0, veya Visual Basic 6.0 sürümleri, hangi Windows sürümü üzerinde çalışıyorsa (Windows 3.1, Windows 95 veya Windows 98), bunların da, yani Windows'un sürümü konusunda da, kitabımızın, uygulamalarımızın ve örneklerimizin, bir sınırlaması yok !

Bazı başlıklarda özel durumlar için detaylı bilgilere ihtiyaç duyduğu- nuzda Visual Basic'inizin sürümüne göre, on-line help system (çevrimiçi yardım sistemi)'nden yararlanabilirsiniz.

Eğer, henüz, lisanslı bir Visual Basic Programı almadıysanız veya daha sonra almayı düşünüyorsanız, Internet üzerinden, Microsoft Web Sitesi'nde bulunan VBCCE (Visual Basic 5 - Control Creation Edition) sürümünü bilgisayarınıza yükleyebilirsiniz.

Bu, size, virüs riski taşıyan, eksik gedik, tam çalışmayan, oradan buradan bulduğunuz Visual Basic Programları yerine, sağlam, eksiksiz ve virüs riski taşımayan bir Visual Basic edinme imkanı sağlar. Fakat burada belirtmek gerekir, Visual Basic öğrenmeniz ve uygulamalar yazmanız için ihtiyacınızı karşılayan bu sürüm ile yazdığınız uygulamaları .exe dosyaları haline çeviremezsiniz. Çünkü .exe dosyası haline dönüştürülmüş bir uygulama veya program dosya ya da dosyaları, artık ticaridir ve Visual Basic olmadan da çalışabilir, yani ticareti yapılabilir. İşte, ticareti yapılacak kadar iyi uygulamalar (programlar) yazmaya başladığınızda zaten artık lisanslı bir Visual Basic alma zamanınız gelmiş demektir.

Yine belirtmekte fayda var, yukarıda sözü edilen sürümün 'Help' yani yardımı yoktur. Fakat bu kitaptaki uygulamalar da dahil olmak üzere her türlü uygulamayı geliştirebilir, çalıştırabilir ve Visual Basic'i bir eksiksiz bu sürüm üzerinde öğrenebilirsiniz.

Kursu Nasıl Gerçekleştireceksiniz:

Bu kitapta, dersler süresince işlenen ve geliştirilen projeleri (uygulamalar-programlar), bu kitapla birlikte verilen CD içinde bulunan Projeler dizini ve buna bağlı VB4Projeleri, VB5Projeleri ve VB6Projeleri alt dizinleri içinden bulabilirsiniz. Bu dizinlerde bulunan projeler aynı projeler olup, sadece farklı Visual Basic sürümleri ile hazırlanmıştır. Sizin Visual Basic sürümünüz ne ise, o alt dizindeki projeleri kullanın. Örneğin, Visual Basic 5.0 kullanıyorsanız VB5Projeleri alt dizini içindeki projeleri kullanın.

*Herkes için Visual Basic, kendi kendine öğrenim türü bir kurstur. Tavsiyemiz, her ders (kitabın her kısmı) için bir hafta ayırarak, kursu toplamda 10 haftada tamamlamanız !
Her hafta-her bir dersin kavramlarını iyice anlamanız için 4 ila 8 saat kadar bir zaman gerekecektir. Dersinize başlamadan evvel, ilgili dersi açın ve kendi tarzınızda dersi işlemeye(çalışmaya) başlayın. Her örneği, kitapta karşı karşıya geldiğiniz sayıda (veya tavsiye edilen sayıda) tekrar edin.
3. ders ile 10. ders arasındaki projeleri(örnekleri) çalışın. Eğer yardıma ihtiyacınız olursa, bütün tamamlanmış projeleri, Projeler dizininde bulabilirsiniz.*

Kolay gelsin !

Herkes İçin
Visual Basic

İÇİNDEKİLER

1. Visual Basic'e Giriş

<i>Bill ve Paul hakkında bir hikaye</i>	<i>1-1</i>
<i>Haydi, başlayalım.....</i>	<i>1-3</i>
<i>Visual Basic'e Başlayalım.....</i>	<i>1-5</i>
<i>Bir Visual Basic Projesini Açmak.....</i>	<i>1-10</i>
<i>Bir Visual Basic Projesini Çalıştırmak (Run).....</i>	<i>1-14</i>
<i>Bir Visual Basic Projesini Durdurmak.....</i>	<i>1-19</i>
<i>Visual Basic'i Durdurmak.....</i>	<i>1-19</i>
<i>Özetle.....</i>	<i>1-20</i>

2. Visual Basic Ortamı

<i>Tekrar Gözden Geçirme ve Bir Önbakış.....</i>	<i>2-1</i>
<i>Bir Visual Basic Projesinin Parçaları.....</i>	<i>2-2</i>
<i>Visual Basic Programının Parçaları.....</i>	<i>2-4</i>
<i>Ana Pencere (Main Window).....</i>	<i>2-4</i>
<i>Form Penceresi (Form Window).....</i>	<i>2-5</i>
<i>Araç Kutusu Penceresi (Toolbox Window).....</i>	<i>2-6</i>
<i>Özellikler Penceresi (Properties Window).....</i>	<i>2-8</i>
<i>Proje Penceresi (Project Window).....</i>	<i>2-10</i>

<i>Visual Basic İçinde Bir Tur.....</i>	<i>2-12</i>
<i>Proje Penceresi (Project Window).....</i>	<i>2-12</i>
<i>Özellikler Penceresi (Properties Window).....</i>	<i>2-13</i>
<i>Kod Penceresi (Code Window).....</i>	<i>2-16</i>
<i>Özetle.....</i>	<i>2-20</i>

3. İlk Visual Basic Projeniz

<i>Tekrar Gözden Geçirme ve Bir Önbakış.....</i>	<i>3-1</i>
<i>Bir Visual Basic Projesi Yapılandırmanın Adımları.....</i>	<i>3-2</i>
<i>Kontrollerin Form Üzerine Yerleştirilmesi.....</i>	<i>3-5</i>
<i>Örnek.....</i>	<i>3-6</i>
<i>Kontrol Özelliklerini Düzenlemek (Dizayn Modu).....</i>	<i>3-8</i>
<i>Örnek</i>	<i>3-10</i>
<i>Kontrolleri İsimlendirme (Naming Controls).....</i>	<i>3-13</i>
<i>Run Mod'unda Özellikleri Düzenlemek.....</i>	<i>3-14</i>
<i>Olay Prosedürlerinde Kontrol İsimleri Nasıl Kullanılır.....</i>	<i>3-15</i>
<i>Olay Prosedürleri Yazmak</i>	<i>3-17</i>
<i>Örnek.....</i>	<i>3-18</i>
<i>Özetle.....</i>	<i>3-23</i>

4. Proje Dizaynı, Formlar, Komut Butonları(Command Buttons)

<i>Tekrar Gözden Geçirme ve Bir Önbakış.....</i>	<i>4-1</i>
<i>Proje Dizaynı.....</i>	<i>4-2</i>
<i>Bir Visual Basic Projesini Kaydetmek (Save).....</i>	<i>4-4</i>
<i>Çevirim İçi Yardım (On-Line Help).....</i>	<i>4-6</i>
<i>Kontrol Olarak Form.....</i>	<i>4-8</i>
<i>Özellikler (Properties).....</i>	<i>4-8</i>
<i>Örnek.....</i>	<i>4-9</i>
<i>Olaylar (Events).....</i>	<i>4-10</i>
<i>Komut Butonu Kontrolü (Command Button Control).....</i>	<i>4-12</i>
<i>Özellikler (Properties).....</i>	<i>4-12</i>
<i>Örnek.....</i>	<i>4-13</i>
<i>Olaylar (Events).....</i>	<i>4-16</i>
<i>BASIC – İlk Ders.....</i>	<i>4-17</i>
<i>Olay Prosedürü Yapısı (Event Procedure Structure).....</i>	<i>4-17</i>
<i>Atama Bildirileri(Assignment Statement).....</i>	<i>4-18</i>
<i>Özellik Tipleri</i>	<i>4-18</i>
<i>Yorumlamalar-Kodlama İçine Notlar Düşme(Comments).4-21</i>	
<i>Proje – 'Eğlenceli Form'</i>	<i>4-22</i>

Proje Tasarımı.....	4-22
Form Üzerine Kontrollerin Yerleşimi	4-22
Kontrol Özelliklerini Düzenleme.....	4-24
Olay Prosedürlerini Yazmak.....	4-25
Projeyi Çalıştırın.....	4-30
Denemeniz Gereken Diğer Bazı Şeyler.....	4-30
Özetle.....	4-32

5. Etiketler(Labels), Metin Kutuları(Text Boxes), Değişkenler(Variables)

Tekrar Gözden Geçirme ve Bir Önbakış.....	5-1
Bir Visual Basic Projesinin Hatalarını Ayıklamak (Debugging).....	5-2
Söz Dizimi Hataları (Syntax Errors).....	5-3
Çalışma Süresi Hataları (Run-Time Errors)	5-4
Mantıksal Hatalar (Logic Errors).....	5-8
Etiket Kontrolü (Label Control).....	5-10
Özellikler (Properties).....	5-10
Örnek.....	5-11
Olaylar (Events).....	5-13
Metin Kutusu Kontrolü (Text Box Control).....	5-14
Özellikler (Properties).....	5-14
Örnek.....	5-15
Olaylar (Events).....	5-17
BASIC – İkinci Ders.....	5-18
Değişkenler (Variables).....	5-18
Değişken İsimleri (Variable Names).....	5-19
Değişken Tipleri (Variable Types).....	5-19
Değişkenleri Deklare Etmek (Declaring Variables).....	5-20
Aritmetik İşlemciler (Arithmetic Operators).....	5-24
Val ve Str Fonksiyonları.....	5-28
String Sonuçlarının Sıralanması.....	5-31
Proje-‘Tasarruf Hesabı’.....	5-33
Proje Dizaynı.....	5-33
Kontrolleri Form Üzerine Yerleştirin.....	5-33
Kontrol Özelliklerini Düzenleyin.....	5-34
Olay Prosedürleri Yazın.....	5-36
Projeyi Çalıştırın (Run).....	5-38
Denenmesi Gereken Diğer Bazı Şeyler.....	5-40
Özetle.....	5-43

6.Kaydırma Çubukları(Scroll Bars), Kararlar(Decisions), Rasgele Sayılar(Random Numbers)

Tekrar Gözden Geçirme ve Bir Önbakış.....	6-1
---	-----

Kaydırma Çubukları Kontrolü (Scroll Bar Control).....	6-2
Özellikler (Properties).....	6-3
Örnek.....	6-7
Olaylar (Events).....	6-8
BASIC – Üçüncü Ders.....	6-10
Mantıksal İfadeler (Logical Expressions).....	6-10
Kıyaslama İşlecileri (Comparison Operators).....	6-11
Mantıksal İşleciler (Logical Operators).....	6-14
Karar verme- If (Eğer) İfadeleri.....	6-18
Rasgele Sayı Fonksiyonu (Random Number Function).....	6-24
Proje – ‘Sayıyı Tahmin Et ’ Oyunu.....	6-28
Proje Tasarımı.....	6-28
Kontrolleri Form Üzerine Yerleştirin.....	6-29
Kontrol Özelliklerini Düzenleyin.....	6-29
Olay Prosedürleri Yazın.....	6-32
Projeyi Çalıştırın (Run).....	6-36
Denemeniz Gereken Diğer Bazı Şeyler.....	6-36
Özetle.....	6-38

7. Icons(İkonlar), Frames(Çerçeveler), Check Boxes(Onay Kutuları), Option Buttons(Seçenek butonları)

Tekrar Gözden Geçirme ve Bir Önbakış.....	7-1
İkonlar (Icons).....	7-2
IconEdit Programını Kullanarak, Kendi İkonunuzu Tasarlamak.....	7-2
İkonları Formlara Atama (İkonları Formlarda Kullanma).....	7-7
Çerçeve Kontrolü (Frame Control).....	7-8
Özellikler (Properties).....	7-8
Bir Çerçeve İçine Kontrollerin Yerleştirilmesi.....	7-9
Örnek.....	7-10
Onay Kutusu Kontrolü (Check Box Control).....	7-12
Özellikler (Properties).....	7-12
Örnek.....	7-14
Olaylar (Events).....	7-14
Seçenek Butonu Kontrolü (Option Button Control).....	7-15
Özellikler (Properties).....	7-16
Örnek.....	7-17
Olaylar (Events).....	7-18
BASIC – Dördüncü Ders.....	7-19
Kararlar-Seçim Durumu (Decisions - Select Case).....	7-19
Proje – ‘ Sandviç Yapımcısı ’.....	7-24
Proje Dizaynı.....	7-24
Kontrolleri Form Üzerine Yerleştirin.....	7-25

<i>Kontrol Özelliklerini Düzenleyin</i>	7-26
<i>Olay Prosedürleri Yazmak</i>	7-32
<i>Projeyi Çalıştırın (Run)</i>	7-40
<i>Denemeniz Gereken Diğer Bazı Şeyler</i>	7-41
<i>Özetle</i>	7-43

8. *Picture Boxes(Resim Kutuları), Mouse Events(Fare Olayları), Colors(Renkler)*

<i>Tekrar Gözden Geçirme ve Bir Önbakış</i>	8-1
<i>Resim Kutusu Kontrolü (Picture Box Control)</i>	8-2
<i>Özellikler (Properties)</i>	8-2
<i>Resim Özelliği (Picture Property)</i>	8-4
<i>Örnek</i>	8-6
<i>Grafik Metotları (Graphics Methods)</i>	8-8
<i>PSet Metodu</i>	8-10
<i>Örnek</i>	8-10
<i>Çizgi Metodu (Line Method)</i>	8-11
<i>Örnek</i>	8-14
<i>Daire-Çember Metodu (Circle Method)</i>	8-14
<i>Örnek</i>	8-16
<i>Cls Metodu</i>	8-17
<i>BASIC – Beşinci Ders</i>	8-17
<i>Fare Olayları (Mouse Events)</i>	8-17
<i>Fare Aşağı Olayı (MouseDown Event)</i>	8-17
<i>Fare Yukarı Olayı (MouseUp Event)</i>	8-20
<i>Fare Taşıma Olayı (MouseMove Event)</i>	8-20
<i>Renkler (Colors)</i>	8-22
<i>Sembolik Sabitler (Symbolic Constants)</i>	8-22
<i>QB Renk Fonksiyonu (QBColor Function)</i>	8-24
<i>RGB Fonksiyonu (RGB Function)</i>	8-26
<i>Proje – ‘ Eğlenceli Karatahta ’</i>	8-28
<i>Proje Dizaynı</i>	8-28
<i>Kontrolleri Form Üzerine Yerleştirin</i>	8-28
<i>Kontrol Özelliklerini Düzenleyin</i>	8-29
<i>Olay Prosedürleri Yazmak</i>	8-32
<i>Projeyi Çalıştırın (Run)</i>	8-38
<i>Denemeniz Gereken Diğer Bazı Şeyler</i>	8-38
<i>Özetle</i>	8-40

9. Lines(Çizgiler), Shapes(Şekiller), Images(Görüntüler), Arrays (Sıralamalar)

Tekrar Gözden Geçirme ve Bir Önbakış.....	9-1
Çizgi Kontrolü (Line Control).....	9-2
Özellikler (Properties).....	9-2
Örnek.....	9-3
Şekil Kontrolü (Shape Control).....	9-5
Özellikler (Properties).....	9-5
Örnek.....	9-6
Görüntü Kontrolü (Image Control).....	9-8
Özellikler (Properties).....	9-8
Resim ve Sündürme Özellikleri (Picture and Stretch properties).....	9-9
Örnek.....	9-10
Olaylar (Events).....	9-11
Form Katmanları (Form Layers).....	9-11
BASIC – Altıncı Ders.....	9-14
Sıralı Değişkenler (Variable Arrays).....	9-14
For/Next Döngüleri (For/Next Loops).....	9-16
Prosedür Seviye Değişkenleri.....	9-19
Rasgele İskambil Kağıdı Dağıtma Rutini (Shuffle Routine).....	9-21
Proje – ‘Kart Savaşları’.....	9-27
Proje Dizaynı.....	9-27
Kontrolleri Form Üzerine Yerleştirin.....	9-27
Kontrol Özelliklerini Düzenleyin.....	9-29
Olay Prosedürleri Yazmak.....	9-34
Projeyi Çalıştırın (Run).....	9-43
Denemeniz Gereken Diğer Bazı Şeyler.....	9-43
Özetle.....	9-44

10. Timers(Zamanlayıcılar), Animation(Hareketli Görüntüler), Keyboard Events(Klavye Olayları)

Tekrar Gözden Geçirme ve Bir Önbakış.....	10-1
Zamanlayıcı Kontrolü.....	10-2
Özellikler (Properties).....	10-2
Olaylar (Events).....	10-3
Örnekler.....	10-3

BASIC – Son Ders.....	10-9
Animasyon(Hareketli Görüntü) – Taşıma Metodu.....	10-9
Gözden Kaybolma Kontrolü.....	10-12
Sınırı geçme.....	10-14
Çarpışma Tespiti.....	10-18
Klavye Olayları (Keyboard Events).....	10-21
Tuşa Basılı Tutma Olayı (KeyDownEvent).....	10-21
Tuşa Dokunma Olayı (KeyPress Event).....	10-24
Proje – ‘Balonlar’.....	10-28
Proje Dizaynı.....	10-28
Kontrolleri Form Üzerine Yerleştirin.....	10-28
Kontrol Özelliklerini Düzenleyin.....	10-29
Olay Prosedürleri Yazmak.....	10-33
Projeyi Çalıştırın (Run).....	10-44
Denemeniz Gereken Diğer Bazı Şeyler.....	10-44
Özetle.....	10-46
Ödül Proje- Pong’.....	10-48

Herkes İçin
Visual Basic

Visual Basic’le Tanışma

Bill ve Paul hakkında bir hikaye

Bill ve Paul, bundan yıllarca önce, 1970’li yıllarda, Seattle, Washington Lakeside School’da iki iyi arkadaşlıklar.

Bill ve Paul bilgisayarlarla çalışmayı çok seviyor, oynayabilecekleri Tic-Tac-Toe ve Moon Landing gibi çeşitli oyunlar yazıyorlardı. Bu bilgisayar oyunlarını BASIC (Beginner’s All Purpose Symbolic Instruction Code – Başlangıç Seviyesindekiler İçin Her Amaca Uygun Sembolik Talimat Kodu) adı verilen bir bilgisayar programlama dili kullanarak yazıyorlardı.

1970'li yılların ortalarında, ilk ufak bilgisayar geliştirildiğinde, Bill ve Paul henüz yeni okula başlamışlardı. Zamanla bilgisayar tecrübelerini daha da geliştirdiler ve bu yeni bilgisayara BASIC dilini nasıl kullanacağını öğretebilecek duruma geldiler. Kendi BASIC sürümlerini, 350 \$ fiyatla, kaset üzerinde kaydedilmiş olarak başka insanlara sattılar. BASIC satmaya başlayınca para sahibi de olmaya başladılar ve bir şirket kurmaya karar verdiler. Şirketlerinin ismini duymuş olmalısınız ! Microsoft

Bill, Bill Gates'dir ve Microsoft'u çalıştırarak milyarlarca dolar kazandı ve hala da kazanmaya devam ediyor.

Paul, Paul Allen, artık Microsoft için çalışmıyor. İçinde bilgisayar, yazılım, profesyonel spor malzemeleri ve emlak işleri bulunan, kendi işlerini yürütüyor.

Evet, Microsoft'un sattığı ilk ürün, bir bilgisayar dili olan BASIC'tir. Ürünleri olan BASIC, Microsoft tarafından geçen yıllar içinde değişti, gelişti ve değişik isimler aldı.

Eski sürümleri GW-BASIC, QuickBasic ve QBasic, ve artık en yeni sürümü ise Visual Basic adı ile anılıyor.

Visual Basic ; öğrenilebilecek en kolay bilgisayar programlama dillerinden birisidir. Öğrenilmesi ve kullanımı oldukça kolay olmasına karşın, bununla ters orantılı olarak, çok güçlü ve iddialı bilgisayar programlarının hazırlanıp geliştirilebileceği bir programlama dilidir.

Bu kitap ile, kendi bilgisayar programlarınızı hazırlamak üzere, Microsoft Visual Basic'i nasıl kullanacağınızı öğreneceksiniz. Bill ve Paul gibi milyarlarca dolar kazanamayabilirsiniz, fakat umarız nasıl kullanılacağını öğrenerek, büyük zevkler alacağınız, iddialı programlar yazabileceksiniz. Yani, bugün irili ufaklı pek çok kuruluş ve şirketin yoğun bir şekilde aradığı ve giderek artan bir tempo ile arayacağı bir "Visual Basic Bilen " olacaksınız !

Haydi, başlayalım !

Bir bilgisayar programı yazmak üzere Visual Basic öğrenmek (yeni olan diğer herhangi bir şeyi öğrenmek gibi), pek çok adım, pek çok yeni ifade ve pek çok yeni deneyim gerektirmektedir. Bunları yavaş, yormadan, her adımını açıklayarak ve her deneyim kazanımını detayları ile ele alacağız.

Başlamadan evvel, aşağıdakileri nasıl yapacağınızı bildiğinizi varsayıyoruz:

- *Bilgisayarı nasıl çalıştıracığınızı ve (mouse) fare'yi nasıl kullanacağınızı biliyor olmalısınız,*
- *İşletim sistemi (Windows 3.1, Windows 95, Windows 98, veya Windows NT) ile nasıl çalışıldığı konusunda az da olsa bilginiz olmalıdır,*
- *Ekran üzerinde açılan pencerelerin boyutlarını nasıl küçültüp büyüteceğinizi ve onları nasıl taşıyacağınızı biliyor olmalısınız,*
- *Bilgisayarınızda Start Menu (Başlat Menüsü) kullanarak bir uygulamayı nasıl çalıştıracığınızı biliyor olmalısınız,*
- *Ekranınızda, Windows ortamında, pop-up(aniden ortaya çıkan) pencerelere nasıl bilgi gireceğinizi biliyor olmalısınız,*
- *Folder ve file (dizin ve dosyalar)'ı tanımalı ve onları bilgisayarınızda nasıl bulacağınızı bilmelisiniz,*
- *File extension (Dosya uzantıları)'nın ne olduğunu ve nasıl tanımlandığını bilmelisiniz. Örnek, Ornek.ext isimli bir dosyada üç harften oluşan ext , uzantı (extension) olarak adlandırılır.*

Eğer bilgisayarınızda bir kelime işlemci, tablolaştırma veya herhangi başka bir bilgisayar programını kullandıysanız, yukarıda sözü edilenler hakkında deneyiminiz. var demektir. Eğer bu konularda tecrübe eksikliğiniz var ise, bilen birisinden yardım alınız.

Bunları bilen kişiler, birkaç dakika içerisinde, bunların nasıl yapıldığını size gösterecektir. Gerçekten de bu konuları öğrenirken, eğer eksiklik hissederseniz, ne zaman olursa olsun sormaktan ve yardım almaktan çekinmeyiniz. Bizler, hepimiz bir zamanlar başlangıç seviyesinde idik ve insanlar gerçekten de bir şeyler öğretmeyi severler.

Devam edelim ! Söylediğimiz gibi yavaş yavaş ilerleyeceğiz.

Bu ilk derste, bilgisayar üzerinde Visual Basic'i nasıl çalıştıracığımızı, bir programı (veya projeyi) Visual Basic'e nasıl yükleyeceğimizi, projeyi nasıl çalıştıracığımızı, nasıl durduracağımızı ve Visual Basic'ten nasıl çıkacağımızı öğreneceğiz.

Visual Basic'e Başlayalım !

Bilgisayarınızda Visual Basic'in herhangi bir sürümünün kurulu ve çalışır durumda olduğunu varsayıyoruz. Eğer böyle değilse, ilk önce bunu yapmalısınız. Tekrar edelim, eğer ihtiyaç duyuyorsanız, birisinin yardımını almak için bu iyi bir fırsattır.

Visual Basic'in geçtiğimiz birkaç yıl içerisinde piyasaya çıkarılan pek çok sürümü mevcuttur. Bu dersleri Visual Basic 4.0 (32 bit sürüm), Visual Basic 5.0, veya Visual Basic 6.0 ile yapabilirsiniz. Dersler süresince Visual Basic'ten kısaca VB olarak bahsedeceğiz ve size, Visual Basic'in hangi sürümünü kullanıyorsanız, sürümüne bağlı olarak ne yapacağınızı anlatacağız.

Eğer Visual Basic 4.0 kullanıyorsanız, sürümünüz ile ilgili bilgiyi, kısa yazımı ile VB4 'den alabilirsiniz. Eğer Visual Basic 5.0 kullanıyorsanız, sürümünüz ile ilgili bilgi için VB5 'e bakınız. Eğer Visual Basic 6.0 kullanıyorsanız, sürümünüz ile ilgili bilgi için VB6 'ya bakmalısınız.

Visual Basic'e başlamak için:

- *Windows'da task bar üzerinde bulunan Start(Başlat) buton'unu tıklayın.*

- *Programs(Programlar)'ı seçin, daha sonra buradan Microsoft Visual Basic # 'ı seçin (burada #, sizin Visual Basic sürümünüzdür).*
- *Visual Basic # 'ı tıklayın (burada yine # sizin Visual Basic sürümünüzdür).*

Buradaki bazı başlıklar bilgisayarınızda ufak farklılıklar gösterebilir, fakat, doğru karşılık gelenleri bulmakta zorluk çekmeyeceksiniz.

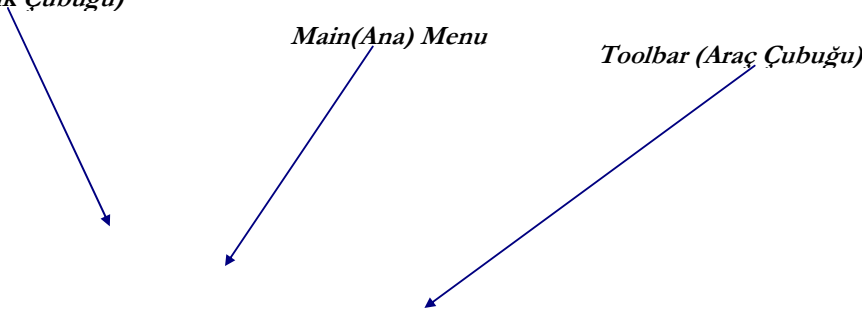
Visual Basic çalıştırılmalıdır. Birkaç pencere, ekranınıza, VB'nizin sürümüne bağlı olarak belirli bir yerleşim biçimi ile gelecektir.

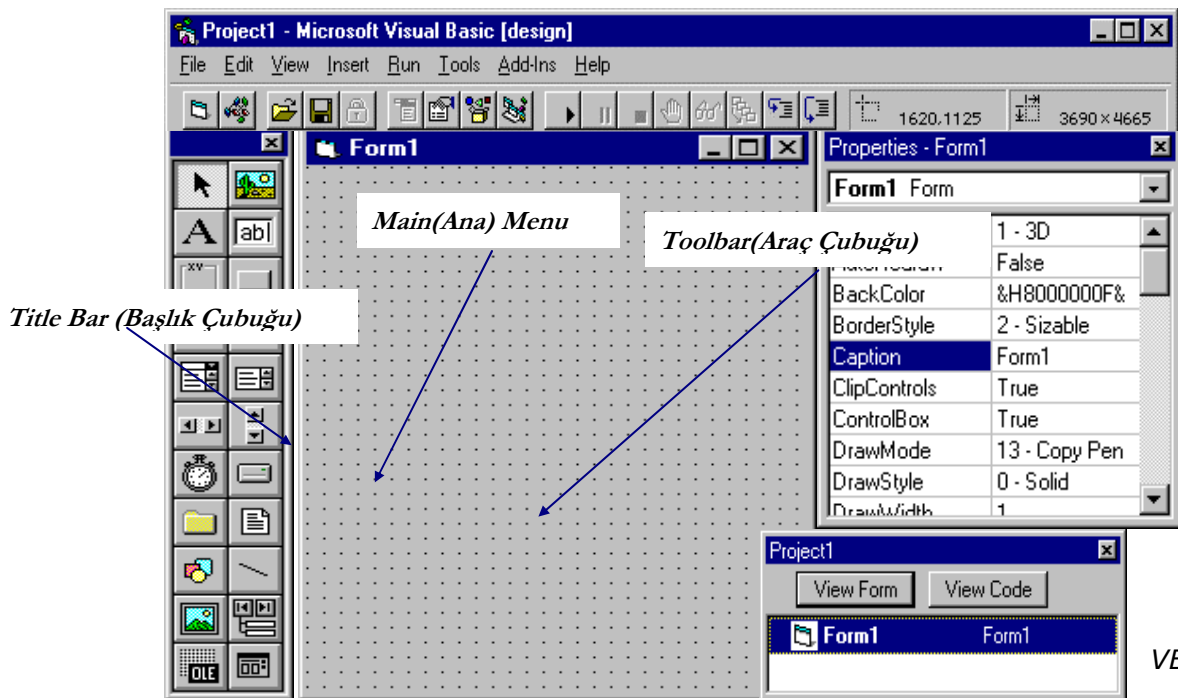
Bar (Başlık Çubuğu)

Main(Ana) Menu

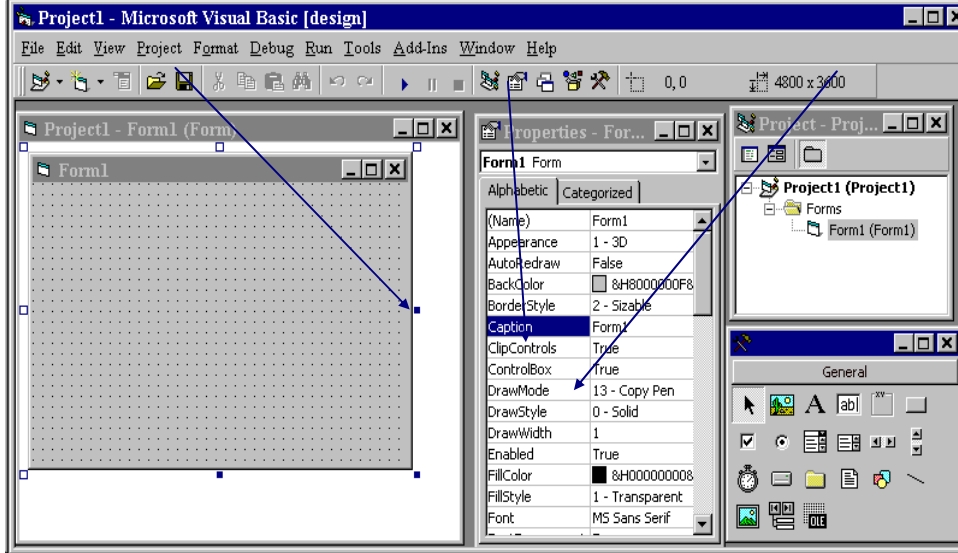
Toolbar (Araç Çubuğu)

VB4

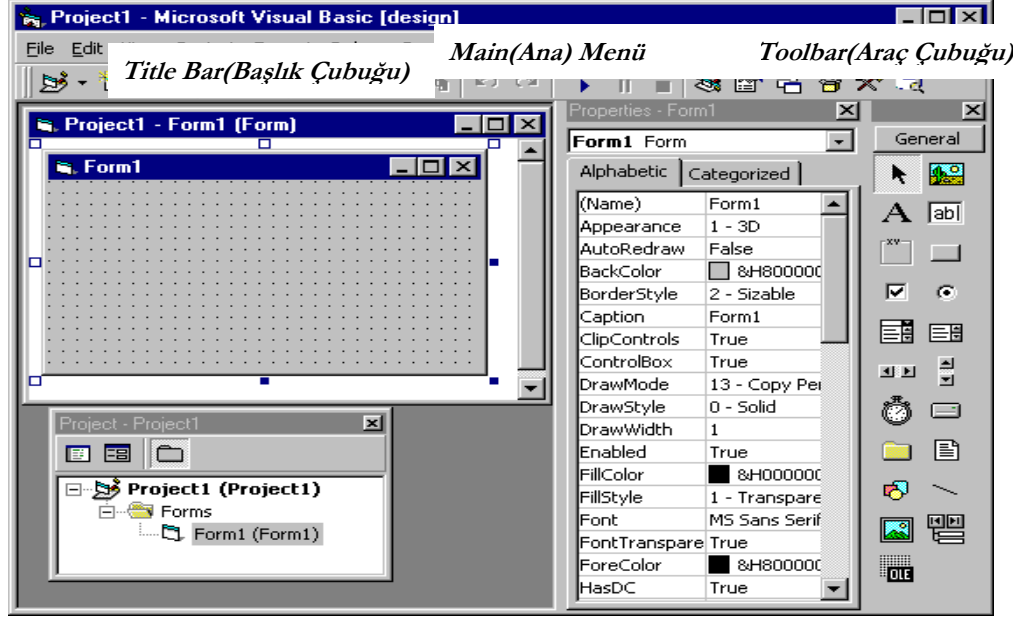




VB5



VB6



VB5 ve VB6'ya başlarken, New Project(Yeni Proje) yazılı olan ve yeni proje adını soran bir diyalog kutusu görünebilir. Eğer böyle ise Open(Aç) butonunu tıklayın.

Ekran üzerinde bulunan bazı başlıklardan bahsedelim:

Ekranın en üstünde Visual Basic Main Window(AnaPencere) bulunur.

Ana Pencere'nin en üstünde Title Bar(Başlık Çubuğu) bulunur. Başlık çubuğu, hangi programı kullandığımız veya hangi Visual Basic programı ile çalıştığımız konusunda bilgi verir. Başlık Çubuğu altında Main(Ana) Menu bulunur. Buradan, Visual Basic Program'ını kontrol ederiz.

Kelime işlemciler veya oyunlar gibi diğer programları kullandığınızdan, menü'lerin nasıl çalıştığını biliyor olmalısınız. Main(Ana)Menu altında ise Toolbar(Araç Çubuğu) bulunmaktadır.

Burada, üzerinde resimler bulunan ufak butonlar (Ana Menü gibi) bize

Visual Basic'i kontrol etme imkanı verir. Eğer fare işaretcisini bu butonlardan birisinin üzerine getirir ve bir-iki saniye tutarsanız küçük bir tooltip (bilgi kutucuğu) görünür.

Bu belirli bir buton'un ne işe yaradığını anlatır – DENEYİN !

Neredeyse bütün Windows Uygulamaları (tablolama, kelime-işlemciler, oyunlar), bize yardım etmek üzere farklı amaçlara yönelik araç çubukları ile donatılmıştır.

Bu tıpkı onlarda olduğu gibi, Visual Basic Araç Çubuğu'nun amacıdır. Yapmak istediğimizi bir sürü şeyi yerine getirmek üzere hazır olacaklardır.

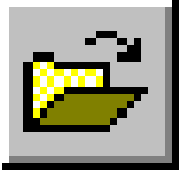
Ekranda bulunan diğerlerini daha sonra ele alacağız.

Bir Visual Basic Projesini Açmak

Hemen şimdi yapacağımız, bir project(proje) açmak olacak. Visual Basic kullanılarak yazılan programlar, proje olarak adlandırılır. Projeler, bizim bilgisayar programımız için gerekli bütün bilgileri içerirler. Visual Basic kullanarak bilgisayar programımızı yazdığımızda, onu proje olarak kaydederiz-saklarız (save).

Bu kitapla birlikte verilen pek çok Visual Basic projesini açabilir (open) ve kullanabiliriz. Haydi, şimdi bir tanesini açalım.

Bir proje açmak için Visual Basic'in araç çubuğunu kullanacağız. Üzerinde dosya açılma resmi bulunan aşağıdaki butonu araç çubuğu üzerine bakarak bulun. Bu aşağıda görünen Open Project (Proje Aç) butonudur.

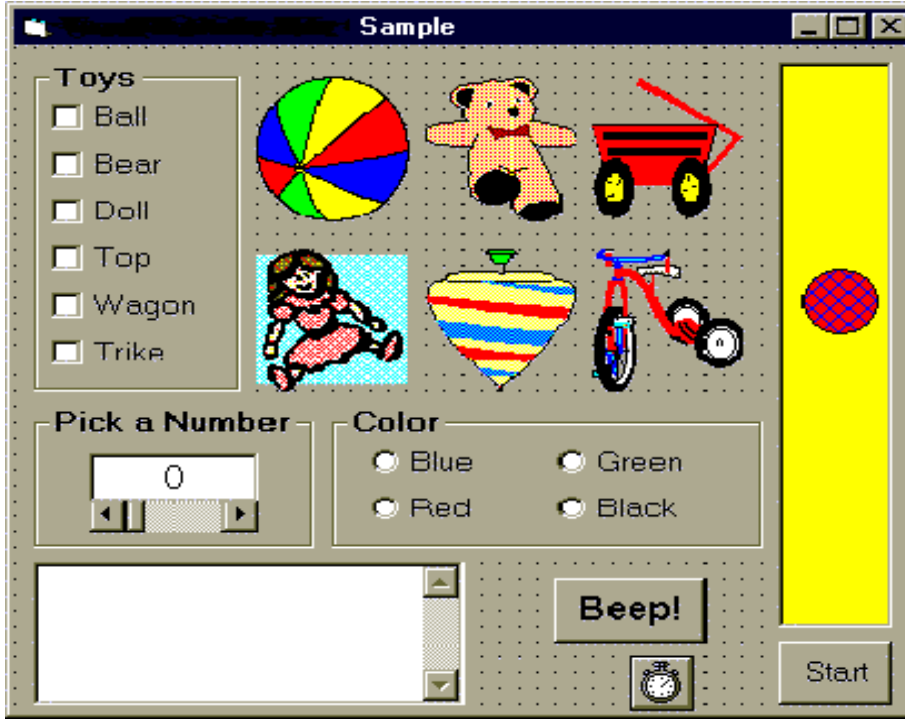


- Bu butonu tıklayın. Open Project penceresi ortaya çıkacaktır.*
- VBProjeleri adlı dizin (folder)' u bulun. Bu dizin bu kitapla ilgili bütün projeleri içinde bulundurmaktadır. Bu dizin'e geçin.*
- Bu dizine bağlı üç alt dizin bulunmaktadır : VB4Projeleri (Visual Basic 4.0, 32 bit sürüm için projeler içermektedir), VB5Projeleri (Visual Basic 5.0 için projeler içermektedir) ve VB6Projeleri (Visual Basic 6.0 için projeler içermektedir).*

Visual Basic Sürümü'nüzle ilgili dizini seçin.

Bu dizine nasıl ulaştığınız, sürekli olarak aklınızda bulunsun. Bu kitap boyunca, ihtiyaç duyduğunuz projeleri açmak üzere bu dizine gideceksiniz.

Şimdi, bu projelerden Sample(Örnek) isimli olanını bulun. Bu dosyayı açın. Ekranda pek çok pencere birlikte belirecektir. Aşağıdaki gibi görünen bir pencere göreceksiniz – Sample :



Eğer projeyi açtığınızda bu pencere görünmüyorsa :

VB4: Ana menü'den önce View'ı daha sonra Form'u tıklayın.

VB5, VB6: Ana menü'den önce View'ı daha sonra Object'i tıklayın.

Eğer pencere görünmüyorsa, bir alternatif olarak, klavyeden <Shift> tuşunu basılı tutarken, F7 fonksiyon tuşuna basın. Bu işlem, söz konusu pencereyi göstermek için klavye girişli bir komut' tur.
Şimdi, burada görünen her şeyi açıklayalım :

Bunlar, Visual Basic'te kullandığımız bazı kelime veya sözcükleri bizimle tanıştıracaktır. Visual Basic'te kullanılan pek çok kelime veya sözcük vardır. Her şeyi ezberlemeye çalışmayın. Bu yeni kelimeleri birçok kez bu kitap boyunca göreceğiz.

Görünen proje penceresine Form(Form) adı veriyoruz. Bütün Visual Basic projeleri (veya programları) form'lar kullanılarak oluşturulur. Gerçekte, belki siz de farketmişsinizdir, bütün Windows uygulamaları herhangi bir tipte form kullanarak hazırlanmışlardır. Form'un üstünde Başlık Çubuk 'u bulunmaktadır. Üzerinde, sol kenarda, formla ilgili bir icon (ikon - küçük resim) form'un ne iş yaptığını gösterir. Burada bulunan yazı, formun yaptığı işi açıklar, (örneğimizde - Sample-Örnek). Burada, sağ kenarda yan yana dizili, daha küçük büyüklükteki üç adet buton, formun görünümünü kontrol etmek üzere burada bulunmaktadır (şimdilik bu üç butonu unutalım). Form üzerinde daha başka pek çok şey bulunmaktadır. Bunlar Visual Basic bilgisayar program'ının 'kalbi' dir.

Form üzerinde, kenarlarında farklı oyuncak isimlerinin bulunduğu bir takım kare butonlar bulunmaktadır. Bu oyuncakların resimlerini de görüyorsunuz.

Yanlarında renk isimleri yazılı olan bir takım yuvarlak butonlar görmektesiniz.

İçine bir şey yazabileceğiniz ve sağ tarafında adı scroll bar (kaydırma çubuğu) olan bir kutu bulunmaktadır.

Form içinde üzerinde Beep yazılı olan bir buton bulunmaktadır.

Bunlara ilaveten, bir sayının değerini almak için ufak bir alet de (pick a number) bulunmaktadır. Büyükçe bir dikdörtgen içerisinde bir top ve bunun altında, üzerinde Start (Başlat) yazan bir buton ve ufak bir saat gibi görünen bir nesne bulunmaktadır.

Form üzerinde bulunan ve yukarıda tanımladığımız bütün bunlara Controls (Kontroller) veya Objects (Nesneler) diyoruz. Kontroller; bir interface(arabirim)'dir veya başka bir deyişle sizin ile (veya program kullanıcısı ile) bilgisayar arasında iletişim

sağlar. Kontrolleri kullanarak, bilgisayara birtakım şeyler söyleriz. Bilgisayar, bunu takiben kendisinden ne hesaplaması veya sonuçlandırması istendi ise bunları sonuçlandırır ve bu sonuçları kontroller üzerinden gerisin geriye kullanıcıya gösterir.

Her koşulda (her durumda) form'un kendisi de bir kontrol'dür.

Eğer herhangi bir Windows uygulaması kullandı iseniz, önceden kontrolleri görmüş olacaksınız – muhtemelen bunların kontrol olarak adlandırıldıklarını da bilmiyordunuz. Örnek olarak, araç çubukları üzerinde bulunan butonlar birer kontroldür. Kelime işlemcilerde bulunan ve dökümanlar için kullanılan kaydırma çubukları, menü seçenekleri ve dosyaları açarken veya kaydederken fare ile tıklayarak kullandığınız butonlar, birer kontroldür.

Sanırım, "kontrol" 'lerin Visual Basic'in çok önemli parçaları olduğu fikri sizde oluştu. Gerçekten de haklısınız ! Bunlar Visual Basic'in en önemli kısımlarıdır ve uygulamanızı oluşturmanıza izin veren araçlardır.

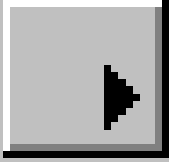
Bu dersin çoğunu, sadece kontrolleri anlamak için harcadık. Hemen şimdi, bu programı çalıştıralım (run) ve bir Visual Basic Projesi'nin (ve onun kontrollerinin) nasıl çalıştığını anlamak üzere, içine, iç yapısına bir göz atalım.

Bu ekrandan ayrılmadan evvel Visual Basic ana penceresi başlığına, köşeli parantez içindeki kelime'ye [design] bakalım.. Bu, bize Visual Basic'in Design (dizayn-tasarım) modunda olduğunu bildirir. Bu, bir Visual Basic projesi oluşturmak için kullanılan bir mod'dur. Sürekli olarak, başlık çubuğuna, Visual Basic'in hangi mod'da çalıştığını görmek için bakın.

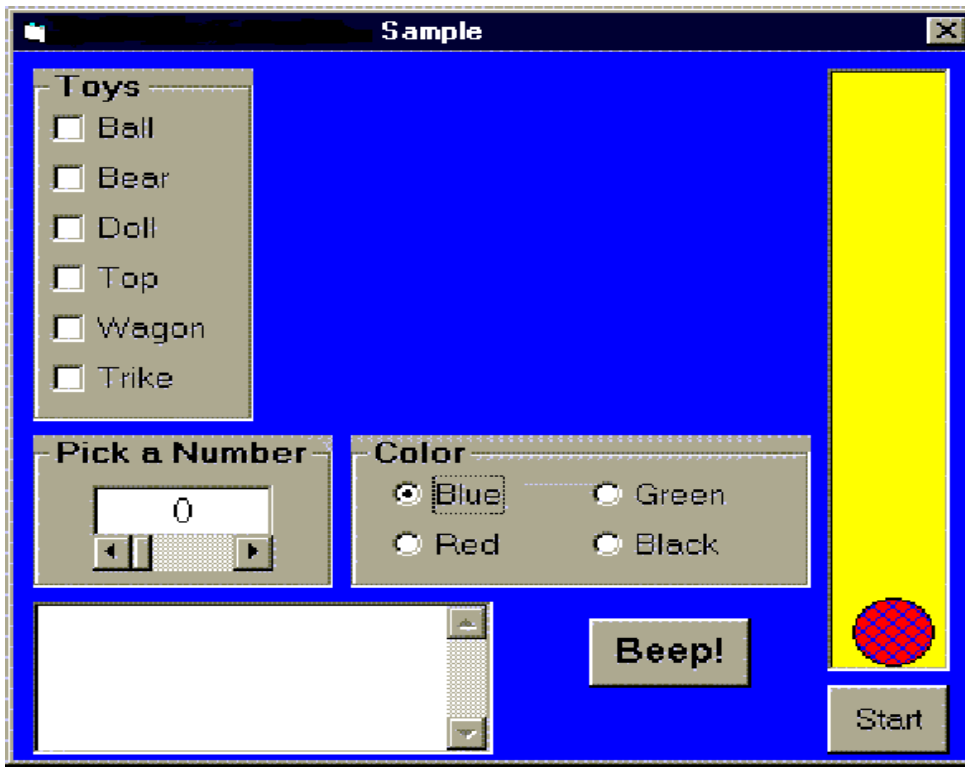
Bir Visual Basic Projesini Çalıştırmak (Run)

Bir Visual Basic projesi oluşturduktan sonra, programı başlatmak (veya çalıştırmak) isteyebilirsiniz.

Bu işlem programın çalışmasını (ilerlemesini) ve kullanıcının kontroller yolu ile form üzerinden etkileşime girmesini ve bilgisayarın kendisine verilen görevleri yerine getirmesini sağlar. Araç çubuğunu kullanarak bir projeyi çalıştırabiliriz. CD cihazları, kasetli teypler veya video cihazları üzerinde bulunan Play tuşuna benzeyen bu butondur :



- *Sample (daha önce açtığımız) projesini çalıştırmak için bu butonu tıklayın.*



Proje form'u ortaya çıkacak ve aşağıdakiler olacaktır:

Dikkat çekici olan, birkaç şeyin değiştiğidir. Bütün oyuncaklar gözden kayboldu. Form'un background (artalan) rengi mavi oldu. Yanında blue(mavi) yazan yuvarlak buton içinde siyah bir nokta oluştu. Ufak saat kontrolü gözden kayboldu. Küçük top büyük dikdörtgenin dibine indi. Peki, neler oldu ?

Visual Basic'i daha fazla öğrendikçe, bunların nasıl ve neden olduğunu bulacak ve anlayacağız. Aynı zamanda, ana pencere içinde, Visual Basic'in şimdi artık Run(Çalıştır) mod'unda olduğunu belirtmek gerekir.

Proje şimdi çalışıyor, fakat ne yapıyor? Hiçbir şey olmuyor, veya gerçekten öyle mi ? Bu noktada Visual Basic, sizi, kullanıcıyı bir şeyler yapması için bekliyor. Biz buna Visual Basic projesi, bir event(olay) oluşmasını bekliyor diyoruz.

Bir Visual Basic programında bir event(olay) gerçekleşene kadar hiçbir şey olmaz. Bu yüzden biz Visual Basic'e event-driven(olay-güdümlü) programlama dili diyoruz. Öyleyse bir event(olay) gerçekleştirelim.

Bir *olay*, form üzerinde bir şey yaptığınızda; fare ile birşeyi tıkladığınızda, üzerine yazı yazılabilecek yerlere yazı yazdığınızda veya bir nesneyi form boyunca sürüklediğinizde oluşur.

Formun en üst sol köşesinde, dikdörtgen içine alınmış bir alan içinde altı adet kutucuktan oluşan bir grup, Toys(Oyuncaklar) başlığı altında yer almaktadır. Bu her küçük kutucuğun kenarında, farklı bir oyuncağın adı yer almaktadır. Bu kutucuklardan birisini tıklayın. Neler olduğunu izleyin.

İşaretlediğiniz kutucuk içinde bir check (onay-OK) işareti çıkacak ve ekranda işaretlediğiniz bu kutucuğun yanında yazan oyuncağın resmi belirecektir.

Bir kutucuğu tıkladığımızda, Click (tıklama) adını verdiğimiz bir olay'a sebep olduk. Eğer belirli bir olay gerçekleştirilirse, bilgisayar olay'ı fark edecek ve bilgisayar programı yolu ile ona ne yapmasını istediğinizi bildirdiğinizden bu işi yerine getirecektir.

Söz konusu bu durum için belirli bir olay, bilgisayara seçilen oyuncağı göstermesini söyleyecektir. Tekrar bu kutuyu tıklayın. Check (onay) işareti ve yanında yazılı isme sahip oyuncağın görüntüsü kaybolacaktır.

Siz böylelikle başka bir olay'a sebep olarak bilgisayara oyuncağın görüntüsünü kaldırmasını söylediniz.

Bu tipte kontroller check box(onay kutusu) olarak adlandırılır. İsteddiğiniz kadar değişik kutucuğu işaretleyerek hangi oyuncakları seçtiğinizi bilgisayara bildirirseniz, bu oyuncakların görüntüleri ekran'a gelecektir. Kontrol kutuları, bir liste içinden bazılarını seçmek istediğinizde kullanılır.

Söz konusu bu örnekte, başka iki kontrol de kullanılmıştır. Kontrol kutularının içinde bulunduğu ve bir dikdörtgen ile sınırlandırılmış bu alana frame (çerçeve) adı veriyoruz. Oyuncak resimlerinin görüldüğü alana image(görüntü) kontrol'ü adı veriyoruz.

Şimdi form üzerinde bulunan olay'lara sebep olan diğer kontrolleri inceleyelim:

Bir çerçeve içerisinde bulunan, dört yuvarlak butondan oluşan ve bir çerçeve içerisine alınmış, ekranın ortasına yakın, Color(Renk) başlıklı kontrollere bakalım. Her buton'un yanına bir renk ismi yazılmıştır. Blue(Mavi) butonunun içinde siyah bir nokta bulunmaktadır ve o anda seçili rengin bu olduğunu belirtir (form renginin mavi olduğuna dikkat edin). Bu butonlardan başka birisini tıklayın. Ne olduğunu izleyin. Form rengi seçilen renge dönüşecektir. Bu Click(tıklama) olayı bilgisayara form'un background (artalan) rengini değiştirmesini söylemektedir. Yeni bir renk seçtiğinizde dikkat edin; seçtiğiniz buton içinde siyah bir nokta belirecek ve daha önceden seçtiğiniz buton içindeki siyah nokta kaybolacaktır. Daha evvel gördüğümüz onay kutularından farklı olarak, bu butonlardan sadece birisini seçebilirsiniz. Bu ise, şu anlama gelir – yalnızca bir renk seçebilirsiniz! Bu yuvarlak butonlar'a option button(seçenek butonu) adı veriyoruz. Bu butonlar birçok seçeneği içeren bir listeden kesinlikle sadece bir tanesini seçmek istediğinizde kullanılır.

Toys çerçevesinin altında Pick a Number(Bir Sayı Alın) başlıklı bir çerçeve bulunmaktadır. Label(etiket) adını verdiğimiz bir kontrol bulunmaktadır. Örneğimizde sayının içinde görüldüğü kontrol budur.

Etiket altında başka bir kontrol vardır ; scroll bar(kaydırma çubuğu). Belki de bu kaydırma çubuklarını, kullandığınız başka uygulamalarda (programlarda) görmüşsünüzdür.

Kaydırma çubuğu, görünen sayıyı değiştirmek için kullanılır. Kaydırma çubuğunun sağ tarafında bulunan oku tıklayın. Görünen sayı 1 artacaktır. Bu oku tıklamaya devam ederseniz, görünen sayı birer birer artmaya devam edecektir. Okun solunda bulunan beyaz alanı tıklayın. Sayının değeri 10 artacaktır.

Kaydırma çubuğu ortasındaki ufak çubuk parçasına (thumb adı verilmektedir) tıklayın ve fare'nin tuşunu basılı tutarak önce sağa sonra sola doğru sürükleyin. Fark ettiniz mi? Görünen sayı değeri çok hızlı, yani çok miktarda değişti.

Bu örnekte bilgisayar, kaydırma çubuğunun thumb'ı hareket ettirildiğinde oluşan Change(değiştir) olayına cevap vermektedir. Change(değiştir) olayı, bilgisayara kaydırma çubuğunun değerinin değiştiğini söyler ve bilgisayarın etiket kontrolü içinde yeni değeri göstermesine izin verir.

Pick a Number (bir sayı seç) çerçevesinin altında, sağ tarafında bir kaydırma çubuğu bulunan bir çerçeve bulunmaktadır. Bu kontrol text box(metin kutusu) olarak adlandırılır. Bunu tıklayın ve içine istediğiniz herhangi bir metni yazın. DENEYİN !

Bu metin kutusu sanki küçük bir kelime işlemcidir. Metin kutusuna her seferinde bir şeyler yazdığınızda, birkaç olay olur. Bir tuşa bastığınızda KeyPress (Tuşa Basma) olayı gerçekleşir ve metin kutusu içindeki metin her değiştiğinde Change(değiştir) olayı oluşur.

Metin kutusunun yanında; üzerinde Beep yazılı bir buton bulunmaktadır ve bunu tıklattığınızda bilgisayarınızın hoparlöründen 'bip' sesi duyacaksınız. Bu kontrol command button(komut butonu) olarak adlandırılır ve Visual Basic'te en çok kullanılan kontrollerden birisidir. Click olay'ı, bilgisayara hoparlörden bip sesi çıkarmasını söyler.

Formumuzdaki son şey; uzun, sarı ve dikdörtgen şeklinde adına picture box (resim kutusu) adını verdiğimiz kontroldür. Adına shape (şekil) kontrol'ü adını verdiğimiz küçük bir daireyi içine alır. Resmin altında üzerinde Start(Başla) yazısı bulunan bir komut butonu bulunmaktadır. Bu butonu tıklayın. Bu bir Click olay'ına sebep olacaktır.

Top yukarı doğru, resim kutusunun üstüne çarpana kadar yukarı çıkacaktır. Daha sonra aşağı doğru düşmeye başlayacaktır. Bu aşağı ve yukarı zıplama artık üzerinde Stop(Dur) yazan komut butonunu tıklayana kadar sürecektir.

Hatırlatmakta fayda var, ufak saat; dizayn mod'unda form üzerinde görünür durumdaysa da, projeyi çalıştırdığımızda gözden kaybolacaktır. Zıplayan top örneğinde kullanıldığı gibi, biz bunu timer(zamanlayıcı) kontrolü olarak adlandırıyoruz. Komut butonu üzerindeki tıklama olayı, buton üzerindeki yazıyı Stop 'a dönüştürmesine ilaveten, aynı zamanda zamanlayıcı kontrolünü de çalıştırmaya başlar.

Zamanlayıcı kontrolü, tamamen kendi kendisine, önceden ayarlanmış zaman aralıkları ile olaylar üretir. Bu örnekte timer (zamanlayıcı) olayı her 1/10 saniyede bir tekrarlanır ve bu olaylar serisi ile topun pozisyonu hareket ediyor görüntüsü verecek şekilde değişir.

Belirtmekte fayda var; top aşığı yukarı zıplarken aynı anda form rengini deęiştirebilir, oyuncak resimlerini gösterip, gözden kaybedebilir, yazı yazabilir ve bilgisayara bip sesi verdirebilirsiniz. Yani, Visual Basic, çoklu olay'ları aynı anda yerine getirme yeteneğine sahiptir.

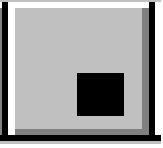
Bu proje, Visual Basic ile, bir tanıtımdan daha ileri bir şey yapamıyor gibi görünmesine rağmen, önemli bir kavramdır.

Bu örnek, sizin kendi Visual Basic projelerinizi nasıl oluşturacağınız konusunda bilgi vermesi açısından önemlidir. Bir proje, kullanıcının, bilgisayara, üzerinden bilgi sağlayabileceęi kontrollerden oluşur. Bu kontroller ile olaylara sebep olarak, bilgisayara gereken sonuçlar ürettirilir. Bu sonuçları hesaplamak için olayları nasıl kullanacağımız konusunda endişelenmeyin, zira bunların hepsini bu kitapta ilerleyen bölümlerde öğreneceğiz. Bu kitabı bitirdiğinizde, Sample projesinin yaptığı herşeyi (hatta çok daha fazlasını) yapan projeleri oluşturabilecek hale geleceksiniz.

Haydi, şimdi projeyi nasıl durduracağımıza bakalım !

Bir Visual Basic Projesini Durdurmak

Bir Visual Basic projesini durdurmak için pek çok yol vardır. Biz araç çubuğunu kullanacağız. CD cihazları, kasetli teypler veya video cihazları üzerinde bulunan ve Stop tuşuna benzeyen bu butondur (projenizde formu biraz aşığı kaydırarak araç çubuğunu ekranda görünür hale getirmeniz gerekebilir):



- *Bu buton'u tıklayın. Proje'nin çalışması duracak ve Visual Basic dizayn mod'una dönecektir.*

Projeyi durdurmak için başka bir alternatif yol ise, form üzerinde bulunan Close(Kapat) butonunu tıklamaktır. Bu, formun sağ üst köşesinde yer alan küçük X şeklinde bir butondur.

Visual Basic'i Durdurmak

Visual Basic projesi üzerinde çalışırken, Visual Basic Programından çıkmak isteyebilirsiniz. Visual Basic'ten çıkma yöntemi, bütün Visual Basic sürümleri için aynıdır. Gerçekte neredeyse bütün Windows uygulamaları için de aynıdır:

- *Ana menü içerisinden File'ı seçin.*
- *Exit seçeneğini tıklayın (File menu'nün dibinde yer almaktadır).*

Şimdi Visual Basic kapandı. Visual Basic bütün açık pencereleri kapatacak ve Windows Desktop'a dönecektir. Visual Basic'i Sample aktif iken kapatmak isterseniz, size bazı dosyaları kaydedip kaydetmeyeceğiniz sorulacaktır. Cevap No(Hayır)'dır. Bir projeyi durdurmak gibi, Visual Basic'ten çıkmak için alternatif bir yol, ana pencerenin en üst sağ köşesinde yer alan kapat(close) butonunu tıklamaktır. Bu, pencerenin sağ üst köşesinde yer alan küçük X şeklinde bir buton'dur.

Özetle ;

Yoruldunuz mu ?

Burada pek çok şey öğrendik. Sizin için tamam ise sorun yok !

Daha evvel söylediğimiz gibi, pek çok yeni ifade ve kavram öğrendiniz. Burada bahsettiğimiz her şeyi hatırlayamazsınız. Dert etmeyin ! Bu bilgileri pek çok kereler tekrar göreceksiniz. Yalnızca bir kavram önem taşımaktadır; bir Visual Basic Projesinde ne olup bitiyor ve nasıl çalışıyor.

Şimdi artık kendi kendinize Visual Basic'i nasıl çalıştırıp, kapatacağınızı biliyorsunuz.

Özetle, bir Visual Basic Projesinin form üzerine kurulduğunu artık biliyoruz. Form üzerindeki Control(Kontrol)'lerin, kullanıcı ile bilgisayarın birbirlerine etkileşime geçmesi için konulduğunu biliyoruz. Kullanıcı, kontroller ile olaylar üretir. Bu ise bilgisayarın kendi görevini yapmasına olanak verir.

Diğer derste, artık yavaş yavaş kendi Visual Basic projelerinizi yapılandırmaya götüren tecrübe kazanma çalışmalarına başlayacağız. Bir projenin parçalarının nasıl birbirine uydurulacağını (birleştirileceğini) göreceğiz. Sample projesini bir örnek olarak kullanarak, bir projenin önemli kısımlarının nasıl yerleştireceğinizi öğreneceksiniz. Ve daha sonra 3. derste ilk gerçek projenizi üreteceksiniz !

Herkes İçin
Visual Basic

2. Visual Basic Ortamı

Tekrar Gözden Geçirme ve Bir Önbakış

İlk derste, bir Visual Basic Projesi'nin önemli kısımlarını tanıdık. Bir projenin, bir form üzerinde, kontrol'ler kullanılarak nasıl yapılandırıldığını öğrendik.

İkinci (bu) derste; playlar'ı kullanarak ve kontroller yolu ile etkileşime girerek, bizim sağladığımız talimatlar ile, istenilen hedefleri (işleri), bilgisayarın yerine getirmesini sağlayacağız.

Yine bu derste, bir Projenin farklı parçalarını kullanarak, kendi Visual Basic Projelerimizi yapılandırma yolunda, başlangıç adımlarını öğreneceğiz.

Ayrıca bunların Visual Basic ortamı içerisinde birbirine nasıl uygun hale getirileceğini öğreneceğiz.

İlk derste olduđu gibi, öğrenilmeyi bekleyen pek çok yeni terim ve deneyimimiz olacak!

Bir Visual Basic Projesi'nin Parçaları

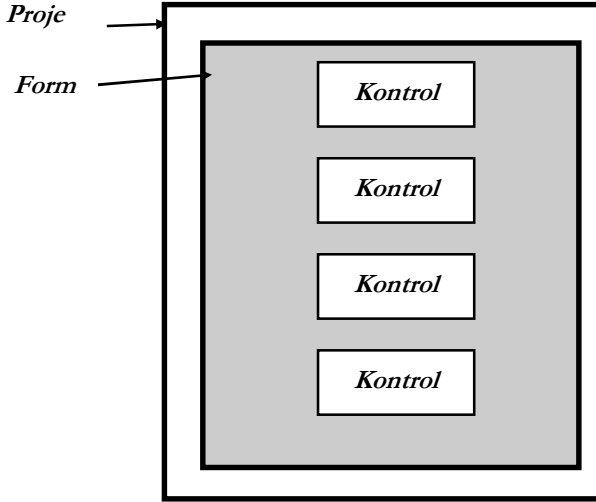
İlk derste, bir Visual Basic Projesinin üç ana bileşeninin olduğunu gördük: projenin kendisi, form ve kontroller.

Visual Basic için Proje kelimesi, proje içinde bulunan her şeyi içine almaktadır. Bir projeyi açıklamak için kullanılan diğer terimler uygulama veya program kelimeleridir.

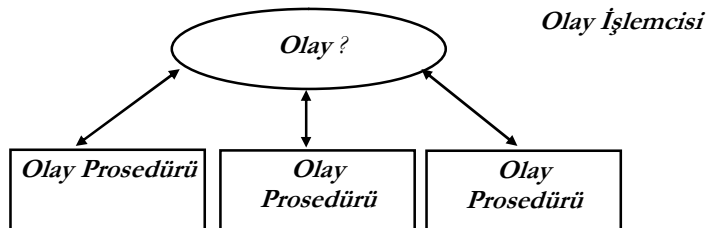
Form bir pencere olup, burada kullanıcı ile bilgisayar arasında arabirim yaratabileceğimiz bir yerdir.

Kontroller ise grafik özellikler veya araçlar olup, kullanıcının bilgisayar ile etkileşime girmesine imkan vermek üzere form üzerine yerleştirilirler (metin kutuları, etiketler, kaydırma çubukları, komut butonları).

Formu çağırmak da bir kontroldür. Kontroller aynı zamanda object(nesne) olarak da adlandırılırlar. Şekil olarak bir proje aşağıdaki gibidir:



Özetle; bir proje birkaç kontrol içeren bir formdan oluşur. Bilgisayarınızda dosyaların bulunduğu dizine göz attığınızda, Visual Basic Projeleri ile ilgili bazı dosyalar gözünüze çarpacaktır.



Olay prosedürleri bizim gerçek bilgisayar programlarını yaptığımız yerlerdir ve uzantısı frm olan dosya içine kaydettiğimiz form ile saklanırlar. Bu prosedürler (yöntemler), BASIC dili statement (bildiri)'lerini kullandığımız yerlerdir.

Bu derste birçok programlama deneyimi ve BASIC dilini öğreneceksiniz. Öğreneceğiniz BASIC, Bill Gates ve Paul Allen'in Microsoft'u başlatırken kullandıkları ilk orijinal BASIC ile büyük benzerlik göstermektedir.

Visual Basic Program'ının Parçaları

Visual Basic; bir bilgisayar programlama dilinden de öte, daha da fazlasıdır.

Bir proje yapılandırma ortamıdır. Bu ortam içerisinde projemize başlayabilir, yapılandırabilir, çalıştırabilir, test edebilir, projemizde (eğer varsa) hatalarımızı ayıklayabilir ve ileride kullanmak üzere projemizi kaydedebiliriz.

Diğer bilgisayar dilleri ile, programınızı yazmak için, ayrı bir Metin Editör'üne, programınızı yaratmak için compiler(derleyici) adı verilen ayrı bir programa ve programınızı test etmek için ayrı bir alana ihtiyaç duyarsınız.

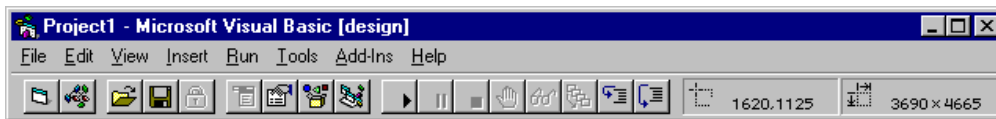
Visual Basic ise, proje oluşturma (yapılandırma) işleminin her adımını bütünleştirerek, hepsini tek bir ortamda gerçekleştirir.

Visual Basic ortamının parçalarına bir gözatalım.

Ana Pencere

Birinci derste öğrendiğiniz prosedür ile Visual Basic'i çalıştırın. Ekranda birkaç pencerenin birden açıldığını gözlemleyin. Ana Pencere, Visual Basic Projesinin pek çok amacını kontrol etmek ve çalıştırma(run) işlemini gerçekleştirmek için kullanılır.

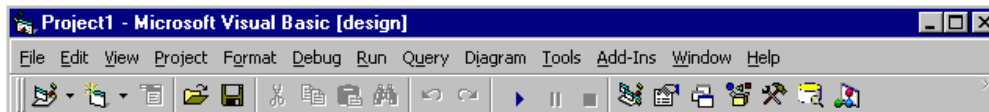
VB4:



VB5:



VB6:



Ana pencere, başlık çubuğu, menü çubuğu ve araç çubuk'undan oluşur.

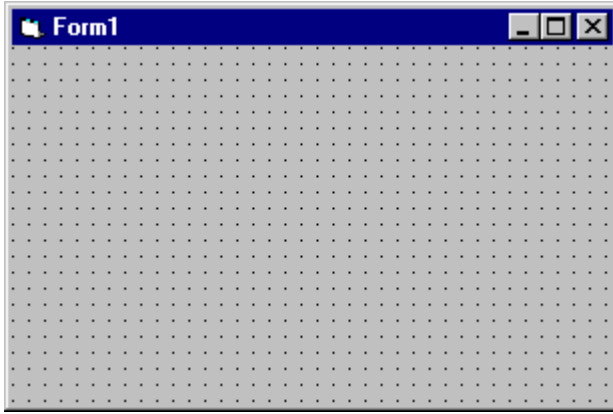
Ufak boyuttaki başlık çubuğu; proje ismini ve o anki çalışma modunu (dizayn, break (ara verme) veya çalışma (run)) gösterir.

Menü çubuğu üzerinde bulunan drop-down menu(çek menü)'lerden, Visual Basic ortamının işlemlerini kontrol ederiz.

Bu menü seçeneklerinden bazılarına kısa yol oluşturmak üzere, araç çubuğu üzerinde bazı butonlar bulunmaktadır. Birinci derste, bir projeyi açmak, çalıştırmak ve durdurmak için kullandığımız butonlara bakınız.

Form Penceresi

Form Penceresi, Visual Basic uygulamaları geliştirmenin 'kalbi'dir. Sizin uygulamanızı geliştirdiğiniz yerdir:



Eğer bu form ekran üzerinde görünmüyorsa:

VB4: Ana menü'den View, daha sonra da Form seçeneğini tıklayın.

VB5, VB6: Ana menü'den View, daha sonra da Object seçeneğini tıklayın.

Bir alternatif olarak, eğer pencere görünmüyorsa, <Shift> tuşunu basılı tutarken, F7 fonksiyon tuşuna basınız.

Araç Kutusu Penceresi

Araç Kutusu Penceresi (Toolbox Window), uygulamalarınızda kullandığınız kontroller için bir seçim menü'südür. Birçok kereler, kontrollerden tools (araçlar) olarak söz edilir. Kısaca, kontrollerden söz etmek için üç değişik kelime kullanılır : object (nesne), tool (araç) ve en çok kullanılan olarak control (kontrol).

VB4:



Pointer (İşaretleyici)

Label (Etiket)

Frame (Çerçeve)

Check Box (Onay Kutusu)

Combo Box (Birleşik Kutu)

Horizontal Scroll Bar (Yatay Kaydırma Çubuğu)

Timer (Zamanlayıcı)

Directory(Dizin)

Shape(Şekil)

Image(Görüntü)

OLE (Nesneyi Bağlantılama ve Dahil Etme)

Picture Box (Resim Kutusu)

Text Box (Metin Kutusu)

Command Button (Komut Butonu)

Option Button (Seçenek Butonu)

List Box (Liste Kutusu)

Vertical Scroll Bar (Dikey Kaydırma Çubuğu)

Drive (Sürücü Liste Kutusu)

Files (Dosya Liste Kutusu)

Line (Çizgi)

Data Control (Veri Kontrol'ü)

*Common Dialog Box (Ortak
Diyalog
Kutusu)*

VB5, VB6:



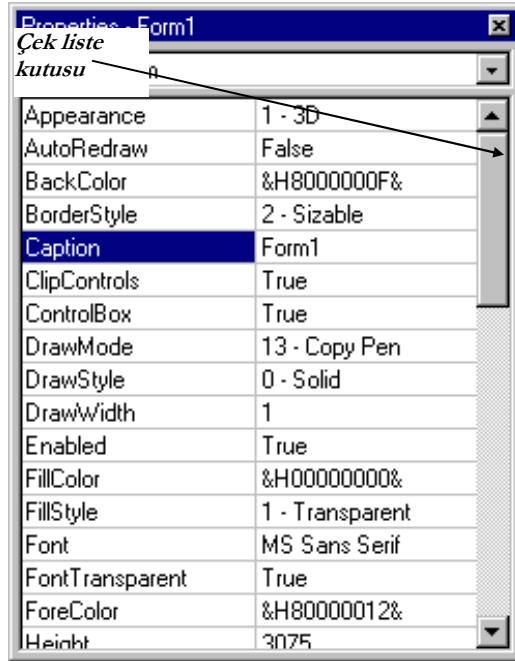
Eğer toolbox(araç çubuğu) ekranda görünmüyorsa, ana menüden önce View daha sonra da Toolbox seçeneğini tıklayın. Birinci derste, Sample projesi ile tanımladığımız kontrolleri inceleyin.

Properties(Özellikler) Penceresi

Properties Penceresi, kontroller için başlangıç özellik değerlerini vermek üzere kullanılır.

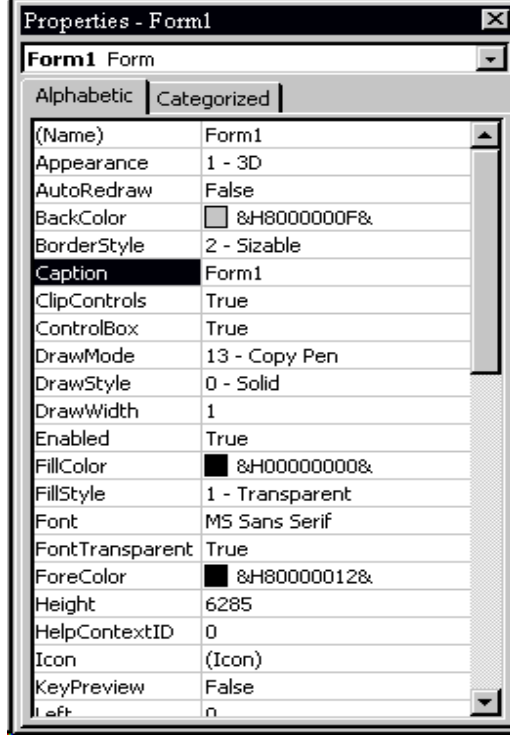
Pencerenin en üstünde yer alan drop-down box (çek kutu), o anki form üzerinde bulunan bütün kontrolleri listeler. Bu kutu altında halihazır seçilmiş nesneye ait uygun özellikler mevcuttur.

VB4:



VB5, VB6:

İki şekilde görüntülenebilir : *Alphabetic (Alfabetik)* ve *Categorized (Sınıflandırılmış)*. Biz daima Alfabetik olanını kullanacağız.



Çek liste
kutusu

Eğer ekranda properties penceresi görünmüyorsa :

VB4: Ana menü'den View, daha sonra da Properties seçeneğini tıklayın.

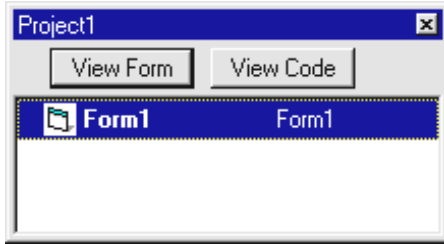
VB5, VB6: Ana menü'den View, daha sonra da Properties Window seçeneğini tıklayın.

Bir alternatif olarak, eğer properties penceresi ekranda görünmüyorsa, F4 fonksiyon tuşuna basınız. Properties penceresi daima form ve onun kontrolleri ile birlikte görüldüğünde ortaya çıkar.

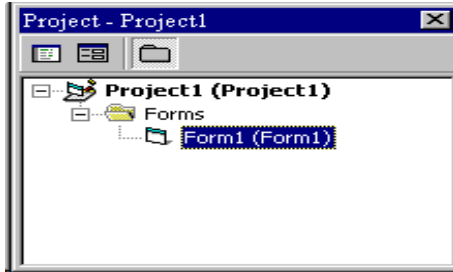
Proje Penceresi

Proje Penceresi hangi formun projenizi oluşturduğunu gösterir. Daha tecrübeli bir Visual Basic programcısı olduğunuzda, **birden fazla sayıda form'dan oluşan projeleri** nasıl oluşturacağınızı öğreneceksiniz. Böylesi bir durum için projenizdeki bütün formlar bu pencerede listelenecektir. Aynı zamanda ekranda proje penceresinde görünen butonlardan birisini tıklayarak Form Penceresi veya Code(Kod) Penceresi'ni (gerçek BASIC kodlamanın görüldüğü pencere) göreceksiniz. Kod penceresini diğer derste göreceğiz.

VB4:



VB5, VB6:



Eğer ekranda **project (proje) penceresi** görünmüyorsa :

VB4: Ana menü'de View, daha sonra da Project seçeneğini tıklayın.

VB5, VB6: Ana menü'de View daha sonra da Project Explorer seçeneğini tıklayın.

Bir alternatif olarak, **eğer project (proje) penceresi** ekranda görünmüyorsa, <Ctrl> tuşunu basılı tutarken R tuşuna basınız.

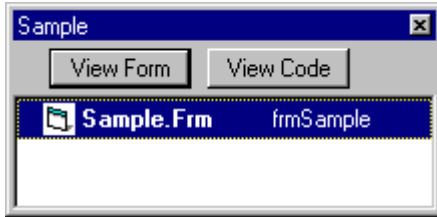
Eğer bu pencerelerden herhangi birisi ekrana gelmezse; onların nerede olduğunu ve onları nasıl yerleştireceğinizi biliyor olmalısınız. Ayrıca, artık bu pencereleri de tanıyorsunuz. Bir sonraki seferde, birinci derste kullandığımız projeyi, burada işlediğimiz bazı konuları değerlendirmek üzere, tekrar ziyaret edeceğiz.

Visual Basic İçinde Bir Tur

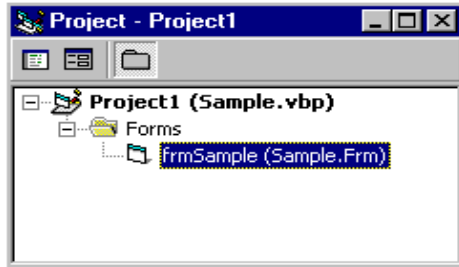
Proje Penceresi

1. ders'te kullandığımız Sample adlı projeyi tekrar açın. Project penceresine gidin ve inceleyin.

VB4:



VB5, VB6:



Proje penceresi; Sample.vbp isimli bir proje dosyasının kaydedildiğini ve projenin bir adet Sample.frm adlı bir proje formuna sahip olduğunu göstermektedir.

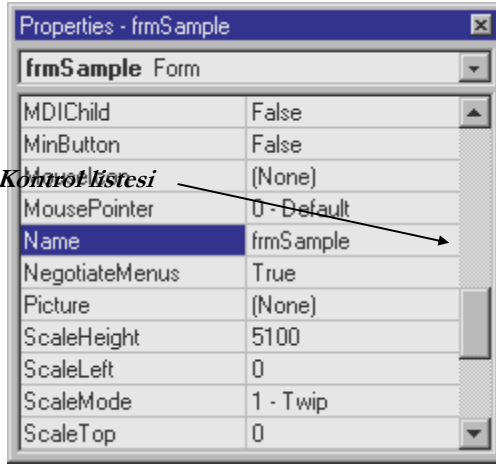
Burada fark etmeniz gereken şey, formun frmSample adı ile anılmasıdır. Properties penceresini hatırladınız mı ? Bir kontrole atadığımız en önemli özelliklerden birisi, onun Name(isim)'inin, 'frmSample', yani formumuzun isim özelliğinin bu olmasıdır. Bizler kontrollere daima isim özellikleri(property) ile değiniriz.

Yani, onlara nasıl bir isim verileceği konusu kritik bir konudur (bu konuda, gelecek derste, birçok şeyden bahsedeceğiz). Proje penceresinde, listelenmiş isim vermek, son derece önemlidir.

Properties(Özellikler) Penceresi

Şimdi Properties(özellikler) penceresine bakalım: Hatırlayın, bu sadece form görüntüye geldiğinde ortaya çıkar. Böylece, sizin ilk yapacağınız şey, öncelikle formun görüntüye geldiğinden emin olun. İstenilen pencereleri nasıl ekrana getireceğiniz konularını gözden geçirin.

VB4:



VB5, VB6:



Properties(özellikler) Penceresi'nin en üstünde bulunan çek kutusu control list(kontrol listesi) olarak adlandırılır. Projede kullanılan her kontrol tipinin ne olduğu kadar, ismini (isim özelliğini) de gösterir. Dikkat edin, şekilde görüldüğü gibi, şu anki kontrol Form 'dur ve ismi de frmSample'dır.

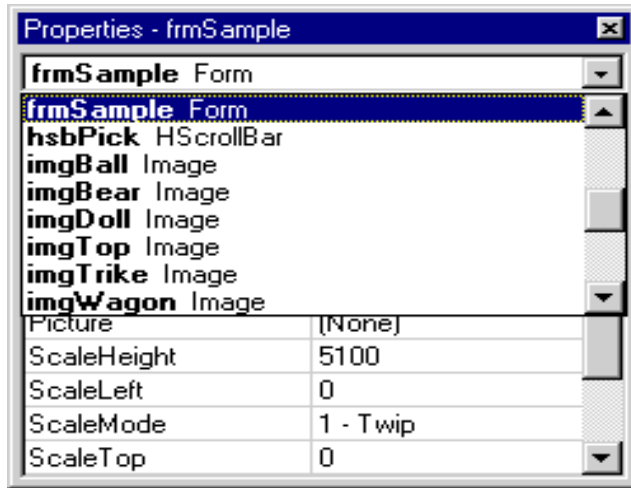
İlgili özelliklerin listesi bu kutunun altındadır. Bu listede, seçilen kontrol için, sıralanmış özellikleri kaydırarak (kaydırma çubuğunu kullanarak) inceleyebilir veya değiştirebilirsiniz. Özelliklerin isimleri listenin solunda yer alırken, şu anki geçerli özelliklerin değerleri ise sağ tarafında yer almaktadır. Form için sıralanan özellikleri kaydırarak inceleyin.

Gördünüz mü, kaç tane özellik var ?

Bunların birçoğunu derslere devam ettikçe öğreneceksiniz.

Kontrol listesindeki aşağı oku tıklayın(hatırlayın bu özellikler penceresinin en üstündeki çek-kutudur).

VB4:



VB5, VB6:

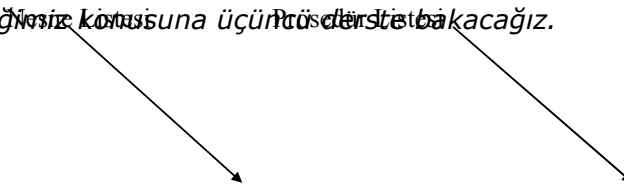


Form üzerinde bulunan bütün kontrollerin görünen listesini kaydırın. Bu kontrollerden pek çoğunu burada göreceksiniz. Bunlara atanan isimlere, kontrol tiplerine ve hangi isimlerin hangi kontrolleri tanımlamak için kullanıldığına dikkat edin. Örnek olarak imgTrike üç tekerlekli bir bisikletin resmini tutan bir image control 'ü olarak kullanılmıştır.

Kontrollere uygun isim vermenin mantığı budur, yani bir kontrolü yalnızca isminden tanıma veya tanımlama mantığıdır. Daha evvel sözünü ettiğimiz gibi, daha sonraki bölümlerde, kontrollere isim verme konusunda pek çok şeyden bahsedeceğiz.

Bir kontrol seçin ve bu kontrol'ün property(özellikler) bilgilerini gezin. Değişik birkaç kontrol için, bunların özelliklerine bakın. Her kontrol başlı başına pek çok property (özellik)'e sahiptir. Pek çok property(özellik) değeri, default(varsayılan)

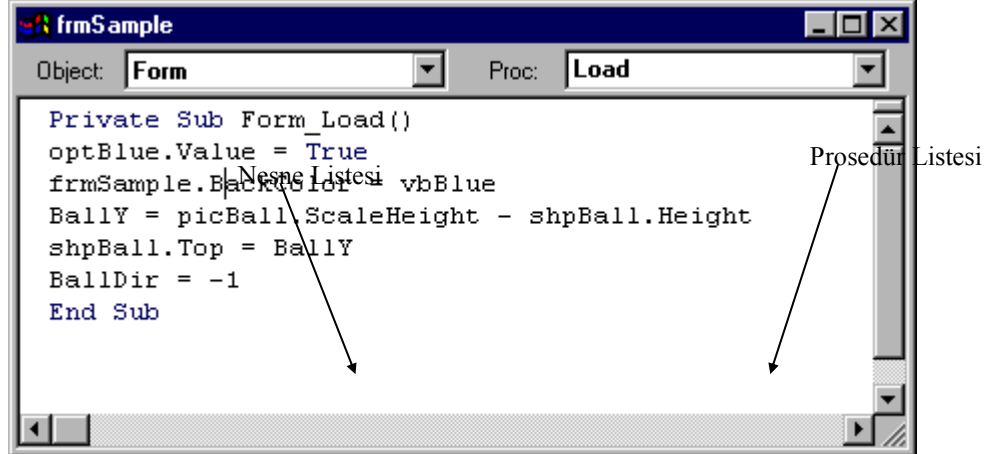
olarak, Visual Basic tarafından verilmiştir. Bizler bu varsayılan (default) değerleri, kendi kullanımımız için gerekli uygun değerler ile değiştireceğiz. Bu property (özellik)'lerini nasıl değiştireceğimize konusuna üçüncü bölümde bakalım.



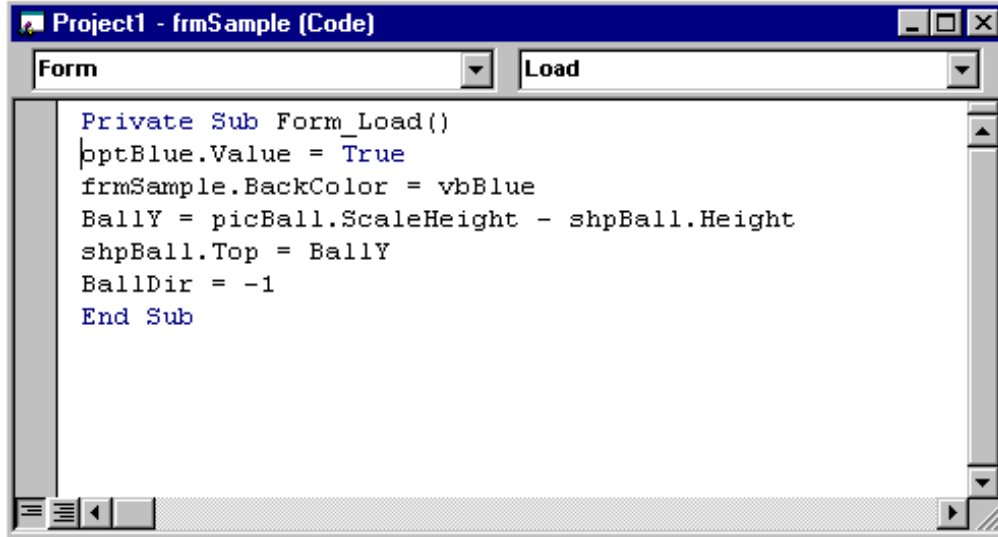
Kod Penceresi(Code Window)

Yeni pencereye bakalım. Tekrar hatırlayalım, Visual Basic event-driven (olay- güdümlü)'dür ve bir olay fark edildiğinde, project(proje) doğru event procedure(olay prosedür)'üne gider. Olay Prosedürler, Bilgisayara, bir event(olay)'a ne cevap vermesi gerektiğini söyler. Bunlar gerçek bilgisayar programlmasının (BASIC dili kullanılarak) olduğu yerlerde olur. Code Window (Kod Penceresi)'ndeki olay prosedürlerine bakalım. Kod penceresini göstermenin pek çok yolu vardır. Bir yolu, proje penceresinde bulunan uygun butonu bulmaktır. Bir diğeri ise Ana Menü'de bulunan View ve daha sonra Code seçeneklerini tıklamaktır. Veya bir alternatif olarak, F7 fonksiyon tuşuna basınız. Sample projesi için kod penceresini bulunuz.

VB4:



VB5, VB6 :



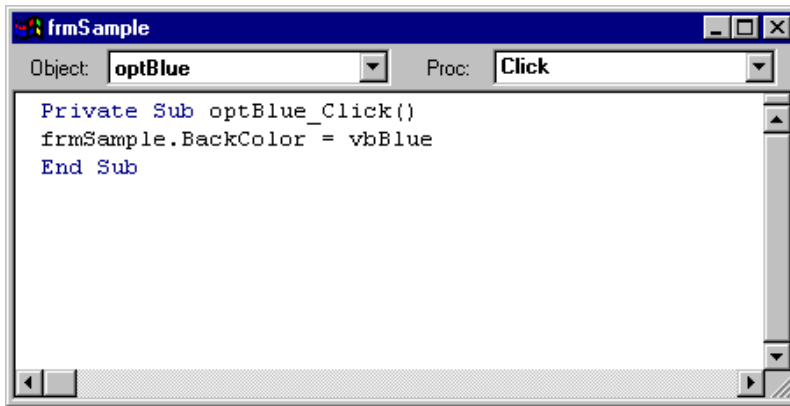
Kod Penceresinin en üstünde iki tane yan yana kutu vardır, object(nesne) (veya kontrol) listesi ve prosedür listesi. Nesne listesi, özellikler(properties) penceresinde bulunan liste ile benzerdir. Form üzerinde bulunan bütün nesneleri, isimleri ile listeler.

Bir nesne veya kontrol için, listede bir isim seçilmişse, prosedür listesi bu kontrol için mümkün olan bütün olay prosedür'leri(event procedures)'leri gösterir. Görüldüğü gibi, kod penceresi, Form kontrolü için Load olay prosedürü göstermektedir. (Eğer nesne listesinde Form Kontrolü seçili değilse, aşağı-sürükleyerek okunu tıklayarak Form kelimesi çıkana kadar kaydırın ve bunu seçin.). Proje ilk olarak bilgisayara yüklendiğinde, load(yükle) olayı olur. Nesne ve prosedür liste kutuları altında bu olay prosedürü için gerçek BASIC kodları yer almaktadır.

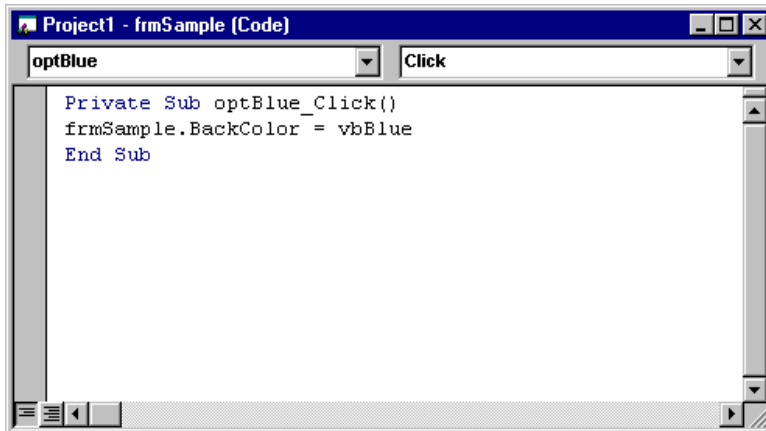
Bu kod muhtemelen şimdi size bir yabancı dil gibi görünebilir fakat endişelenmeyin – anlayacaksınız ! Takip eden derslerde, BASIC öğrenmeye başlayacaksınız ve böylesi kodları okumak size çok kolay gelecek.

Prosedür liste kutusunun yanındaki aşağı-sürükleyerek okunu tıklayın. Form kontrolü için bütün diğer olay prosedürlerine dikkat edin. Şanslıyız, çünkü bütün bu prosedürler için BASIC kodu yazmak zorunda değiliz. Yalnızca projemiz çalışırken (run) olmasını beklediğimiz olaylar için kod yazacağız. Nesne listesinde aşağı-çek okunu tıklayın. optBlue 'yu nesne olarak seçin.

VB4:



VB5, VB6:



optBlue isimli kontrol için olay prosedürü Click 'dir. Eğer hiç BASIC bilmiyorsanız bile, burada neler olduğunu çıkartmaktasınız. Kontrollere isimler verirken dikkatli olduğumuzdan, fark etmeniz gerekir, bu kontrol bir option (seçenek) butonudur (ufak daireli olan) ve onun yanında Blue kelimesi yer almaktadır (seçenek butonunun yanındaki kelime onun Caption(Başlık) özelliğidir). Yani, bu olay prosedürü, biz ne zaman Blue seçenek buttonunu tıklarsak çalışır. Prosedür, yalnızca bir satırdan oluşan bir talimata sahiptir (ilk ve son satırları şimdilik bir kenara bırakın) :

Bu BASIC kod'u satırı, frmSample adı verilen kontrolün BackColor property (özellik)'inin mavi renge ayarlanmasını söylemektedir (vbBlue kelimesi ile temsil edilen).

Kolay değil mi ? Gerçekten de, çoğunluk BASIC kodları kolay anlaşılır.

Nesne listesinden diğer nesneleri seçin ve karşılık gelen olaya, BASIC koduna bakın. Kodun en üst satırından başlayın (ve yine, en üstte yer alan başlık satırını göz önüne almayın) ve çalışın.

Hiç BASIC bilmemenize rağmen neler olup bittiğini anlayabiliyormusunuz ?

Düşüncem, sizin pek çok durumda yapabileceklerinizi, kendinizin bulacağı yönündedir. BASIC kodu yazmak, temelde, pek çok detaya dikkat etmektir. Pek çok kısmı için çok mantıklı ve açıktır. Ve, şimdi artık kendi kodunuzu yazmaya başlayabilirsiniz!

Özetle;

Bu ikinci derste, Visual Basic Ortamının parçalarını ve bu ortam içerisinde nasıl hareket edeceğimizi gördük. Aynı zamanda properties (özellikler) ve event procedures (olay prosedürleri) gibi bazı önemli terimleri öğrendik. Şimdi artık ilk Visual Basic projenizi yapılandırabilecek durumdasınız.

Diğer derste, bir form üzerine kontrolleri nasıl yerleştireceğinizi, onlarla nasıl hareket edeceğinizi ve onların nasıl olmasını istiyorsanız, o şekilde hazırlamayı öğreneceksiniz. Event Procedure(Olay Prosedürler)'e nasıl BASIC kodları koyacağınızın bütün önemli adımlarını hep birlikte öğreneceğiz !

3. İlk Visual Basic Projeniz

Tekrar Gözden Geçirme ve Bir Önbakış

İlk iki derste forms(formlar), controls(kontroller), properties(özellikler), ve event procedures(olay prosedürler)'i öğrendiniz.

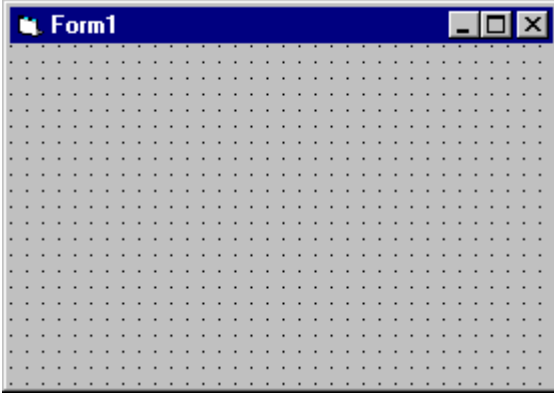
Bu derste, şimdiye kadar öğrendiğiniz bütün bu bilgileri, ilk basit Visual Basic Projenizi oluşturmak üzere kullanacaksınız. Yine bu derste, bir proje yapılandırmanın adımlarını, form üzerine kontrolleri nasıl yerleştireceğinizi, bu kontroller için properties(özellikler)'i nasıl düzenleyeceğinizi ve kendi olay prosedürlerinizi nasıl yazacağınızı (biraz BASIC kullanarak) öğreneceksiniz.

Bir Visual Basic Projesi Yapılandırılmanın Adımları

Bir Visual Basic Projesinin yapılandırılmasında, üç temel adım bulunmaktadır:

- 1. Form üzerine control(kontrol) 'leri yerleştirin (veya çiziniz).*
- 2. Bu control(kontrol)'lere properties(özellikler) atayın (değerler verin).*
- 3. Bu kontroller için, event procedure(olay prosedür)'ler yazın.*

Visual Basic'de, bu adımların her birisi, design(dizayn) mod'unda yapılmaktadır. Tekrar hatırlatalım, Visual Basic'de mode(mod), ana pencerenin başlık çubuğu üzerinde köşeli parantez içinde görünür. Visual Basic'i çalıştırın. Çalıştığında ortaya çıkması gereken formu bulun. Formu tıklayın. Aşağıdaki gibi görünmelidir:



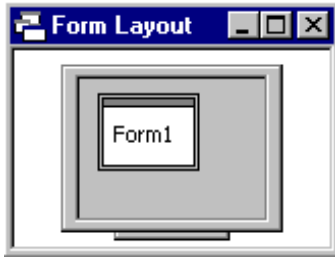
Projemizi yapılandırmadan evvel, birkaç 'Windows' tekniğini gözden geçirelim. Kontroller kullanarak, iki şey yapmanız gerekmektedir: (move)taşımak ve (resize)yeniden boyutlandırmak (form'un kendisinin de bir kontrol olduğunu hatırlayın!).

VB4:

- Bir VB4 projesinde form'u taşımak için, sol fare tuşu ile başlık çubuğu alanını tıklayın. Sol fare tuşunu basılı tutarak, form'u istenen yere taşıyın(move) (veya sürükleyin (drag)). Fare'nin tuşunu, formu istediğiniz yere taşıdıktan sonra bırakın. Proje başladığında, form'un ekranda alacağı pozisyon (yerleşim) budur.
- VB4'de bir form'u yeniden boyutlandırma(resize) için, fare işaretcisini form'un bir kenar veya köşesine getirin. Küçük bir 'double-arrow' (iki yönlü ok) görüldüğünde tıklayın ve bu kenar veya köşeyi sürükleyerek(drag) istenen büyüklüğe getirin.

VB5, VB6:

- Eğer VB5 veya VB6 kullanıyorsanız, formun kendisini hareket ettiremezsiniz-kendi penceresinde sabit bir pozisyonadadır (yerleşimdedir). Bu durumda pencerenin yerleştirilmesi Form Layout Window (Form Yerleşim Penceresi) kullanılarak yapılır:



Eğer ekranda Form Layout Window (Form Yerleşim Penceresi) görünmüyorsa, ana menü'den önce View seçeneğini daha sonra da Form Layout Window'u tıklayın.

Şimdi, ufak ekranda form'u tıklayın ve istediğiniz yerleşim pozisyonuna sürükleyin.

Bu, bilgisayarınızda, VB5 veya VB6 uygulamaları başladığında formun yerleşimini oluşturur.

- Dikkat edin, VB5 veya VB6'da, form'u yeniden boyutlandırmak için, form'un her kenarında ve her köşesinde 'sizing handles' ('boyutlandırma tutamaçları') bulunmaktadır.

Eğer fare işaretçisini bu tutamaçlardan birisinin üzerine doğru hareket ettirerseniz, küçük bir 'double-arrow' ('iki yönlü ok') belirecektir. Bu noktada, istediğiniz kenar ya da köşeyi, istediğiniz boyuta çekmek üzere tıklayarak sürükleyebilirsiniz.

Form'u taşımak ve yeniden boyutlandırmak için alıştırmalar yapın.

Bu alıştırmalar, aynı zamanda form üzerine kontrolleri yerleştirmek istediğimizde de gerekli olacaktır.

Control(Kontrol)'lerin Form Üzerine Yerleştirilmesi

Bir Visual Basic Projesinin yapılandırılmasındaki ilk adım, kontrollerin form üzerinde istenilen yerlere yerleştirilmesidir. Şimdi, bu noktada, projenizi yapılandırmanız için hangi kontrollerin gerekli olacağına karar vermeniz gerekmektedir.

Birçok kez, zaman tüketen bir çalışmadır. Size garanti ederim, kafanızdakini birçok kez değiştireceksiniz! Şimdilik, yalnızca, form üzerine kontrolleri nasıl koyacağımızın alıştırmalarını yapıyoruz.

Kontroller, Visual Basic'in Toolbox (Araç Kutusu)'ndan seçilir.

Bir form üzerine bir kontrol yerleştirmenin iki yolu vardır:

- 1. Toolbox Window (Araç Kutusu Penceresi) içinden, istenen kontrolü double-click(çift-tıkla)'nın. Kontrol default(varsayılan) boyutlarda oluşturulacak ve formun ortasına yerleştirilecektir.*
- 2. İsteddiğiniz kontrol'ü, Toolbox window içinde tek sefer tıklayın. Şimdi, fare işaretçisini form üzerinde hareket ettirin. Dikkat edin, işaretçi bir (+) işaretine dönüştü. Kontrolünüzü nereye koymak istiyorsanız, bu işaret yani (+)'yı kontrolünüzün sol üst köşesi olacak şekilde yerleştirin. Sol fare tuşunu tıklayın ve basılı tutun. Şimdi fare işaretçisini istediğiniz kontrolün sağ alt köşesine gelecek şekilde sürükleyin. Dikdörtgen biçiminde bir dış hat ekrana gelecektir. Bu dikdörtgen şeklindeki dış hat, eğer sizin istediğiniz kontrolün boyutunu temsil ediyorsa fare tuşunu bırakın. Kontrol ortaya çıkacaktır.*

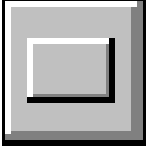
Kontrol, bir kere form üzerinde yer alınca, oraya nasıl yerleştirdiğiniz fark etmez.

Yine yeniden boyutlandırır ve taşıyabilirsiniz. Bir kontrolü move(taşımak) için, kontrolü seçmek üzere fare sol tuşunu tıklayın (boyutlandırma tutamaçlarını yeniden ortaya çıkacaktır). Onu yeni yerleşim yerine sürükleyin ve fare tuşunu bırakın.

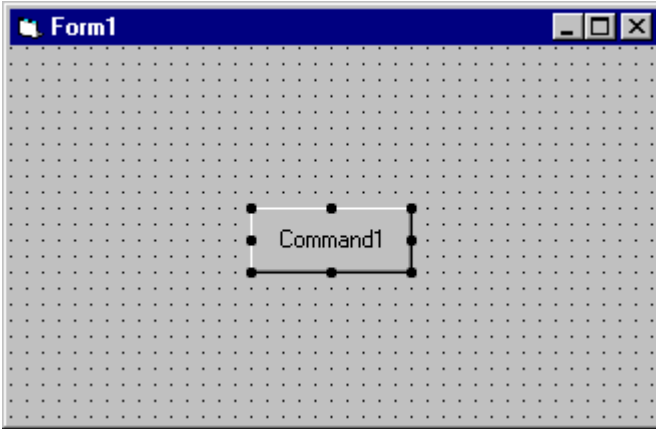
Resize(yeniden boyutlandırma) için sol fare tuşunu tıklayarak kontrolü seçin. Eğer fare işaretçisini bu boyutlandırma tutamaçlarından birisinin üstüne getirirseniz, küçük bir 'double-arrow' (iki yönlü ok) görünecektir. Bu noktada, karşılık gelen kenar veya köşeyi tıklayıp sürüklerseniz, seçilen kenar veya köşeyi istenen yere kadar getirebilirsiniz.

Örnek

Visual Basic'in çalıştığından, ekranda bir form olduğu kadar, Toolbox(Araç Kutusu)'nun da yerinde olduğundan emin olun (Eğer yerinde yoksa Ana Menü'de önce View daha sonra Toolbox seçeneğini tıklayın. Araç kutusuna gidin ve command button(komut butonu) kontrolünü bulun. Aşağıdaki gibi görünecektir:



Kontrolü çift-tıkla'yın(double-click). Form'un ortasında, ekrana gelmelidir:



Dikkat edin, boyutlandırma tutamaçları, buton etrafındadır. Bu, onun active(aktif) kontrol olduğunu göstermektedir. Formu tıklayın. Bu tutamaçlar gözden kaybolacaktır. Bu ise butonun artık aktif kontrol olmadığını göstermektedir.

Komut butonunununu tekrar aktif yapmak için yeniden tıklayın. Komut butonunu değişik yerlere taşıyın ve onu yeniden boyutlandırmayı deneyin.

Gerçek bir büyük buton, gerçek bir küçük buton, gerçek bir geniş buton ve gerçek bir kısa buton oluşturun. Komut butonunu form üzerinde değişik yerlere taşımayı deneyin.

İkinci bir yerleşim metodunu kullanarak, başka bir komut butonunu form üzerine koyalım. Tekrar geriye araç kutusuna gidin ve komut butonu kontrolüne tek tıklama yapın. İşaretciyi form üzerinde hareket ettirin. Bir (+) işareti göreceksiniz. İkinci metodu kullanarak, form üzerinde komut butonunu çizin: (+) işareti üst sol köşede iken fare sol tuşunu tıklayın ve istenen boyuta ulaşana kadar fare tuşunu basılı tutarak işareti sürükleyin. Fare tuşunu bırakın. Dikkat edin, hala siz bu ikinci komut butonunu taşıyabilir ve yeniden boyutlandırabilirsiniz.

Form üzerine kontrol yerleştirme ile ilgili bu iki yola da alışkın olmalısınız. Zamanla bir metodu diğer metottan daha kolay bulup, onu tercih eder hale geleceksiniz. Fakat daima, her ikisinin de nasıl kullanılması gerektiğini bilin. Bazı durumlarda; bu metotlardan yalnızca birisini kullanabileceksiniz. Kontrolleri form üzerine yerleştirme konusunda biraz zaman harcayın ve alıştırma yapın.

Etiketler, metin kutuları, seçenek butonları gibi diğer kontrolleri de kullanın. Form içinde bunları dolaştırın ve yeniden boyutlandırın.

Kontrollerinizi iyi sıralanmış gruplar halinde organize edin. Bu tecrübeler, Visual Basic projelerimizi yapılandırırken bize gerekecektir.

Bunlara ilaveten, form üzerinden bir kontrolü nasıl kaldırabileceğinizi de bilmeniz gerekmektedir. Kolay bir işlemdir! Kaldırmak istediğiniz kontrolü tıklayın, aktif kontrol haline gelecektir. Klavyeden Del (delete-sil) tuşuna basın. Kontrol kalkacaktır.

Bir kontrolü silmeden önce, onu gerçekten silmek istediğinizden emin olun!

Control Properties(Kontrol Özellikleri)'ni Düzenlemek

(Design Mode-Dizayn Mod'u)

İstediğiniz kontrolleri form üzerine yerleştirdiğinizde, bu sefer de bu kontrollere properties(özellikler) atamak, (değerler vermek) isteyeceksiniz. Hatırlayın! Properties(özellikler), bir kontrolün, form üzerinde nasıl görüneceğini belirler.

Özellikler, kontrolün boyutları, rengi, bir kontrolün "ne söylediği", form üzerindeki pozisyonu gibi şeyleri oluştururlar.

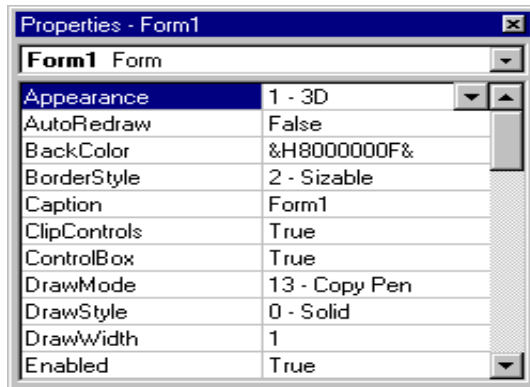
Bir kontrolü form üzerine yerleştirdiğinizde, bu kontrol Visual Basic tarafından verilen birtakım değerler alır. Form üzerinde kontrolü yerleştirip boyutlandırdığınızda, özellikle geometrik özellikler (boyutlarını ve yerleşimini içine alacak şekilde) düzenlenir.

Fakat birçok kereler, Visual Basic tarafından verilen bu varsayılan değerler kabul edilebilir değildirler ve onları değiştirmek isteyeceksiniz. Bu Properties Window(Özellikler Penceresi) yolu ile yapılır.

Eğer bilgisayarınızda Visual Basic çalışmıyorsa, şimdi başlatın. Eğer çalışıyorsa, File' ı daha sonra da New Project 'i tıklayın (eğer halihazır form ve projenin kaydedilip edilmeyeceği sorulursa cevap No(Hayır)'dır).

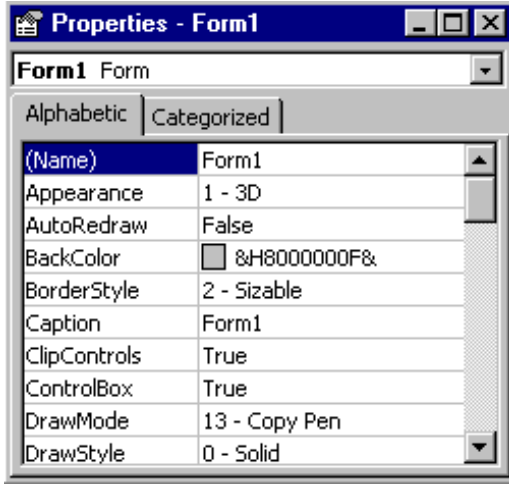
VB5 veya VB6' da ise ne çeşit bir projeye başlayacağınız sorulur: Standard EXE olarak cevap verin. Ekranda boş bir form bulunmalıdır. Properties Window 'u bulun (eğer yoksa <F4> 'e basın):

VB4:



VB5, VB6:

Eğer Categorized seçili görölüyor ise, Alphabetic olanını seçin.



Hatırlayalım, Properties Window'un üzerinde control list(kontrol listesi) bulunmaktadır. Bu ise bize, form üzerinde hangi kontrollerin bulunduğunu söylemektedir. Burada listede ise, yalnızca bir kontrol vardır ve bu da formun kendisidir.

Haydi, form özelliklerinin bazılarına bakalım:

İlkin, form ne büyüklükte ? Bütün kontroller dikdörtgen şekildedir ve dört özellik bu dikdörtgenin boyutlarını belirler. Özellikler listesini kaydırma çubuğu ile kaydırın ve Height(yükseklik) özelliğini bulun. Bu özellik formun yüksekliğidir ve twip adı verilen bir birim ile değer bulur. Bir cm içerisinde 567 twip bulunmaktadır. Yani yüksekliği (Height) 567 'ye bölerek, formun yüksekliğinin cm olarak ne kadar olduğunu hesaplayabilirsiniz. Benzer şekilde Width(Genişlik) özelliği, formun genişliğini twip birimi ile verir. Formun boyutlarını değiştirin ve buna bağlı olarak yükseklik ve genişlik değerlerinin nasıl değiştiğine dikkat edin.

Left(Sol) özelliği, formunuzun sol kenarının monitör ekranının sol kenarından ne kadar uzakta olduğunu söyler. Top(üst) özelliği, formunuzun üst kenarının monitör ekranının üst kenarından ne kadar aşağıda olduğunu söyler.

Formunuzu hareket ettirin ve bu özelliklerdeki değişimleri izleyin (VB5 ve

VB6' da ise formu, Form Layout Window (Form Yerleşim Penceresi) üstünde hareket ettirmelisiniz).

Veya Left veya Top özelliklerini tıklayarak, buralara yeni değerler verebilirsiniz. Bundan sonra da formun ekran üzerinde nasıl hareket ettiğini izleyin. Böylece dört özellik, Left, Top, Width, ve Height formun bilgisayar üzerinde tümü ile yerleşim ve boyutlarını tanımlar.

Kaydırma çubuğu ile BackColor(Arka Renk) özelliği üzerine gelin. Burada gördüğünüz değer, formun background(artalan) renk değerini göstermektedir.

Bu özellik için listelenen değer muhtemelen &H000000F& 'dir. Belki farkında değilsiniz, fakat bilgisayar dili ile bu gri bir renktir. İleri derslerde renkleri düzenlemenin diğer yollarını da inceleyeceğiz. BackColor özelliğini şimdi değiştirmek için, BackColor 'u tıklayın ve daha sonra aşağı açılan liste okunu tıklayın (Eğer VB5 veya VB6 kullanıyorsanız, Palette Tab'ını tıklayın.) Çeşitli renklerin bulunduğu bir renk paleti ortaya çıkacaktır. Yeni bir renk seçin ve sonucunu izleyin. Kaydırma çubuğunu Caption(Başlık) özelliği üzerine getirin. Bu özellik, formun başlık çubuğu üzerinde ne yazısı bulunduğunu belirler. Caption(Başlık)'ı tıklayın ve daha sonra özellik penceresinden bu satırın sağ tarafına bir şeyler yazın. Dikkat edin yeni Caption(Başlık), form'un başlık çubuğu üzerinde de görünmektedir.

Property(özellik)'lerin kontrollerini ayarlamak, işte budur !

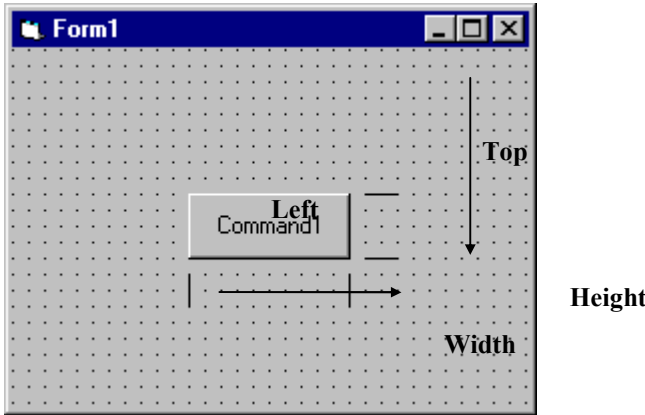
İlkin kontrol listesinden ilgilendiğiniz kontrolü seçin. Daha sonra kaydırma çubuğu ile kaydırarak özellikler boyunca gezinin ve değiştirmek istediğiniz özelliği bulun. Bu özelliği tıklayın. Özellik yeni bir değer girilerek değiştirilebilir (geometrik değerleri veya başlık özelliği gibi) veya önceden tanımlanmış bir listeden seçim yaparak gerçekleştirilebilir (renk değerleri gibi aşağı ok vasıtası ile açılan liste gibi).

Örnek

Yeni bir Visual Basic Projesine başlayalım. Bir form görünecektir. İstenilen yerleşim ve boyutlara getirmek üzere, formu hareket ettirin ve yeniden boyutlandırın.

Left, Top, Width ve Height özelliklerini kontrol edin. BackColor özelliğini ayarlayın. Caption özelliğini ayarlayın. Form üzerine bir komut butonu koyun. Komut butonunu ölçülendirin ve yerleşimini ayarlayın.

Komut butonu özelliklerine bakalım. Özellikler penceresinin kontrol listesinden command button'ü seçin. Form gibi komut butonu da dikdörtgendir. Width özelliği twip birimi ile genişlik değerini, Height özelliği ise twip birimi ile yükseklik değerini verir. Formlar dışındaki kontroller için Left(Sol) ve Top(Üst) özellikleri biraz farklıdır. Form olmayan bir kontrol için, Left değeri kontrolün sol kenarının, formun (ekranın değil) sol kenarına olan uzaklığını verir. Yani, twip birimi ile form üzerinde kontrol pozisyonunu verir. Benzer şekilde Top değeri, kontrolün üst kenarının pozisyonunun twip birimi ile formun (ekranın değil) form'un üst kenarına (form'un üst kenarı başlık çubuğunun alt tabanı olarak tanımlanır) olan uzaklığını verir. Tek bir komut butonu için, bu özellikler aşağıdaki gibidir:



Komut butonu için diğer önemli bir özellik ise Caption(Başlık) özelliğidir.

Buton üzerinde görünen yazı Caption'dır. Bu yazı, bu buton tıklandığında ne olacağını ifade etmelidir. Komut butonunun Caption özelliğini değiştirin. Komut butonu için bir BackColor özelliği listelenmiş olsa bile, bu değiştirilemez.

Form üzerine birkaç tane daha komut butonu koyun. Bunları taşıyın ve yeniden boyutlandırın. Bunların Caption özelliklerini değiştirin.

Properties Window(Özellikler Penceresi) 'nde, bir kontrolden diğerine geçiş yapmasını öğrendik. Kontrol listesinden aşağı oku tıklayabilir ve listeden seçtiğimiz değeri alabiliriz. İstenilen bir kontrole, listelenen özellikler içerisinde, geçmenin kısa(kestirme)

bir yolu, form üzerindeki kontrolü basitçe tıklamak ve onu active(aktif) kontrol yapmaktır. Komut butonlarından birisini tıklayın. Dikkat edin özellikler penceresindeki seçilmiş kontrol, bu seçtiğimiz yeni kontrole geçti. Başka bir butonu tıklayın, değişmeye dikkat edin. Formu tıklayın. Seçilen kontrol form oldu. Kendi Visual Basic projelerinizi hazırlarken, bu kestirme metodu; bir kontrolden diğerine kolayca geçişi, çok faydalı bulacaksınız !

Kontrolleri İsimlendirme (İsim Verme)

Herhangi bir kontrol için en önemli özellik, onun Name(İsim)' i dir. Öneminden dolayı bu kısmı ayrıca ele alacağız. Bir kontrole isim verirken, iki parça bilgi tanımlamak isteyeceğiz: kontrol'ün type(tip)' i ve kontrol'ün purpose(amacı). Böyle isimlendirmeler programlama çalışmalarımızı çok daha kolaylaştıracaktır.

Visual Basic Programcı Topluluğunda kontrolleri isimlendirmek için bir kural geliştirilmiştir. Kontrol isminin üç harfi (prefix – önek olarak adlandırılır) kontrol tipini belirler. Bazı önekler aşağıdadır (daha fazlasını, ilerleyen dersler boyunca göreceğiz):

<u>Kontrol</u>	<u>Prefix(Önek)</u>
Form	frm
Command Button(Komut Butonu)	cmd
Label(Etiket)	lbl
Text Box (Metin Kutusu)	txt
Check Box (Onay Kutusu)	chk
Option Button (Seçenek Butonu)	opt

Kontrol ismi önekinden sonra, kontrolün ne yaptığını belirten bir isim seçeriz (bu; önekin sona erdiğini göstermek üzere, genellikle büyük harfle başlar). Komple bir kontrol ismi 40 karaktere kadar uzunluğa sahiptir. İsim bir harf ile başlamalıdır (bu kullanılan önek dikkate alınarak yapılır) ve yalnızca büyük ya da küçük olmak üzere harfler, sayılar ve (_) karakterlerini içerir. 40 karaktere kadar uzunlukta kontrol isimleri oluşturma hakkınız olsa bile, isimleri mümkün olduğunca, anlamlarını da kaybetmesine izin vermeden kısa tutun. Bu, sizi, klavyeden uzun uzun isim yazma zahmetinden kurtarır.

Size bir fikir vermek üzere, isimlerin nasıl seçileceği konusunda bazı örnek kontrol isimlerine göz atalım. Bu isimler, birinci ve ikinci derslerde baktığımız Sample projesinde kullanılan isimlerdir.

Örnekler:

frmSample - Sample projesi için form

cmdBeep - Bip(beep) sesi verdiren komut butonu(Command button)

lblPick - Seçilen sayıyı gösteren etiket(Label)

*optBlue - Background(artalan) rengini maviye çeviren Option button
(Seçenek Buton)'u*

chkTop - Oyuncak top'u gösteren veya gizleyen Onay Kutusu(Check box)

Bunlar, size, kontrol isimlerinin nasıl seçileceği konusunda bir fikir vermelidir. Bir programcı olarak çalışmanızı çok daha kolay gerçekleştirmenizi sağlayacaktır.

Run Mod'unda, Properties(Özellikler)'i Düzenlemek

Uygun kontrol isimleri kullanmanın önemini belirtmek için, Visual Basic'te ortak bir amaca bakalım. Bir Visual Basic Projesini geliştirme adımlarından birisinin Design(dizayn) mod'unda kontrol özelliklerini oluşturmak olduğunu gördük. Siz aynı zamanda, projeniz run(çalışma) mod'unda iken properties(özellikler)'ini değiştirebilir veya oluşturabilirsiniz. Örneğin, Sample projesinde, bir option button(seçenek buton)'una tıkladığınızda, formun BackColor özelliği değişecektir. Run(çalışma) modunda bir özelliği(property) değiştirmek için, bir satır BASIC kodu kullanmamız gerekmektedir. (İlk BASIC satırını öğrenmek üzeresiniz !). Bu kod için format (yapı -şekil) aşağıdaki gibidir:

ControlName.PropertyName = PropertyValue

Kontrolün ismi, bir nokta (küsurat-ondalık noktası gibi), değiştirdiğimiz özelliğin ismi, (özellikler penceresi içinde bulunur), bir eşittir işareti (bir atama işlemcisi olarak adlandırılır) ve yeni bir değer yazdık. Böylesi bir format dot notation-Nokta Notasyon olarak adlandırılır.

Sample Projesi Form'unun artalan rengini maviye dönüştüren kod:

```
frmSample.BackColor = vbBlue
```

Dikkat edin, hiç BASIC bilmemenize rağmen, uygun isim kullanılması, bu kod satırını çok anlaşılabilir bir hale getirmiştir. Sample form'un artalan(background) rengini maviye ayarlanmasını söylemektedir.

Olay Prosedürlerinde Kontrol İsimleri Nasıl Kullanılır

Uygun kontrol ismi vermenin önemi bir kez daha event procedures (olay prosedürleri) yazarken ortaya çıkar (diğer derslerde ele alınacaktır). Bir form üzerine bir kontrol koyduğunuzda, kontrol yapan olay prosedürlerinin tümü, projenize eklenecektir. Bu event procedures(olay prosedürleri)'ni, code window (kod penceresi)'nde göreceğiz. Bu olay prosedürleri için yapı:

```
Başlık satırı : Private Sub ControlName_EventName()
```

```
[BASIC kodu buradan devam eder]
```

```
Alt(taban) satırı : End Sub
```

Başlık satırının kontrol ismi kullandığına dikkat edin. Böylece, uygun isimlendirme ile bizler kolaylıkla her olay prosedürünü tanımlayabiliriz

Bir örnek olarak, yine Sample programını kullanarak, optBlue kontrolü için Click olay prosedürü:

```
Private Sub optBlue_Click()  
frmSample.BackColor = vbBlue  
End Sub
```

Fark etmemiz gereken şey, kullanıcının optBlue option buttonunu çalıştıracak kod budur.

Uygun isim verme, olay prosedürlerin çok kolaylıkla tanımlanmasını ve okunmasını sağlar. Yine, bu, sizin bir programcı olarak, işinizi çok daha kolay yapmanızı sağlar. Şimdi, ilk event procedure(olay prosedür)'ümüzü yazalım.

Olay Prosedürü (Event Procedure) Yazmak

Bir Visual Basic Uygulaması yapılandırırken üçüncü adım, form üzerinde bulunan kontroller için event procedures(olay prosedür)'leri yazmaktır. Bir olay prosedürü yazmak için, code window(kod penceresi)'ni kullanırız.

Kod penceresini projenizde ekrana getirmek için gerekli yolları gözden geçirelim. Bu adım, gerçekten BASIC kodları yazmamız gerektiğinde veya bilgisayar programlaması yaparken gerekmektedir. Şimdiye kadar pek BASIC öğrenmediniz fakat, olay prosedürlerini bulma ve kod yazmanın prosesini öğrendiniz.

Henüz şimdi belirttiğimiz gibi, bir form üzerine bir kontrol yerleştirdiğimizde bu kontrol ile ilintilendirilmiş(bağlantılanmış) olay prosedürleri, projenin bir parçası haline gelecek ve bunlara code window(kod penceresi) vasıtası ile ulaşılabilecektir.

Her kontrol birçok olay prosedürüne sahiptir. Her prosedür için BASIC kodu yazmak zorunda değilsiniz-yalnızca bilgisayarın cevap(karşılık) vermesini istedikleriniz için yazmalısınız. Bir kere bir olayın 'kodlanması' gerektiğine karar vererseniz, bu olay prosedüründe ne gerçekleşmesi gerektiğine karar verin ve bu isteklerinizi gerçek BASIC kod satırlarına çevirin. Daha evvel görüldüğü gibi, her olay prosedürü için format(yapı-şekil) aşağıdaki gibidir:

Başlık Satırı : Private Sub ControlName_EventName()

[BASIC kodu buradan devam ediyor]

Alt(taban) satırı: End Sub

'Private Sub' kelimeleri , bunun bir Subroutine (prosedür için başka bir kelime) ve bunun yalnızca forma Private(Özel) olduğunu belirtmektedir (yalnızca bu form tarafından kullanılabilir – şimdilik bunun ne anlama geldiği konusunu dert etmeyin!).

BASIC kodu geliştirme, bir Visual Basic Uygulamasının yaratıcı kısmıdır ve aynı zamanda, nerede ihtiyaç duyarsanız çok kesin olmak zorundadır. Yanlış heceleme ve yazım, işaretlemelerdeki hatalar ve işlemcilerin yanlış veya eksik olması programınızı çalıştırılmaz hale getirir.

Bir bilgisayar programı yazmanın kesinlik gerektirdiğini göreceksiniz.

Olay prosedürleri yazmanın yöntemleri şunlardır:

- *Hangi olaylara bilgisayar tarafından karşılık verilmesi istediğinize karar verin.*
- *Bu karşılığın ne olacağı konusunda karar verin.*
- *Bu karşılığı BASIC koduna çevirin.*
- *Kod penceresindeki olay prosedürünü bulun.*
- *BASIC kodunu yazın.*

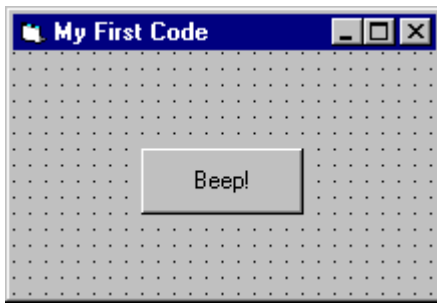
Ve, bu en iyi şekilde örnekle anlatılacak bir yöntemdir:

Örnek

Eğer Visual Basic bilgisayarınızda çalışmıyorsa , çalıştırın ve yeni bir projeye başlayın.

- *Form üzerine tek bir komut butonu yerleştirin.*
- *Formun Name property(özelliğini) frmFirstCode olarak verin.*
- *Caption property(özelliğini) My First Code(Benim İlk Kodum) olarak verin.*
- *Komut butonunun Name özelliğini cmdBeep olarak verin.*
- *Komut butonunun Caption özelliğini Beep! Olarak verin.*

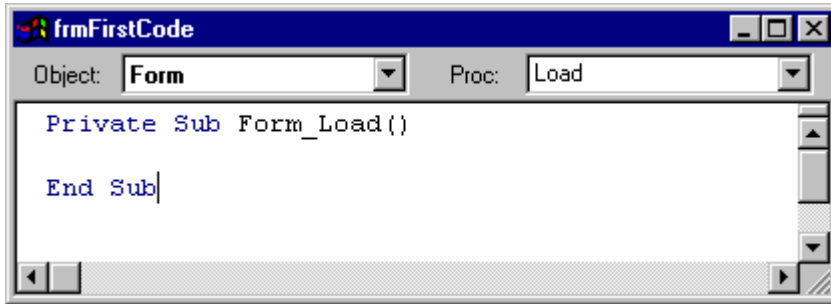
Bu noktada , dizayn prosesinde, form'unuz şu şekilde görünmelidir:



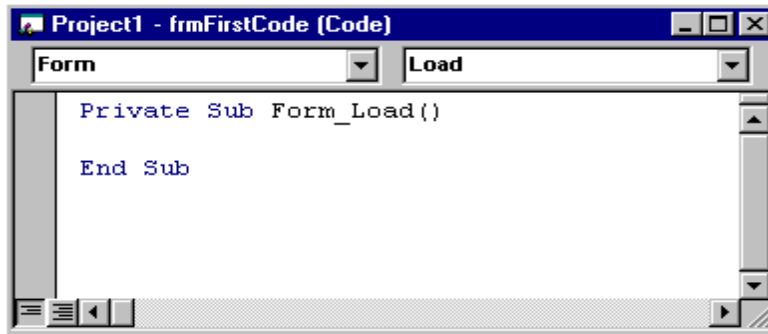
Tek bir olay prosedürü yazmak istiyoruz - prosedür, komut butonunun Click(tıklama) olayına karşılık vermesini, bu butonu tıkladığımızda, bilgisayarın beep(bip) sesi vermesini istiyoruz.

Kod pencersini ekrana getirin (<F7> 'ye basmak yollardan birisidir):

VB4:



VB5, VB6:



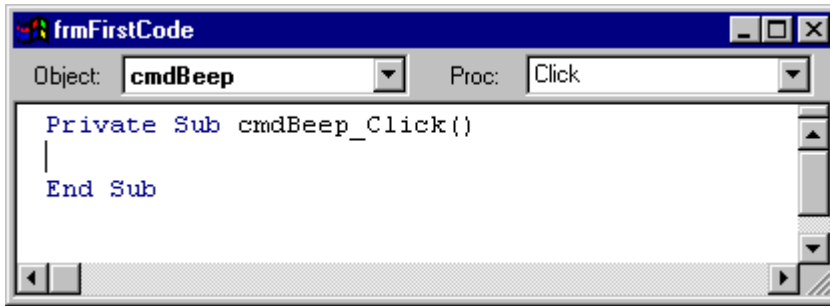
Sizin kod pencereniz bu şekilde görünmeyebilir. Visual Basic ortamında

full-module adı verilen bir display(görünme) seçeneği vardır. Bu seçenekte; olay prosedürleri burada görüldüğü gibi ayrı ayrı değil, biri diğerinin arkasından listelenmiştir.

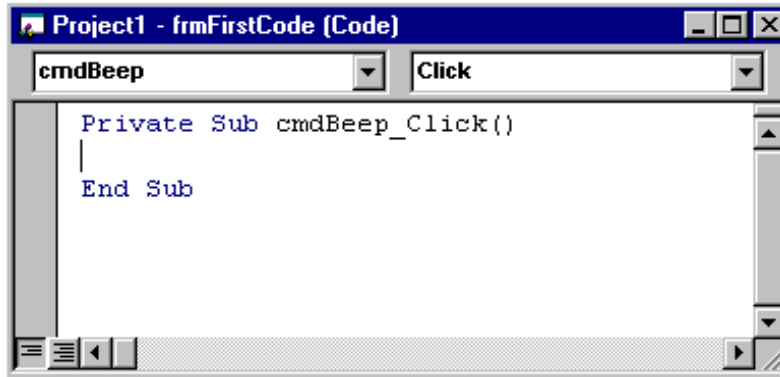
Eğer Visual Basic ortamınız(environment) full-module view ' de ise değiştirmeniz gerekmektedir, çünkü bu kitapta onu kullanmayacağız. Full-module view'den çıkmak için, Tools'ı ve daha sonra Options'u tıklayın. Çıkan pencerede Editor'ü tıklayın ve Full-Module View seçeneğinin yanındaki kutuda onay işareti bulunmadığından emin olun. Eğer varsa , kutuyu tıklayarak işareti kaldırın. Eğer cmdBeep nesnesi, Object(Nesne) listesinde görünmüyorsa, bu listede bulunan aşağı ok ile açılan listeyi tıklayın ve cmdBeep (komut buton)'unu seçin.

Kod penceresi şimdi aşağıdaki gibi görünmelidir:

VB4:



VB5, VB6:



Dikkat edin, cmdBeep buton'u için Click prosedürü şimdi ekrana geldi. Birçok kereler istediğiniz prosedürü bulmak için Procedures listesini kullanacaksınız-bu sefer yalnızca bir tanesi ekrana geldi

(prosedürler alfabetik olarak listelenmiştir). Burası bizim bilgisayara 'bip' sesi verdireceğimiz kodları yazacağımız yerdir.

Kod penceresi bir kelime işlemcisi gibi davranır. Pencere içine metin yazılabilir, pek çok normal düzenleme özellikleri kullanabilir. Örnek kes, yapıştır, kopyala, bul ve değiştir gibi. Daha tecrübeli bir programcı olmaya başladığınızda, kod penceresini kullanırken daha rahat olacaksınız. Başlık ve dip satırları arasındaki alanı tıklayın.

Tek bir satır yazın:

Beep

Bu bir BASIC talimatı olup, bilgisayara bip sesi vermesini söylemektedir.

Şimdi ilk BASIC kod satırını yazmış bulunuyorsunuz !

Projeniz şimdi artık çalıştırılmaya hazırdır(run). Projeyi Run(çalıştır)'ın

(araç çubuğu üzerindeki Start butonunu tıklayın veya <F5>'e basın). Size bazı dosyaları saklayıp saklamayacağınız sorulabilir: Şimdilik No(Hayır) ile cevap verin veya Cancel(iptal)'i tıklayın.

Form şu şekilde belirecektir:



Üzerinde 'Beep!' yazan komut butonunu tıklayın. Bilgisayar bip sesi vermelidir. cmdBeep kontrol'ü üzerinde bir Click event(olay)'ına sebep oldunuz. Bilgisayar bunu fark edecek ve cmdBeep_Click olay prosedürüne gidecektir.

Buradaki kod satırı(Beep) yorumlanacak ve bilgisayar bip sesi verecektir. Projenizi durdurun. Kod penceresine gidin ve cmdBeep_Click olayını bulun. Beep satırından sonra, şu satırı ekleyin:


```
frmFirstCode.BackColor = vbBlue
```

Aynen yukarıda yazıldığı gibi yazdığınızdan emin olun. Hatırlayın, bilgisayar programları kesin olmalıdır. Projeyi yeniden çalıştırın(Run). Komut butonunu tıklayın. Kontrol, olay prosedürü ve BASIC kodu ile olan ilişkisini açıklayın. Projenizi durdurun.

Özetle;

Şimdi ilk ve komple bir Visual Basic projesini tamamladınız. Bir uygulamayı yapılandırırken şu üç adımı takip ettiniz:

- 1. Form üzerine kontrolleri yerleştirin.*
- 2. Kontrol özelliklerini atayın(verin).*
- 3. Kontrol olay prosedürleri yazın.*

Aynı adımları, ister buradaki gibi çok basit, isterse de çok karmaşık bir projeyi yapılandırırken aynen takip edeceksiniz.

Şimdi, bu adımları bilerek, Visual Basic Toolbox(araç kutusu) üzerinden çalışmaya her kontrolün ne yaptığını öğrenmeye hazırsınız. Programlarınızı yazarken, size yardımcı olması için, BASIC dilinin elemanlarını öğrenmeye başlayabilirsiniz. Proje geliştirmenize yardımcı olmak için, Visual Basic Ortamının yeni özelliklerini öğrenmeye başlayabilirsiniz. Her bir sonraki derste bazı yeni kontroller, bir miktar daha BASIC ve Visual Basic hakkında daha fazla yeni şeyler öğreneceğiz.

Herkes İçin
Visual Basic

4. Proje (Tasarımı)Dizayn'ı, Formlar ve Komut Butonları

Tekrar Gözden Geçirme ve Bir Önbakış

Şimdiye kadar, bir Visual Basic projesinin parçalarını ve bir proje yapılandırılması ile ilgili şu üç adımı öğrendiniz:

- 1. Form üzerine kontrolleri yerleştirin.*
- 2. Kontrol özelliklerini düzenleyin.*
- 3. İstenilen olay prosedürünü yazın.*

Visual Basic ile yapılandırmak istediğiniz projeler hakkında bir fikriniz var mı ? Eğer böyle ise, çok güzel ! Bu dersten başlayarak, kendi programlama becerinizi geliştirmeye başlayacaksınız. Geldiğimiz her yeni derste, Visual Basic ortamının bazı yeni özelliklerini öğreneceğiz. Birtakım yeni kontroller ve BASIC dilinin bazı yeni elemanları bunlardan bazılarıdır. Bu derste, proje dizayn(design)'ı, form, komut butonu kontrolleri ve komple bir projenin nasıl yapılandırıldığını öğreneceksiniz.

Proje Design(Dizayn-Tasarım)'ı

Şimdi, Visual Basic kullanarak projeler geliştirmeye başlamak üzeresiniz!

Size, yapılandırmanız için bazı projeler vereceğiz ve belki de sizin kendi projelerinizi oluşturmak için baz fikirleriniz olacak. Her ne şekilde olursa olsun, bilgisayar programları olarak sonuçlanan fikirleri görmek, hoş ve etkileyici olacak ! Fakat, bir projeye başlamadan evvel, ne yapmaya çalıştığınız konusunda biraz durup düşünmeniz iyi olacaktır. Uygun proje dizayn'ı size çok fazla zaman kazandıracak ve sonuçta, ileride daha iyi bir proje olarak sonuçlanacaktır!

Uygun proje dizayn'ı gerçekten zor değildir ! Bir proje yaratmanın ana fikri; kullanımı kolay, kolay anlaşılır ve hatalardan arındırılmış bir proje yaratmaktır.

Anlamlı, değil mi? Projenizde yapmak istediğiniz herşey için bir süre durun ve düşünün ! Program ne türde bilgilere ihtiyaç duyuyor? Hangi bilgileri bilgisayar hesaplayacak? Bu bilgi topluluğunu sağlayın ve hangi kontrollere ihtiyaç duyacağınıza karar verin ! İyi bir

user interface(kullanıcı arabirimi) dizayn edin (arabirim, kontrollerin form üzerine yerleşimini de içine alır). Görünüm ve kullanım kolaylığını göz önüne alın. Eğer mümkünse, arabirimini, diğer Windows uygulamaları ile uyumlu olarak hazırlayın. Visual Basic kullanılarak hazırlanmış olanlar gibi, Windows tabanlı projelere aşinalık, her zaman iyidir.

Olay prosedürünüzü hazırlarken, BASIC kod'unuzu okunabilir ve kolay anlaşılabilir hazırlayın. Bu proje üzerinde daha sonra yapılacak değişiklikleri (veya sizin yapacağınız değişiklikleri) çok daha kolay hale getirir. Kabul edilen programlama kurallarına uyun - BASIC hakkında daha fazla şeyler öğrendikçe, bu kuralları da peyderpey öğreneceksiniz.

Projenizde hata bulunmadığından emin olun. Bu çok açık bir ifade gibi görünmesine rağmen, pekçok programlar hatasız değildir. Windows 95 içinde bile dolaşan birkaç yüz hata bulunmaktadır. Proje dizaynı konusunda bu birkaç ifadenin önemi şimdilik pek anlaşılmasa da, ileride bunların önemi ortaya çıkacaktır. Amaç, basit bir fikir, faydalı, açık yazılmış, kolay kullanımı olan ve değiştirilmesi kolay, hatalardan arındırılmış projeler hazırlamaktır.

Dikkatli bir ileriye yönelik planlama, bu amaca ulaşmanızda yardımcı olacaktır.

Bu derste yapılandırılan her proje için, proje dizayn yöntemi'nin iç yapısına bir bakış vermeye çalışacağız. Bir projeyi yapılandırırken, daima neden yaptığımızı ve ne yaptığımızı anlatmayı deneyeceğiz. Göz önüne aldığımız her şeyin açıklamasını size vereceğiz.

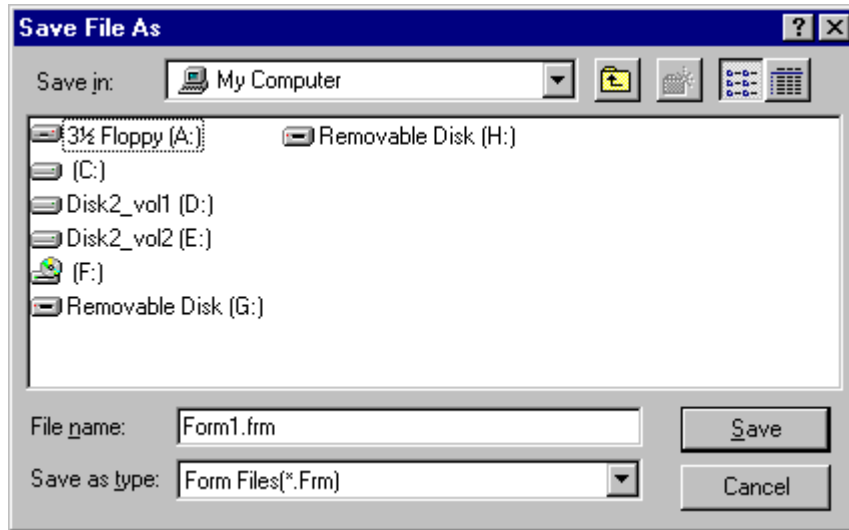
Bir Visual Basic Projesini Kaydetmek(Save)

1.derste , daha önceden kaydedilmiş(saklanmış) bir Visual Basic projesini nasıl açacağınızı, çalıştıracağınızı ve kapatacağınızı öğrendiniz. Fakat şimdiye kadar, hiçbir yerde, bir projenin ileriki kullanımları için nasıl saklanılacağından bahsetmedik. Şimdi artık kendi projenizi yapılandırmaya başlıyorsunuz ve onları nasıl saklayacağınızı bilmeye ihtiyacınız var.

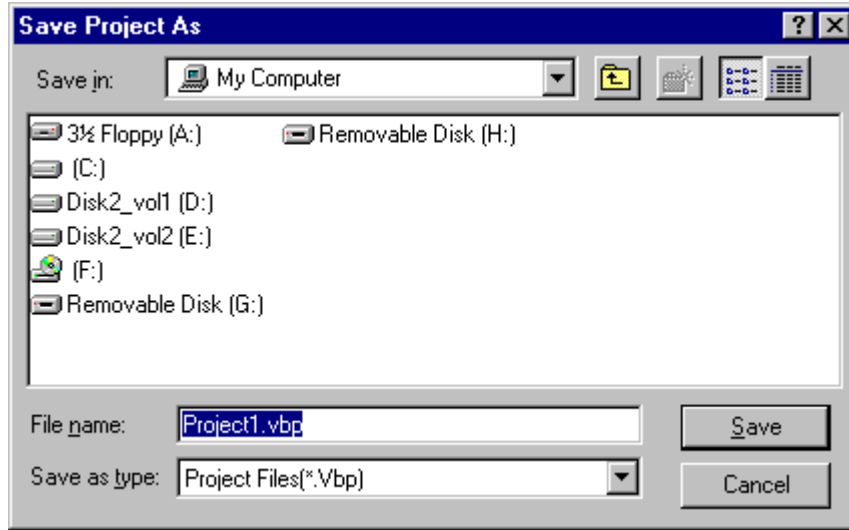
Bu gerçekten de çok kolaydır. Visual Basic ana penceresinde bulunan araç çubuğunu kullanacağız. Üzerinde bir disket resmi bulunan butona bakın. (Gittikçe kullanımı artan ve yazabilen CD sürücüler ile, insanların daha ne kadar bir disketin nasıl görüldüğünü hatırlayacaklarını düşünüyorsunuz ? – yeni Apple iMac’ın üzerinde bir disket sürücüsü bile yok !). Bu Save Project(Projeyi Kaydet) buton’udur:



Bu butonu tıklamak, onu ne zaman tıkladığınıza bağlı olarak, farklı sonuçlar veririr. Yeni bir proje üzerinde çalışıyorsanız ve daha evvel hiç saklamadıysanız, aşağıdaki pencere ortaya çıkacaktır:



Bu pencere, form'unuzu nereye kaydetmek istediğinizi ve ona ne isim vermek istediğinizi sormaktadır (tekrar hatırlayalım bu dosya frm uzantısına sahiptir). İstenen dizine gidin, form dosyasına istediğiniz bir isim verin (anlamalı bir isim) ve daha sonra Save tuşunu tıklayın. Formunuzu kaydettikten sonra, aşağıdaki gibi başka bir pencere belirecektir :



Bu pencere proje dosyanızı, nereye ve hangi isimle kaydetmek istediğinizi sormaktadır(vbp uzantılı bir dosya ismi). Yine bir dizin seçin(genellikle form dosyasını kaydettiğiniz dizindir), bir proje ismi girin(yine bu isim anlamlı olsun) ve Save'ı tıklayın.

Burada projeniz iki dosyada kaydedilmiştir: form dosyası ve proje dosyası.

Eğer daha önceden save(kaydet)'ilmiş bir proje üzerinde çalışıyorsanız ve Save Project buton'unu tıklarsanız, Visual Basic otomatik olarak form dosyası ve proje dosyasını aynı isimle, herhangi bir soru sormadan kaydedecektir. Üzerinde çalıştığınız projeyi ara sıra kaydetmeniz tavsiye edilir. Daima projelerinizi çalıştırmadan evvel veya Visual Basic'ten çıkarken kaydedin. Kaydedilmiş bir projeyi açmak isterseniz sadece pencere üstünde bulunan araç çubuğu üzerinde bulunan Open Project(Proje Aç) buton'unu tıklayın.

İstediğiniz proje dosyasını seçin ve Open(Aç)'ı tıklayın. Proje dosyası ve bu proje ile ilgili form dosyası açılacak ve form ortaya çıkacaktır.

On-Line Help (Çevrim-İçi Yardım)

Birçok kereler, Visual Basic ortamında çalışırken, herhangi birşey hakkında bir sorunuz olabilir. Belirli bir kontrol'ün ne yaptığı konusunda, belirli bir özelliğin ne için olduğu, bir kontrol'ün hangi olaylara sahip olduğunu veya belirli bir terim'in BASIC'te ne anlama geldiği konusunu merak edebilir veya şüpheye düşebilirsiniz. Böylesi askıda (havada) kalan durumlarda en iyi yol cevabını bilen birisine sormaktır. İnsanlar genellikle size yardımcı olmaktan mutlu olurlar-sizin öğrenmenize yardımcı olma fikrini severler.

Aynı zamanda cevabı bir kitapta bulmayı deneyebilirsiniz ve bu konuda bir sürü Visual Basic kitabı bulunmaktadır.

Veya, yardım almanın başka bir güzel bir yolu ise Visual Basic On-Line Help Sistem'ini kullanmaktır. Visual Basic'in de içinde olduğu birçok Windows uygulaması, kullanılabilir durumda, yardım dosyalarına sahiptir.

Visual Basic Help Sistem'ine ulaşmak için, ana menü'de bulunan Help(Yardım) seçeneğini ve daha sonra da Contents(İçerik)'i tıklayın. Bu nokta'da ihtiyaç duyduğunuz yardıma ait başlığı araştırabilir veya bütün başlıkları kaydırma çubuğu ile gezinerek aradığınız yardımı bulabilirsiniz. Visual Basic Help Sistem'i diğer bütün Windows Help Sistem'leri gibidir.

Eğer daha evvel herhangi bir on-line help sistemi kullandıysanız, Visual Basic'deki on-line help sistemini kullanmanız kolay olacaktır. Eğer şimdiye kadar hiç bir on-line help sistemi kullanmadıysanız, yardım için birisine sorun. Kullanımı oldukça kolaydır. Veya, Windows task bar üzerindeki Start'ı tıklayın ve daha sonra da Help'i seçin. On-line Help Sistem'ini kullanarak, bir on-line Help Sistem'inin nasıl kullanıldığını öğrenebilirsiniz.

Visual Basic on-line help sistem'inin gerçekten müthiş bir özelliği vardır; 'context sensitive' (bağlam duyarlı). Bu ne anlama geliyor?

Deneyelim !

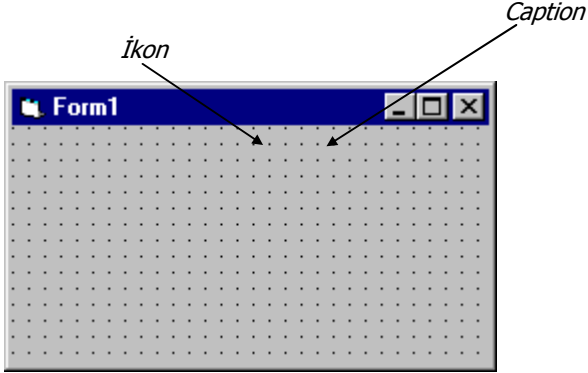
Visual Basic'i çalıştırın ve yeni bir projeye başlayın. Özellikler penceresine gidin. Form özelliklerini gösteren pencere kenarındaki kaydırma çubuğunu kaydırarak, BackColor kelimesini tıklayalım. Kelime belirginleşir. <F1> tuşuna basın.

BackColor özelliđi hakkında bir ekran dolusu bilgi görünür. Help sistem'i akıllıdır ! BackColor kelimesini belirginleřtirdiđinizi ve daha sonra <F1> tuřuna basıldıđını (<F1> daima yardım istendiđinde basılan bir tuřtur) görünce, sizin BackColor hakkında yardım istediđinizi anlar. Visual Basic alıřırken, ne zaman <F1> tuřuna basarsanız, program nerede alıřtıđınıza bakarak, bađlam-duyarlılıđınıza dayandırarak, ne hakkında yardım istediđinizi tespit etmeye alıřır. Bu özellikler penceresinde belirginleřtirilmiř bir kelime veya kod penceresindeki imle olabilir. Visual Basic ile alıřtıđıa, 'context-sensitive' yardım'ı ok faydalı bulacaksınız. Sorularınıza, pek ok kereler, abuk cevaplar alabilirsiniz. Yardım almak iin, Visual Basic on-line help sistem'ine gvenmeyi bir alışkanlık haline getirin !

Visual Basic ortamı iin, bu kadar yeni malzeme yeterli ! řimdi detaylı olarak, iki önemli kontrol'e bakalım: form'un kendisi ve komut butonu. Daha sonra BASIC dili ve komple bir proje yapılandırılma alıřmasına bařlayacađız.

Form Kontrol olarak, Form

Bizler form'un, bir Visual Basic Projesinin geliştirilmesindeki temel kontrol olduğunu öğrendik. Form olmadan, proje olmaz! Form Kontrolü için bazı önemli özellikler ve olaylara bakalım. Yeni bir projeye başladığınızda form aşağıdaki gibi görünecektir:



Özellikler

Bütün kontroller gibi, form da birçok (40'dan fazla) özelliğe sahiptir. Şanslıyız, yalnızca bunların bazılarını bilmek zorundayız. Göz önüne alacağımız özellikler :

<u>Özellik</u>	<u>Açıklama</u>
Name	Form'u tanımlamak için kullanılan isim. Form isimleri için kullanılan üç harfli önek frm'dir.
Caption	Form'un başlık çubuğu üzerinde görünen yazı
Icon	Form'un başlık çubuğu üzerinde görünen ikon(küçük şekil)'e referans (ikon yaratma'ya 7. derste bakacağız).

Left	<i>Form'un sol kenarından, ekranın sol kenarına kadar olan uzaklık</i>
Top	<i>Bilgisayar ekranının üst kenarından, form'un üst kenarına kadar olan uzaklık</i>
Width	<i>Twip birimi ile, form'un genişliği</i>
Height	<i>Form'un twip birimi ile yüksekliği</i>
BackColor	<i>Form'un background(artalan) rengi</i>
BorderStyle	<i>Proje çalıştığında, form ya boyutu değiştirilebilir (sizable) veya sabit tek(fixed single) boyutlardadır (boyut değiştirme, fare kullanılarak yapılır).</i>

Örnek

Bu özelliklere aşinalık kazanmak için, Visual Basic'i çalıştırın ve yalnız tek bir form'u olan yeni bir projeye başlayın. Top, Left, Height ve Width özellik değerlerini ayarlayarak, form'un boyut ve ekran üzerindeki yerleşimindeki değişikliklerini gözlemleyin. Form'u yeniden boyutlandırın ve taşıyın. Özellikler penceresinde bu değerlerin nasıl değiştiğini gözlemleyin. Caption(Başlık) özelliğini ayarlayın. Yeni bir background rengini, 3. derste anlatılan teknikleri kullanarak değiştirin. BorderStyle(Sınır Stili) özelliğinin etkilerini görmek için, bir değer verin (ya 1-Fixed Single (sabit Tek) veya 2-Sizable (Boyutu Değiştirilebilir) kullanın). Bu derslerde, yalnızca bunları kullanacağız. Projeyi çalıştırın. Evet, bir projeyi üzerinde sadece bir formla, formu kontrol olarak çalıştırabilirsiniz ! Her durumda form'u yeniden boyutlandırmayı deneyin. Fark'a dikkat edin. Bu örnek projeyi durdurun.

Olaylar

Form; temelde, diğer kontroller için bir konteyner(kap) gibi davranır, fakat olayları destekler, yani bazı kullanıcı etkileşimlerine cevap verebilir. Bu derslerde yalnızca iki form olayı ile ilgileneceğiz:

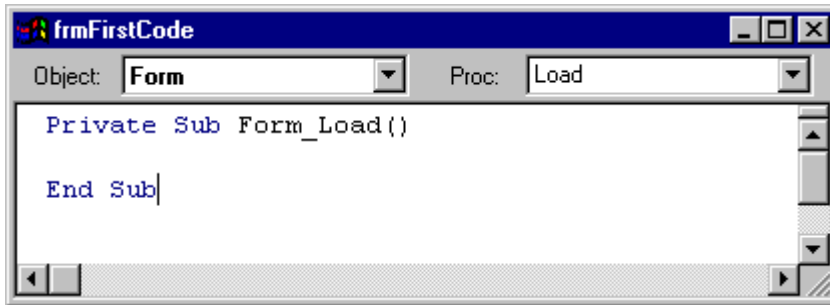
<u>Olay</u>	<u>Açıklama</u>
Click	Click(tıklama) olayı, sadece kullanıcı form'ü fare ile tıklarsa, yerine getirilir.
Load	Olay, form ilk olarak bilgisayarın hafızasına yüklenerek yerine getirilir. Değişik ilk değerler ve diğer proje değerlerini vermek için, iyi bir olaydır.

Eski derslerden hatırlayın, kontrol isimlerinin, olay prosedürlerinde kullanıldığını öğrendik. Bunlar, form'lar için doğru değildir. Bütün form olay prosedürleri aşağıdaki format'a sahiptir:

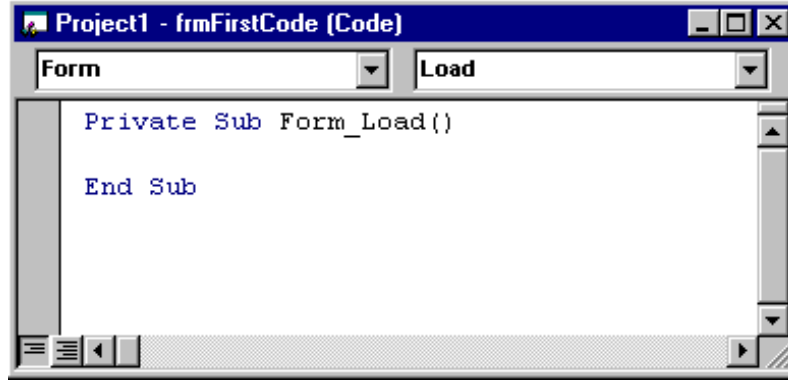
Form_EventName

Anlamı ise, form'a hangi Name özelliği verirsiniz verin, olay prosedürleri Form kelimesi altında listelenirler. Böylelikle, kod penceresinde form olay prosedürlerine baktığınızda, Object List'esini Form'u bulana kadar, kaydırma çubuğu ile kaydırarak arayın. Şunu da belirtmek gerekir, eğer biz form'a frmFirstCode ismi verirsek, kod penceresi aşağıdaki şekilde görünecektir :

VB4:



VB5, VB6:

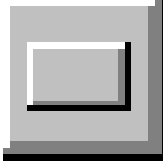


*Dikkat edin, nesne listesinde Form kelimesi görünmektedir(frmFirstCode değil).
Form olay prosedürleri ile çalışırken, daima bu özelliğin farkında olmalıyız. Bütün diğer kontroller,
kendilerine verilmiş isim özellikleri ile birlikte, nesne listesinde görüneceklerdir.*

Komut Butonu Kontrol'ü

Command button(komut butonu) en yaygın olarak kullanılan Visual Basic kontrollerinden birisidir. Komut butonları; başlatmak, durdurmak veya belirli yöntemleri bitirmek için kullanılır. Command button(komut butonu) araç kutusundan seçilir. Aşağıdaki şekilde görünür :

Araç Kutusu İçinde
özellikler:



Form Üzerinde default(varsayılan)



Özellikler

Komut butonu için birkaç faydalı özellik:

<u>Özellik</u>	<u>Açıklama</u>
Name	Komut butonunu tanımlamak için kullanılan isim. Komut butonları isimleri için kullanılan üç harflik örnek cmd 'dir.
Caption	Komut butonu üzerinde görünen yazı
Font	Caption yazısı'nın style(stil), size(boyut), ve (type)tip'ini belirler.
Left	Form'un sol kenarından komut butonunun sol kenarına olan uzaklıktır.
Top	Form'un üst kenarından, komut buton'unun üst kenarına olan uzaklıktır.
Width	Komut butonu'nun twip birimi ile genişliği.
Height	Komut butonu'nun twip birimi ile yüksekliği
Enabled	Komut Buton'unun kullanıcı olaylarına (run mod'unda) karşılık (cevap) verip, vermeyeceğini belirler.

Visible

Komut Buton'unun form üzerinde (run mod'unda) görünür olup, olmayacağını belirler.

Örnek

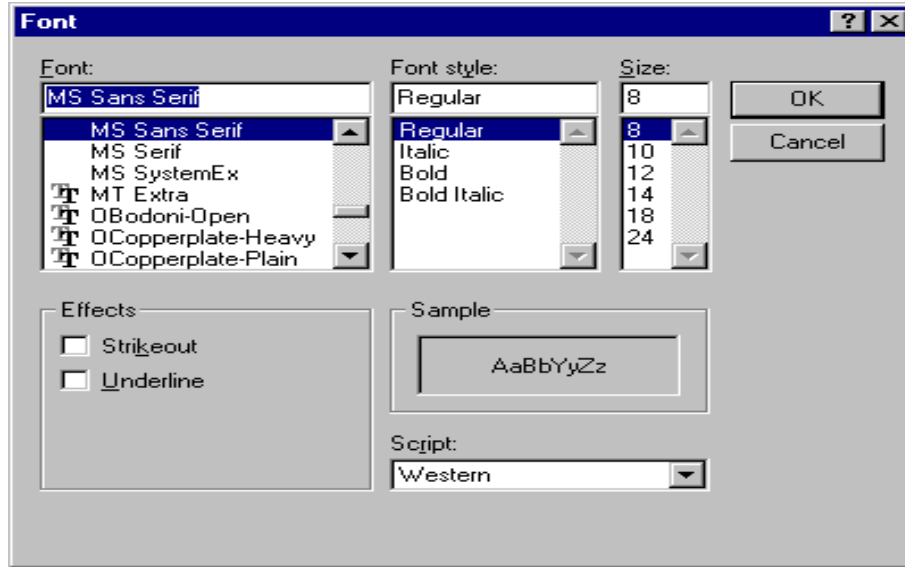
Visual Basic'e ve yeni bir projeye başlayın. Form üzerine bir komut butonu koyun. Butonu gezindirin, Top ve Left özelliklerindeki değişimleri izleyin. Buton'u tekrar boyutlandırın ve Width ve Height değerlerinin nasıl değiştiğine dikkat edin. Caption özelliğini verin.

Komut butonuna ilaveten birçok kontrol bir Font(yazı) özelliğine sahiptir.

Şimdi bunun nasıl değiştirileceğini anlamak için bir bakalım. Font, Caption'ın nasıl görüneceğini belirler. Özellikler penceresinde Font'u tıklarsanız, ellipsis adı verilen ve aşağıda görünen bir buton, pencerenin sağ tarafında ortaya çıkar:



Bu buton'u tıklayın, Font Window(Font Penceresi) görünecektir.



Bu pencereden, bilginin üç temel parçasını seçebilirsiniz: Font(Yazı), Font Style(Font Stili), ve Size(Boyut). Aynı zamanda altı çizili bir font'da seçebilirsiniz.

Bu pencere, bilgisayarınızda depolanmış bütün fontları listeler. Font özelliğini ayarlamak için, bu pencere içinde seçimlerinizi yaparak OK'ı tıklayın. Farklı fontları, font stillerini ve font boyutlarını, komut butonunun Caption özelliğini değiştirmek üzere, deneyin.

Komut butonunda listelenen diğer iki özellik; Enabled ve Visible'dır. Bu özelliklerin her birisi, ya True-Doğru (On) ya da False-Yanlış (Off)'tur. Bir çok diğer kontrol de aynı zamanda bu özelliklere sahiptir.

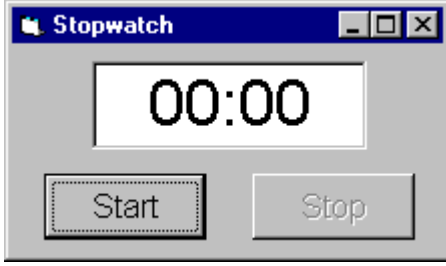
Peki, neden bunlara ihtiyacınız var ?

Eğer bir kontrol'ün Enabled özelliği False ise, kullanıcı bu kontrole ulaşamayacaktır. Üzerinde Start(Başlat) ve Stop(durdur) butonları bulunan Stopwatch(Kronometre) projemize bakalım:



Kullanıcının bu proje ile Start ve daha sonra Stop tuşlarını tıklamasını ve geçen sürenin bulunmasını istiyorsunuz. Kullanıcının Start buton'una tıklamadan, Stop buton'una basamamasını istiyorsunuz. Böylece başlangıç olarak Start buton'unun Enabled özelliğini True'ya ve Stop buton'unun Enabled özelliğini False'a ayarlamanız gerekmektedir. Bu yolla kullanıcı sadece Start'ı tıklayabilir. Bir kere kullanıcı Start'ı tıkladığında, özellik değerlerini değiş tokuş etmeniz gerekmektedir. Bu, Start buton'unun Enabled özelliğini False ve Stop button'unun Enabled özelliğini True yapmaktır. Kullanıcı ancak bu yolla Stop'u tıklayabilir.

Enabled özelliğinin False yapılmasının etkisi, sadece, Visual Basic run mod'unda iken kendisini gösterir. Bir komut butonu Enabled yapılmadığında (Enabled, False olarak verirse), 'silik' olarak görünecektir ve kullanıcı bunu tıklayamayacaktır. Stop, Enabled yapılmazsa, kronometre aşağıdaki gibi görünecektir:



Böylelikle *Enabled* özelliğini, kontrol'ün, form üzerinde geçici olarak *disabled* (Etkinliğini Kaldırma) yapmak için kullanın. Daha evvel üzerinde konuştuğumuz gibi, bu, proje dizayn yönteminde bir karar verme durumudur.

Visible(görünürlük) özelliği, biraz daha zorlayıcıdır. Bir kontrol'ün *Visible* özelliği *False* olarak ayarlandı ise (varsayılan değer *True*'dür), kontrol form üzerinde bulunsa bile, proje çalıştırıldığında kontrol form üzerinde olmayacaktır!

Şimdi form üzerine yerleştirdiğimiz, özelliklerini ayarladığımız ve onun için olay prosedürleri yazdığımız bir kontrol'ün neden *invisible*(görünmez) olmasını isteyebiliriz? Cevap; *Enabled* özelliği ile benzerdir. Birçok kereler, bir proje içinde bir kontrol'ün geçici olarak ortadan kaybolmasını isteyebileceksiniz.

1. derste, *Sample* projesinde, oyuncakların görünüp görünmemesini sağlayan *check box*(onay kutu)'ları bulunmaktaydı. Oyuncakların görünmesi, *image control*'ünün *Visible* özelliğinin kontrolü ile gerçekleştirilmiştir. Veya, küçük *kronometre* örneğinde olduğu gibi, bir *buton*'un *Enabled* özelliğini *False*'a ayarlamak yerine, onu *unclickable* (tıklatılmaz) yaparak, *Visible* özelliğini de *False*'a ayarlayarak, onun daima (sürekli) olarak görünmesini önledik. Her iki yolla da, aynı istenen sonucu elde edersiniz. Bu başka bir proje dizayn kararıdır. *Enabled* özelliği gibi, *Visible*'ın *False* yapılmasının etkileri yalnızca *run mod*'unda görülür. Bu anlamlıdır. Bir projeyi *invisible control*'lerle dizayn etmek zordur!

Şimdi, üzerinde çalıştığınız örnekte bulunan *komut buton*'unun *Enabled* ve *Visible* özellikleri ile oynayın. Bu özelliklerden birisine değer girdiğinizde, projeyi çalıştırın ve sonuçlarını görün. Dikkat edin, *Enabled* özelliği *False* olarak verilmiş *butonu* tıklayamazsınız. Yine dikkat edin, *Visible* özelliği *False* olarak verildiğinde, *buton* orada yerinde olmayacaktır. Daha sonra örnek projeyi durdurun.

Olaylar

İlgilendiğimiz yalnızca bir komut butonu olayı vardır, fakat bu çok önemli bir olaydır:

Olay Açıklama

Click Kullanıcı fare ile komut butonunu tıkladığında, olay yerine getirilir.

Her komut butonu, Click Event(Tıklama Olay)'ına karşılık gelen bir olay prosedürüne sahiptir.

Uzun bir süre sonunda bir Visual Basic projesine giriş yapabilmeye artık hazırız - BASIC dili ! Bir Visual Basic Projesinde, olay prosedürlerin, kontrol olaylarını bilgisayar tarafından yerine getirilenlere bağlantılamak için kullanıldığını öğrendiniz.

Bu olay prosedürleri BASIC kullanılarak yazılır. Yani Visual Basic' i bilmek için, BASIC bilmeniz gerekmektedir. Bu kitapta takip eden her derste, BASIC dili hakkında yeni yeni şeyler öğreneceksiniz.

Event Procedure(Olay Prosedür) Yapısı

Biliyorsunuz, bu olay prosedürleri, şimdiye kadar Visual Basic Kod Penceresi'nde görünebiliyordu. Her olay prosedürü aynı genel yapıya sahiptir. İlk, form'un bir header(Başlık) satırı vardır:

Private Sub ControlName_EventName()

Bu, Private(Özel) (yalnızca form'umuzdan ulaşılabilir), Subroutine'nin (bir olay prosedürü için başka bir isim) , ControlName kontrolü için, EventName olayı olduğunda yerine getirildiğini gösterir. Anlamlı, değil mi ?

Event procedure(olay prosedür)'ü, başlık satırından sonra başlar.

Olay prosedürü kodu, basit bir anlatımla bilgisayara ona ne yapması gerektiğini söyleyen, satır-satır verilen talimatlar serisidir. Bilgisayar, ilk satırı işler, daha sonra ikincisini ve daha sonra bunları takip eden diğer satırları... Bilgisayar, olay prosedürünün footer(taban) satırına ulaşana kadar bu işlemleri yapar. Footer(taban) satırı:

End Sub

Olay prosedürü kod'u, BASIC dili ile yazılır. BASIC, bilgisayara belirli birtakım şeyleri yapmasını sağlayan anahtar kelime ve semboller set'inden oluşur. BASIC'in içeriği çok geniş ve zengindir ve bu kurs'ta bunlardan yapabildiğimiz kadarını inceleyeceğiz. Hemen bu noktada bir uyarı ! Daha önce söyledik, ama bir kere daha söyleyelim.

Bilgisayar Programlaması kesinlik gerektirir - hatalara izin vermez ! Özellikle olay prosedürleri yazarken, kesin olmalısınız ! Bilgisayar çağında iyi klavye kullanımı (yazım) tecrübesi bir

gerekliliktir. Visual Basic Programlaması öğrendikçe, klavyeden yazım becerinizi geliştirebileceğiniz bu amaçla piyasada bulunan yazılımları, faydalı bulabilirsiniz.

Klavyeden daha iyi yazım tecrübesi, Visual Basic uygulamalarınızı yapılandırırken daha az hata yapacağınız anlamına gelir.

Atama (Assignment) Bildirileri

BASIC'te en basit ve en fazla kullanılan assignment statement (atama bildirisi)' leridir. Aşağıdaki şekle sahiptir:

LeftSide = RightSide

' = ' sembolü assignment operator(atama operatör)'ü olarak adlandırılır.

Bu sembol'ün aritmetik'te eşittir anlamına gelen sembol olduğunu biliyorsunuz. Fakat Bilgisayar Programlanmasında eşittir olarak adlandırılmaz.

Neden böyle ?

Bir atama bildirisini, solda ne bulunmaktaysa, sağdakilerle değiştirilecektir şeklinde ifade ederiz.

Atama bildirisinin sol tarafı bir kontrol özelliği gibi, yalnızca tek bir terim olabilir. Sağ taraf ise herhangi bir kabul edilebilir BASIC ifadesi olabilir. Bazı matematiksel hesaplamalar veya hesaplanması gereken bazı şeyler gerekebilir. Eğer böylesi hesaplamalar varsa, atama yapılmadan evvel gerçekleştirilmelidir.

Şimdi çok genel ifadelerle konuşuyoruz ve konuşmak zorundayız da ! Atama bildirileri fikri, daha fazla BASIC öğrendikçe, çok daha açık ve anlamlı olacaktır.

Property(Özellik) Tipleri

Hatırlayın, bir özellik, bir kontrol hakkında birşeyi açıklar: boyutlar(size), renk(color), appearance(görünüm), Her property(özellik), belirli bir tip(type)'te, temsil ettiği bilginin cinsine bağlı olarak, bir özelliği temsil eder.

Design(dizayn) mod'unda bir değer vermek için özellikler penceresini kullanırsak, Visual Basic otomatik olarak uygun olan tipi sağlayacaktır. BASIC atama bildirisi kullanarak, bir olay prosedüründe, eğer bir özelliği değiştirmek istiyorsak, özellik tipini bilmek zorundayız, zira ona uygun ve hatasız tipte bir değer girebilelim. Hatırlayın; 'dot notation' (nokta notasyon) adı verdiğimiz bir atama tipini, run mod'unda değiştirebilmemiz için kullanıyorduk :

ControlName.PropertyName = PropertyValue

ControlName , kontrole atanmış *Name*(İsim) özelliğidir. *PropertyName* özelliğın ismidir ve *PropertyValue*, *PropertyName*’ e atadığımız yeni bir değerdir.

Burada dört tipte özellik ile ilgileneceğiz:

Birinci özellik tipi, integer(tamsayı) tipidir. Bunlar tam, k s ratlı bi imde ifade edilmeyen, sayılardır. Top, Left, Height, ve Width gibi  zellikler integer tipindedir. Yani, e er integer tipte bir  zelli e bir de er atayacaksak, tamsayı kullanaca ız. Bir  rnek olarak, frmExample adını verdi imiz bir form’un width  zelli ini 4,000 twip’e ayarlayacaksak, BASIC’te ş nları yazmalıyız:

frmExample.Width = 4000

Bu, ge erli form’un Width(genişli ini) yeni bir de er olan 4000 olarak atayaca ımızı belirtmektedir. Dikkat edin, BASIC’te 4000 yerine, i inde virg l bulunan 4,000 şeklinde yazamayız.

 kinci bir  zellik tipi long integer(uzun tamsayı) tipidir ve uzun tamsayı tipi, aynen isminin ifade etti i gibidir. Integer tipte bir  zellik en fazla 32,767 de erini alabilir. Bazen bu sayılardan daha b y k sayılara ihtiya  duyabiliriz, bundan dolayı long integer tipi vardır. Bir long integer 2,147,483,647 sayısına kadar de er alabilir.

Yeteri kadar b y k de il mi?

Fakat Bill Gates, sahip oldu u serveti bu tipte bir  zellikle yazamaz,   nk  yetersiz kalır. Belkide, Microsoft ’dan birisinin kendisi i in, very long integer( ok uzun tamsayı)’yı icat etmesi gerekebilir. Uzun tamsayı’ları kullanan  ok genel  zellikler, BackColor(arka-artalan rengi) ve ForeColor( n kısım rengi)’i ve bazı di er kontroller i in olanları g receksiniz. Hatırlayın,  nceki derslerimizde, gri renk i in atanan  zellik de eri bir uzun tamsayı i in &H8000000F& şeklinde yazılmıştı-bu bir kısa g sterim(notasyon)’dur (bir heksadesimal sayı olarak adlandırılır). Renk  zelliklerine de erler atarken, uzun tamsayı’ları kullanmalıyız.

Şanslıyız, Visual Basic renklere long integer sayılar vermek i in, bir  ok kolay yol sa lamaktadır. B ylelikle, long integer’lar ile  alıřmayı kolay hale getirmiştir. Renkleri kullanmanın bir yolu, symbolic constants(sembolik sabitler)’dir. Sembolik sabitler, Visual Basic’de pek  ok yerde kullanılmaktadır – bunlardan pek  o unu bu dersler ilerledik e g rece iz. B t n sembolik sabitler iki harf ile bařlar; vb (Visual Basic).

Renkler için bazı sembolik sabitler aşağıdaki gibidir:

vbBlack - Siyah

vbRed - Kırmızı

vbGreen - Yeşil

vbYellow - Sarı

vbBlue - Mavi

vbMagenta - Mor

vbCyan - Cyan (gök mavisi)

vbWhite - Beyaz

Bu sabitlerin herbirisi, kendisine karşılık gelen, temsil ettiği renge ait long integer değerini depolar(store). Bizim örneğimizdeki formun BackColor özelliğini maviye çevirmek için, şu atama bildirisini kullanmalısınız:

frmExample.BackColor = vbBlue

İfade şu anlama gelir: Form'un BackColor(artalan renk)'i, vbBlue olarak adlandırılan sembolik sabit'in temsil ettiği long integer değeri ile değiştirilir.

Bir diğer özellik(property) tipi ise Boolean tipidir. İsmi meşhur matematikçi Boole'dan almıştır. Yalnızca iki değere sahip olabilir: True(Doğru) veya False(Yanlış).

Daha evvel gördüğümüz gibi, komut butonu'nun Enabled ve Visible özellikleri Boolean değerlerine sahip olabilirler. Böylece, Boolean tipinde özelliklerle çalışırken, yalnızca True veya False değeri verdiğimizden emin olmalıyız. Örneğimizin formunu gözden kaybettirmek için (bunu yapmak iyi bir şey değil !), aşağıdaki atama bildirisini kullanmalıyız:

frmExample.Visible = False

Bu ifade, form'un geçerli Visible özelliğinin, Boolean değeri olan False ile değiştirileceğini söylemektedir. Tekrar eski haline döndürmek için:

frmExample.Visible = True

Belirtmemiz gereken son özellik tipi string(dizilim) tipidir. Bu tipte özellikler, basitçe, adının ifade ettiği gibidir - karakterlerin dizilimi. Bir string; bir isim, bir sayı dizisi, bir cümle, bir paragraf veya herhangi bir tipte karakterler dizisi olabilir.

Birçok kereler, bir string hiçbir karakter içermez(bir boş string). Caption özelliği, string tipte bir özelliktir. Visual Basic’de, string’ler ile birçok çalışmalar yapacağız ve bu yüzden bununla aşına olmamız gerekmektedir.

String tipte özellikler atarken, tek bir sırrı(inceliği) vardır; bu da string değerini tırnak içine almaktır ("). Özellikler penceresi içinde string karakter tırnak içine alınmadığından, sizin tırnak içine alma zorunluluğunu, unutma eğiliminiz bulunmaktadır.

Örneğimizde form caption(başlık)’ı için şunu kullanabiliriz:

```
frmExample.Caption = "Bu tırnak içinde bir baslıktır"
```

Bu atama bildirisi, form’un Caption özelliğinin, ifadenin sağ tarafındaki yazı ile değiştirilmesi bildirisini vermektedir. Şimdi, atama bildirilerinin nasıl olduğu hakkında fikir sahibi olmalısınız.

Yorumlamalar (Kodlama içine notlar düşme)

Proje dizayn’ı konusunda bahsettiğimiz gibi, BASIC kod’u yazarken, uygun programlama kurallarını takip etmek zorundayız. Böyle bir kural, kod’unuzu uygun(düzgün) biçimde yorumlamanız !

Kodlarınızda, bilgisayar tarafından göz önüne alınmayan, çalıştırılmayan ve ne yaptığınızı açıklayan tanımlama yazıları koyabilirsiniz.

Bu comments(yorumlamalar), kod’unuzu anlamak için yardımcı olacaktır. Bunlar aynı zamanda ileride yapacağınız değişiklikleri çok daha kolay hale getirecektir.

Kod’unuzun içerisine bir yorumlama koymak için, bir kesme işareti (') koymanız gerekmektedir. Bu işaret 1 tuşunun sağ tarafında bulunan tuş değil, Ş tuşunun altında bulunup (AltGr) tuşu ile yazılır. Bu işareten sonra yazılan her şey bilgisayar tarafından ihmal edilir. Komple bir BASIC kod satırını açıklayan, böyle bir yorum satırı, aşağıdaki gibidir:

```
'Form'u maviye çevir
```

```
frmExample.BackColor = vbBlue
```

Veya, yorumunuzu aynı satırda atama bildirisinden sonra aşağıdaki gibi yazabilirsiniz:

```
frmExample.BackColor = vbBlue 'Formu mavi yapar
```

Programcı olarak siz, kod’unuzu ne kadar yorumlayacağınıza karar vermelisiniz. Bu kursta size sunulan projelerde yeteri kadar yorumlama sağlayacağız. Şimdi ilk önce böyle bir projeye bakalım:

Proje - Form Fun (Eğlenceli Form)

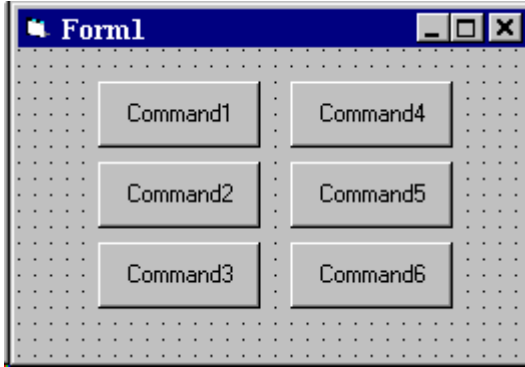
Proje Tasarımı

Bu projede, komut butonlarını kullanarak form özellikleri ile biraz eğlenelim.

Bir butonumuz form'u büyütecek, diğeri küçültecek ve diğeri iki buton ise form'un rengini değiştirecektir. Bunlara ilaveten yine form üzerinde bulunan iki butondan birisi form'u gözden kayedecek, diğeri ise tekrar görüntüye getirecektir.

Form üzerine kontrolleri yerleştirin:

Visual Basic'te yeni bir proje başlatın. Form'un büyüklüğünü, üzerine 6 tane buton sığacak şekilde ayarlayın. 6 tane komut butonunu form üzerine yerleştirin. Butonları aşağıda görüldüğü şekilde yerleştirin :



Bir uyarı ! Eğer Windows uygulamalarını biraz kullandıysanız, Copy(Kopyala) ve Paste(Yapıştır) adını verdiğimiz edit özelliklerini de kullanmışsınızdır.

Bu, ikincisini çıkartacağınız(bir tane daha yapacağınız) bir şeyi kopyalamak, nereye kopyası çıkarılacaksa oraya gitmek ve buraya yapıştırmaktır. Bu kelime işlemcilerde daima kullanılır.

Keşfetmiş olabilirsiniz, Visual Basic ile çalışırken, kontrolleri kopyalayıp yapıştırabilirsiniz. Bunu, burada yapmayı da çok cazip bulabilirsiniz - yalnızca bir komut butonu oluşturup bunu kopyaladıktan sonra 5 kere daha yapıştırmak gibi kısa ve kolay bir yol varken, niçin 6 kere yeni komut butonu oluşturasınız? Evet, bunu yapabilirsiniz, fakat burada yapmayın !

Kontrolleri kopyalamak size farklı tipte bir kontrol verir – bunlardan bir tanesini daha ileri seviye Visual Basic derslerinde göreceksiniz. Bu derslerde ihtiyaç duyduğumuz her kontrolden sadece bir tane oluşturacağız (yani 6 ayrı kontrolü teker teker ve ayrı ayrı oluşturacağız). Daha sonra, daha iyi bir programcı haline geldikten sonra, kontrolleri kopyalayıp sonra yapıştırıldığında ne olduğuna bakacaksınız.

Kontrol Özelliklerini Düzenleme

Kontrol özelliklerini, özellikler penceresini kullanarak düzenleyin. Hatırlayın, özellikler penceresinde seçilen bir kontrolü değiştirmek için, ya pencerenin en üstünde bulunan kontrol listesini kullanın veya sadece tercih ettiğiniz kontrolü tıklayın.

Proje kontrol özellikleri için, kontrolleri her zaman varsayılan değerleri ile listeleyeceğiz (kontrol, form üzerine yerleştirildiğinde Visual Basic tarafından atanan varsayılan(default) değerler).

Form1 Form'u:

<i>Property Name(Özellik İsmi)</i>	<i>Property Value(Özellik Değeri)</i>
<i>Name</i>	<i>frmFormFun</i>
<i>Caption</i>	<i>Form Fun (Form Eğlencesi)</i>

Command1 Command Button(Komut Butonu):

<i>Property Name(Özellik İsmi)</i>	<i>Property Value(Özellik Değeri)</i>
<i>Name</i>	<i>cmdShrink</i>
<i>Caption</i>	<i>Shrink Form (Formu Küçült)</i>

Command2 Command Button(Komut Butonu):

<i>Property Name(Özellik İsmi)</i>	<i>Property Value(Özellik Değeri)</i>
<i>Name</i>	<i>cmdGrow</i>
<i>Caption</i>	<i>Grow Form (Formu Büyüt)</i>

Command3 Command Button:

<i>Property Name</i>	<i>Property Value</i>
<i>Name</i>	<i>cmdHide</i>
<i>Caption</i>	<i>Hide Buttons (Butonları Gizle)</i>

Command4 Command Button:

<i>Property Name</i>	<i>Property Value</i>
<i>Name</i>	<i>cmdRed</i>
<i>Caption</i>	<i>Red Form(Kırmızı Form)</i>

Command5 Command Button:

<i>Property Name</i>	<i>Property Value</i>
<i>Name</i>	<i>cmdBlue</i>
<i>Caption</i>	<i>Blue Form(Mavi Form)</i>

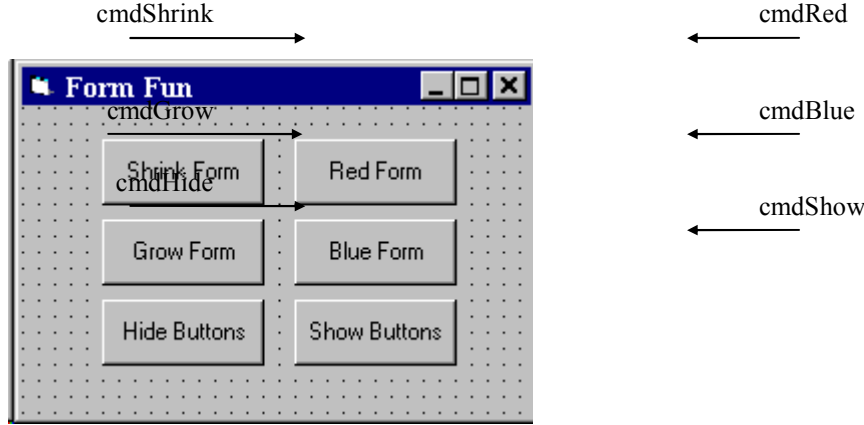
Command6 Command Button:

<i>Property Name</i>	<i>Property Value</i>
<i>Name</i>	<i>cmdShow</i>
<i>Caption</i>	<i>Show Buttons (Butonları Göster)</i>
<i>Visible</i>	<i>False</i>

Eğer isterseniz diğer özelliklerini de değiştirebilirsiniz – belki komut butonlarının Font özelliğini değiştirmek isteyebilirsiniz.

frmFormFun

Özellikleri düzenledikten sonra, form'unuz aşağıdaki gibi görünmelidir:



Form üzerinde 6 buton bulunmaktadır; ikisi form'un boyutlarını ve diğer iki tanesi ise form'un rengini değiştirmek içindir. Bir tanesi form'u gözden kaybetmek, bir tanesi ise tekrar görünür hale getirmek içindir.

Dikkat edin Show Buttons(Butonları Göster) komut butonu False değerine sahip bir Visible özelliğine sahiptir. Onun orada form üzerinde ilk olmasını istemiyoruz. Çünkü butonlar halihazırda orada bulunacaklardır. Hide Buttons(Butonları Gizle) kontrol'ünü tıkladığımızda (onların Visible özelliğini değiştirerek) butonların kaybolmasını ve Show Buttons(Butonları Göster) butonunu görünür hale getiririz.

Anlamlı, değil mi ?

Fakat , Show Buttons buton'unun Visible özelliğini False yapmamıza rağmen hala neden orada ? Hatırlayın, False olan Visible özelliği sadece run modunda ortaya çıkar.

Olay Prosedürleri Yazmak

Form'umuzun üzerinde 6 komut butonu bulunmaktadır. Bu butonlardan herbirisine ait Click olay prosedürü için kod yazmamız gerekmektedir. Aynı zamanda form için de bir Click olay prosedürü

yazmak istiyoruz - neden olduğunu açıklayacağız. Form üzerinde bulunan ve form'u ufaltan bir butonumuz var. Formu o kadar çok ufaltırsak sonuçta yine büyültmek için gerekli butonu bulup tıklayamayız. Bunu önlemek için formu tıklayarak büyütme imkanı veririz. Bu 'ileride ne olacağını düşünme' örneği, daha önce hakkında konuştuğumuz proje dizayn kavramlarından birisidir.

Her olay prosedürü için kod penceresini kullanacaksınız. Nesne listesinden kontrolü ve prosedür listesinden olayı seçin. Başlık satırı ve taban satırı arasındaki alanı tıklayın ve kodu yazmaya başlayın. Bu kadar kolay ! Fakat tekrar ediyoruz, her şeyi bu kitapta tarif edildiği gibi yazdığınızdan emin olun, kesin olmak zorundasınız !

İlk olarak, cmdShrink_Click olay prosedürünü yazalım. Bu prosedürde , form yüksekliğini 100 ve form genişliğini de 100 twip azaltacağız :

```
Private Sub cmdShrink_Click()  
'Formu ufalt.  
'Formun yüksekliğini 100 twip azalt.  
frmFormFun.Height = frmFormFun.Height - 100  
'Formun genişliğini 100 twip azalt.  
frmFormFun.Width = frmFormFun.Width - 100  
End Sub
```

Diğer olay prosedürlerine geçmeden, bu noktaya biraz daha yakından bakalım. Çünkü üzerinde iyice durmadığımız birkaç fikir kullanılmaktadır. Üzerinde Shrink Form yazılı olan butonu tıkladığımızda, yerine getirilen olay prosedürü budur. Bu yorum bildirilerinin kolaylıkla farkına varacaksınız. Yorum olmayan bildiriler form yüksekliğini ve genişliğini değiştirir. Height(yükseklik) değerini değiştiren aşağıdaki bildiriye bakın:

```
frmFormFun.Height = frmFormFun.Height - 100
```

Hatırlayın, atama operatörü nasıl çalışıyor (=) . Sağ taraf önce değerlendirilir. Daha sonra 100 (- işareti kullanılarak) halihazır form yüksekliğinden çıkarılır. Bu değer ifadenin sol tarafına atanır (frmFormFun.Height). Sonuç, form Height özelliği, Height değeri eksi 100 twip'dir. Bu kod satırından sonra, Height özelliği 100 azaltılır ve form ekranda daha ufak görünür.

Bu ifade aynı zamanda buna niçin atama operatörü (=) deyip, eşittir işareti demediğimizi de anlatmaktadır. Herhangi bir kimse ifadenin sol tarafını, ifadenin sağ tarafına eşit olamayacağını görebilir. FormFun.Height değeri ne olursa olsun, sağ taraf daima sol taraftan 100 birim daha ufaktır.

Fakat, bu bir eşitlik olmamasına rağmen, doğru olmadığını bile bile, programcıların çoğunlukla bu bildiriye, "frmFormFun.Height eşittir frmFormFun.Height eksi 100," şeklinde okuduğunu işitirsiniz ! Kendi programlarınızı yazmaya başlarken, atama bildirilerinin nasıl çalıştığını hatırlayın. Şimdi, diğer olay prosedürlerine bakalım. cmdGrow_Click prosedürü, form yüksekliğini 100 ve form genişliğini 100 twip arttırır:

```
Private Sub cmdGrow_Click()  
'Formu büyüt  
'Form yüksekliğini100 twip arttır  
frmFormFun.Height = frmFormFun.Height + 100  
'Form genişliğini 100 twip arttır  
frmFormFun.Width = frmFormFun.Width + 100  
End Sub
```

cmdRed_Click olay prosedürü form background rengini kırmızıya çevirir:

```
Private Sub cmdRed_Click()  
'Formu kırmızı yap  
frmFormFun.BackColor = vbRed  
End Sub
```

cmdBlue_Click olay prosedürü form background rengini maviye çevirir:

```
Private Sub cmdBlue_Click()  
'Formu mavi yap  
frmFormFun.BackColor = vbBlue  
End Sub
```


cmdHide_Click olay prosedürü, bütün komut butonlarını gizlemek için kullanılır (Visible özelliğini False olarak verir) – cmdShow istisnadır ve Visible olarak ayarlanır:

```
Private Sub cmdHide_Click()  
'cmdShow haricinde butonlari gizle  
cmdGrow.Visible = False  
cmdShrink.Visible = False  
cmdHide.Visible = False  
cmdRed.Visible = False  
cmdBlue.Visible = False  
'cmdShow butonunu göster  
cmdShow.Visible = True  
End Sub
```

ve cmdShow_Click olayı, bu etkileri tersine çevirmek için kullanılır:

```
Private Sub cmdShow_Click()  
'cmdShow butonu hariç butonlari göster  
cmdGrow.Visible = True  
cmdShrink.Visible = True  
cmdHide.Visible = True  
cmdRed.Visible = True  
cmdBlue.Visible = True  
'cmdShow butonunu gizle  
cmdShow.Visible = False  
End Sub
```

Son olarak, Form_Click olay prosedürü de aynı zamanda form'u 'grow' (büyütmek) için kullanılır ve bu yüzden cmdGrow_Click ile aynı koda sahiptir:

```
Private Sub Form_Click()  
'formu büyüt  
'form yüksekligini 100 twip arttir  
frmFormFun.Height = frmFormFun.Height + 100
```

```
'Form genişliğini 100 twip arttır  
frmFormFun.Width = frmFormFun.Width + 100  
End Sub
```

Buradaki prosedürün doğru bir prosedür olduğundan emin olun. Form kontrolünü seçtiğinizde, görünecek prosedür Load olacaktır. Prosedür listesini kullanarak, Click olay prosedürüne bakınız. Yeni bir projeyi saklamak(kaydetmek) için daha önce anlatılan teknikleri gözden geçirin. Projenizi kaydedin.

Bu prosedürlerin her birisinde neler olduğunu kolaylıkla görebilmelisiniz.

cmdHide ve cmdShow butonlarının click olaylarında, Visible özelliğinin nasıl kullanıldığına özellikle dikkat edin. Dikkat etmeniz gereken bir başka nokta ise, pek çok olay prosedürünün, bunların kodlamalarına olan sıkı benzerliğidir.

Örnek, Form_Click olayı, cmdGrow_Click olayı ile birebir eşdeğerdir. Bu Visual Basic projelerinde çok rastlanan bir durumdur. Kontrolün yerleştirilmesinden farklı olarak, kod yazarken, Copy ve Paste gibi editör özelliklerinin kullanılmasını özellikle tavsiye ediyoruz ! Bir şeyi kopyalamak için; fareyi kullanarak, istenilen yazıyı işaretleyin - aynı şeyi bir kelime işlemcide yapıyorsunuz. Daha sonra, Visual Basic ana menüsünden, Edit'i daha sonra da Copy 'yi seçin. Nereye yapıştırmak istiyorsanız, imleç'i (cursor)'u oraya götürün.

Diğer olay prosedürlerini de taşıyabilirsiniz. Edit'i daha sonra da Paste'i seçin.

O da ne ? Kopya görüldü. Yapıştırılan yazının bir miktar düzeltilmeye ihtiyacı var. Fakat güzel tarafı, copy ve paste fonksiyonu kod yazarken çok zaman kazandıracaktır. Bu ise yapmak istediğiniz şeydir. Çünkü, belki farketmişsinizdir, programlama yaparken, böylesi basit bir projede bile, belirli bir miktarda yazıyı klavyeden girme gereği vardır.

Bir diğer faydalı özellik ise Find(Bul) ve Replace(Değiştir) özelliğidir. Gerektiği zaman bu özelliği de kullanın.

VB5 ve VB6 yazı yazma yükünü hafifletecek ve yapabileceğiniz hataları azaltmanız için başka bir yol daha sunar. Eğer Visual Basic ana menüsünde Tools, ve daha sonra da

Options'ı seçer ve daha sonra burada bulunan Editor'ü tıklarsanız burada bulunan Auto List Members adı verilen bir seçeneği görürsünüz. Bu seçenek, eğer kod penceresinde BASIC yazarken

seçili ise, belirli bazı noktalarda ufak kutucuklar aniden ortaya çıkacaktır(pop-up). Bunlar, üzerinde çalıştığınız bildiriği mantıksal olarak tamamlayan (bitiren) bilgileri gösterir.

Böylelikle, bundan sonrası için, eksik olan tamamlama bilgisini yazmak yerine kalan kısmı buradan seçebilirsiniz. VB5 veya VB6 kullanıyorsanız, Auto List Members Option'ları denemek isteyebilirsiniz. Kullanımı ile ilgili daha fazla bilgiye erişmek için, on-line help kullanın.

Projeyi Çalıştırın

Devam edin ! Projenizi çalıştırın(Run) - Visual Basic toolbar üzerinde bulunan Start(Başlat) butonunu tıklayın. Eğer doğru biçimde çalışmıyorsa, bu aşamada tavsiye edilebilecek tek şey, projeyi durdurmak, yazılı olanı tekrar kontrol etmek ve yeniden denemektir. 'debugging'(hata ayıklama) tekniklerini diğer derste öğreneceğiz.

Bütün komut butonlarını deneyin. Form'u büyütün, küçültün, form rengini değiştirin, butonları gizleyin, butonları tekrar görünür hale getirin. Bütün butonları denediğinizden ve istediğiniz her türlü çalışmayı yaptığınızdan emin olun. Gerekli yerleri tıklayarak, istediğiniz sonuçları aldığınızdan emin olun. Bu yapılması çok açık ve görünür bir şey gibi gelebilir fakat büyük projelerde, yazdığınız bazı kodlar olacaktır ki bu kodlar asla yerine getirilmeyecektir- çalıştırılmayacaktır(execute) ve hiçbir yolla bu belirli prosedürün doğru çalışıp çalışmayacağını tespit edemeyeceksiniz. Düzgün proje tasarımında başka bir adım, projenizi tam olarak test etmenizdir. Çalıştığınız her olayın hedeflediğiniz gibi çalıştığından emin olun. Projenizi durdurun(Visual Basic toolbar içinden Stop butonunu tıklayın). Eğer bir değişiklik yaptı iseniz, projenizi (Save) saklayın.

Denemeniz Gereken Diğer Bazı Şeyler

Bu derslerdeki her proje için, bazı denemeler yapabilmeniz için birtakım değişiklikler konusunda tavsiyelerde bulunacağız. Shrink Form ve Grow Form butonlarını, formu ekran etrafında hareket ettirmek üzere değiştirin (Left ve Top özelliklerini kullanın).Daha fazla sayıda kullanılabilir renkleri, formlara uygulamak üzere daha evvel tanımladığımız sembolik sabitlerden yararlanarak kullanın.

Hide Buttons butonunu öylesine ayarlayın ki, o sadece komut butonlarının ' Enabled özelliğini False'a çevirsin (Visible(görünürlük) özelliği değil).

Benzer şekilde Show Buttons butonunu da modifiye edin (değiştirin).

Özetle,

Tebrikler! Şimdi daha karmaşık bir Visual Basic projesini tamamladınız

(en azından birden fazla kontrol vardı). Proje dizaynı hakkında bilgiler edindiniz, projeleri saklamayı, komut butonu kontrollerini, formların detaylarını ve komple bir projenin nasıl yapılandırılacağını artık öğrendiniz.

Şimdi bir proje yapılandırmanın üç adımı ile ilgili olarak artık rahat olmalısınız:

Kontrollerin yerleşimi, özelliklerin (property) düzenlenmesi ve olay prosedürlerin yazılması.

Bu adımları, ilerleyen derslerimizde, yeni kontroller ve daha fazla BASIC dili kullanarak, diğer projeleri yapılandırmada kullanmaya devam edeceğiz.

Herkes İçin
Visual Basic

5. Etiketler (Labels), Metin Kutuları (Text Boxes), Değişkenler (Variables)

Tekrar Gözden Geçirme ve Bir Önbakış

Bu derste Visual Basic ortamını, bazı yeni kontrolleri ve birtakım yeni BASIC bildirilerini görmeye devam edeceğiz. Bu derslere devam ettikçe, Visual Basic projesini yapılandırmanın daima üç adımı olduğunu unutmayın:

(1) form üzerine kontrolleri yerleştirin (2) kontrollere özellikler atayın (3) ilgili olay prosedürlerini yazın. Bu derste, projelerinizdeki hatalarınızı nasıl bulup, bu hataları nasıl temizleyeceğinizi, label(etiket), text box controls(metin kutusu kontrolleri) ve BASIC variables(değişkenlerini) öğreneceksiniz. Yine bu derste saklamayı planladığınız şeyleri saklamanıza yardımcı olacak bir proje yapılandıracaksınız.

Bir Visual Basic Projesinin Hatalarını Ayıklamak

Projenizi ne kadar iyi planlarsanız planlayın, kontroller ve olay prosedürleri içine fikirlerinizi yerleştirirken ne kadar dikkatli olursanız olun, hatalar yapacaksınız !

Hatalar veya bilgisayar programcılarının söylediği şekilde bugs (hatalar) projelerinize sessizce sızacaklardır. Bir programcı olarak bu bugs(hatalar)'ı bulmak ve ayıklamak için bir takım stratejilere ihtiyaç duyacaksınız. Bir proje içindeki hataları bulup ayıklama metotlarına debugging adı veriyoruz. Ne yazık ki, bir program içindeki hataları bulmak için çok sayıda güçlü ve hızlı kurallar bulunmamaktadır.

Her programcının hatalara karşı geliştirdiği kendisine has bir yolu bulunmaktadır. Siz de kendi yollarınızı geliştireceksiniz. Bu derste size ancak bazı genel strateji ve düşünceler sağlayacağız.

Proje hataları (veya bugs), üç tipte sınıflandırılabilir:

- Syntax errors (sözdizimi hataları)
- Run-time errors (çalışma süresi hataları)
- Logic errors (mantıksal hatalar)

Syntax errors(sözdizimi hataları), dizayn(design) modunda bir özelliğin değerini düzenlerken veya bir BASIC kodu satırı yazarken oluşur. Bir şey bozuk veya yanlış yazılmış olabilir veya orada bulunması gereken bir şey orada bulunmamaktadır. Eğer sözdizimi hataları varsa, projeniz

çalışmayacaktır. Run-time errors (çalışma süresi hataları) projenizi çalıştırmaya kalkıştığınızda oluşur. Çalışması birdenbire durur, çünkü kontrolü arkasında bir şeyler olmuştur. Logic errors(mantıksal hatalar)'ı bulmak oldukça zorlayıcıdır. Projeniz güzel çalışır, fakat verdiği sonuçlar beklediğinizden farklıdır.

Hadi her hata tipi üzerine alıştırma yapalım ve muhtemel debugging metotlarını tespit edelim.

Syntax Errors(Söz Dizimi Hataları)

Söz Dizimi Hataları(Syntax Errors) tanımlanması ve ayıklanması en kolay olan hatalardır. Visual Basic Program'ı, syntax error'ları bulmak için en büyük yardımcıdır. Syntax error, en çok kontrol özelliklerine değer düzenlerken veya olay prosedürleri için BASIC kodu yazarken ortaya çıkar.

Visual Basic'te yeni bir projeye başlayın. Proje penceresine gidin ve formun Left özellik değeri kısmına yelkenli kelimesini yazın. Ne oldu ? Aşağıdaki gibi küçük bir pencere göreceksiniz (bu bir VB4 penceresidir, diğerleri ise benzerdir):



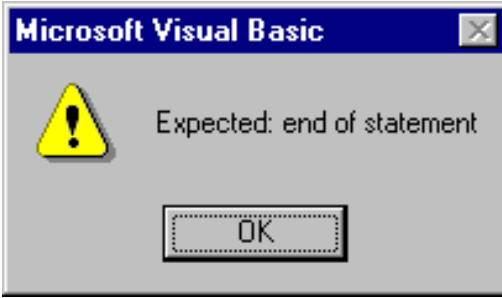
On-line help için <F1> tuşuna basın ve problemin bir açıklamasını görün. Hatırlayın bu özellik değeri uygun tipte olmalıdır. Bir özelliğe uygun olmayan tipte bir atama yapmak bir syntax error'dur. Fakat, bakın Visual Basic bizim bu hatayı yapmamızı engelledi. OK tıklayın. Left değeri onu değiştirmek istediğinizden önceki durumuna dönecektir.

Kod yazarken eğer bir syntax error yaparsanız ne olur ? Deneyelim !

Form_Load prosedürü için kod penceresini açalım. Başlık satırı altına, şu satırı yazın ve <Enter> tuşuna basın:

```
Form1.BackColor 0 vbRed
```

Bir atama bildirisinde eğer = işareti yerine eğer yukarıdaki gibi 0 yazarsanız, aşağıdaki pencere görünür (yine bu bir VB4 penceresidir):



(Beklenen: Bildiri Sonu)

<F1> tuşuna basmak size bazı yardımlar sağlayacaktır. Visual Basic bu bildiride bir şey(ler)in yanlış olduğunun farkına vardı. Bunun ne olduğunu görebilmelisiniz.

Böylelikle eğer bir syntax error yaparsanız, Visual Basic genellikle yanlış bir şey yaptığının farkına varır ve hata yaptığının farkına varmanızı sağlar. Syntax error'larınızın ayıklanması için iyi bir kaynak on-line help sistemidir.

Şunu da belirtelim, syntax error'ler genellikle klavyeden hatalı yazı girilmesinden kaynaklanır. Bu ise klavyeden yazım becerinizi geliştirmeniz için diğer bir önemli nedendir.

Run-Time Errors(Çalışma Süresi Hataları)

Başarılı bir biçimde kontrol özelliklerini düzenleyip olay prosedürlerini yazınca ve bütün tanımlanan syntax error'lar da ayıklayınca, projenizi artık çalıştırmayı deneyeceksiniz. Eğer proje çalışırsa çok iyi ! Fakat pek çok kereler projeniz durup size bir hata bulunduğunu söyleyebilir – bu bir run-time error (Çalışma Süresi Hatası)'dır. Neden durduğunu araştırıp problemi tespit etmeniz gerekmektedir. Yine, Visual Basic ve on-line help genellikle run-time error'ları ayıklamanız için yeterli bilgiyi sağlayacaktır.

Örneklere bakalım. Yukarıdaki aynı örnek ile çalışarak, `Form_Load` prosedüründeki kod satırını aşağıdaki gibi değiştirin:

```
Form1.BackColor vbRed
```

Buradaki örnekte formun background rengini düzenlerken = işaretini unutmuş gibi davranalım. Dikkat edin, Visual Basic size bu hatalı satırı yazmanız için izin verir. Visual Basic, dizayn modunda bütün syntax error'ları bulamaz. Projeyi çalıştırmayı deneyin (burada herhangi bir dosyayı saklamanız gerekmiyor).

Araç çubuğu üzerinde bulunan Start buton'unu tıkladığınızda , aşağıdaki pencere görünmelidir (VB4 sürümü gösterilmiştir):



(Özelliğin Geçersiz Kullanımı)

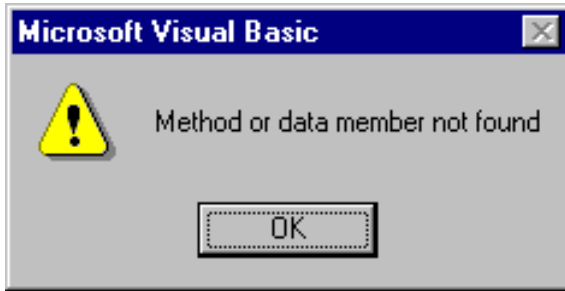
ve kod penceresinde bulunan `BackColor` kelimesi belirgin hale gelir. Visual Basic, bu belirli özelliği nasıl kullandığınız ile ilgili olarak size bir şeylerin yanlış olduğunu söylemektedir. Daha fazla bilgiye ihtiyaç duyarsanız <F1> tuşuna basın.

Visual Basic size hatalarınızı göstermesi açısından, yeteri kadar iyidir.

Varsayalım hatamızı bir = işareti koyarak düzelttik, fakat yanlışlıkla bu sefer de `BackColor` özellik ismi içindeki `k` harfini kaldırdık ve şu şekilde yazdık :

```
Form1.BacColor = vbRed
```

Projeyi çalıştırmayı deneyin ve aşağıdaki pencereyi görün (VB4 sürümü görünmektedir):



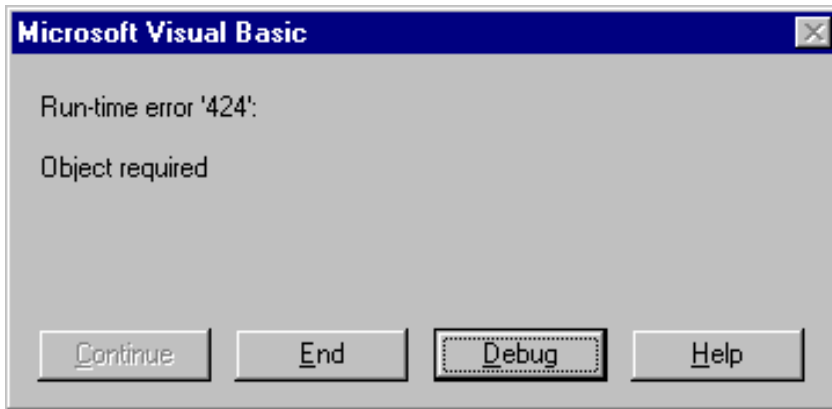
(Metot veya veri elemanı bulunamadı)

ve BacColor kelimesi belirgin hale gelir. Yardıma ihtiyacınız varsa <F1> tuşuna basın. Visual Basic size bu belirli kontrol için bu özelliği bulamadığını söylemektedir. Hatalı yazımı farkedip düzeltmeniz gerekmektedir.

Şimdi, özellik ismini düzeltin fakat bu sefer Form1 yazacağınız yere For1 yazın:

`For1.BackColor = vbRed`

Projeyi çalıştırın. Yeni bir pencere ortaya çıkacaktır:



(Run-time error '424'

Nesne gerekiyor)

Bu mesaj penceresi diğerleri kadar faydalı değildir. <F1> tuşuna basmak veya Help'i tıklamak hatayı açıklayacak, fakat hatanın nerede olduğunu söylemeyecektir. Buradaki anahtar mesaj kelime 'Object required'(Nesne gerekiyor)'dur. Bu genellikle BASIC kodu içinde bir kontrolün ismini atarken yanlış girilmesi sonucu ortaya çıkan mesajdır. Visual Basic birşeye dot notation (nokta notasyon) kullanarak bir özellik atamaya çalışıyor :

`ControlName.PropertyName = Value`

Fakat, verilen isimde bir kontrol bulamamaktadır (örneğimizde For1). Hatanın nerede olduğunu nasıl bulabiliriz ? Bir ipucu: bu pencerede Debug buton' u tıklanarak bulunabilir. Visual Basic durduğu kod satırını işaretler. Deneyin ve kötü kod satırını, bir kutucuk içine alınmış şekilde görün. Hatalı yazılmış form ismini sizin bulup çıkarmanız gerekmektedir - Visual Basic bunu sizin için yapamaz. Debug buton'unu tıkladığınızda, Visual Basic diğer mümkün olan başka bir moda sıçrar – break(ara ver) modu (design ve run modlarını gördünüz). Break mod'dan ayrılmak ve design mod'a dönmek için, yani kodunuzu düzeltmek için, projeyi durdurmanız gerekmektedir.

Burada sebep olduğumuz hatalar run-time error'lar içinde en genel olan üç tanesidir: hatalı yazım ve bunu bir isim özelliğine atamak, bir özellik ismini hatalı yazmak, veya bir atama bildiriminde bir şeyi eksik bırakmak.

Başkalarıda vardır ve başka projeleri yapılandırmaya başladıkça bunların pek çoğu ile karşılaşacaksınız. Fakat gördünüz. Visual Basic hatanın nerede olduğunu tespit ederken genellikle iyidir ve on-line help daima bunları açıklamak için kullanılabilir durumdadır. Run-time error'lar hakkında son bir şey; Visual Basic bütün hataların hepsini bir kerede bulamaz. Karşı karşıya kaldığı ilk run-time error'da durur.

Siz bu hatayı giderseniz bile, daha başka hatalar da bulunabilir.

Run-time error'ları, yalnızca, teker teker düzeltebilirsiniz.

Logic Errors(Mantıksal Hatalar)

Mantıksal hatalar bulunması ve ayıklanması en zor hatalardır. Bu tip hatalar projenizi çalışmaktan alıkoymazlar fakat doğru olmayan ve beklenmedik sonuçlar veren hatalardır. Bu noktada yapabileceğiniz tek şey, eğer mantıksal hataların varlığından şüpheleniyorsanız, projenize iyice dalmak (ilk olarak olay prosedürleri) ve her şeyin kesin olarak istediğiniz şekilde kodlandığından emin olmaktır. Mantıksal hatalar bir zaman tüketme sanatıdır, bir bilim değildir. Mantıksal hatalar (logic errors) bulmanın belirli kuralları yoktur. Her programcı, mantıksal hatalarını arayıp bulmak için, kendi yolunu bulup geliştirmelidir.

Kullandığımız örnek üzerinde bir mantıksal hata; kırmızı olmasını beklediğiniz halde form background rengini maviye ayarlamaktır. Buradan kodlara gidip neden öyle olduğuna bakabilirsiniz. İstenilen vbRed sembolik sabit yerine vbBlue sembolik sabitinin kullanıldığını görebilirsiniz.

Değişikliğin yapılması mantıksal hatayı giderir ve form kırmızı olur.

Ne yazık ki, mantıksal hataları ayıklamak bu örnekte olduğu kadar kolay değildir. Fakat bir yardım bulunmaktadır. Visual Basic, debugger adı verilen bir araca sahiptir ve mantıksal hataları tanımlamaya bu yolla yardımcı olur.

Debugger kullanarak (daha evvel gördüğümüz gibi break modunda çalışır), özellikleri yazıcıda bastırılabilir, kodlarınızı nerede ve ne zaman isterseniz durdurabilir ve projenizi satır-satır çalıştırabilirsiniz. Debugger'ın kullanımı, bir ileri seviye başlık olup bu derste anlatılmayacaktır. Eğer Visual Basic becerilerinizi arttırmak istiyorsanız, aynı zamanda debugger'ın nasıl kullanılacağını da öğrenmeniz gerekmektedir.

Şimdi Visual Basic ile ilgili becerilerimizi arttıralım. Yeni iki kontrole bakacağız:

Label(etiket) ve text box(metin kutusu).

Label(Etiket) Kontrolü

Bir label(etiket), kullanıcının direk kontrol edemeyeceği bilgiyi gösterir.

Bu genellikle diğer kontrollere başlıklar sağlamak üzere kullanılır. Veya , bazı bilgisayar işlemlerinin sonuçlarını göstermek üzere kullanılır. Label kontrolü araç kutusu içerisinde seçilir. Şu şekilde görünür:

*Araç Kutusunda :
(default) özellikler) :*



Form Üzerinde (varsayılan

Özellikler

Etiketler için birkaç faydalı özellik aşağıdaki gibidir:

<u>Özellik</u>	<u>Açıklama</u>
Name	<i>label(etiket)'i tanımlamak üzere verilen isim. Etiket isimleri için kullanılan ve üç harften oluşan örnek lbl'dir.</i>
Caption	<i>Metin (string tipte) ve etiket içinde görünür.</i>
Font	<i>Caption metninin stil, boyut ve tipini düzenler.</i>
Alignment	<i>Caption metninin sola yaslı, sağa yaslı veya ortada olmasını belirleyen özelliktir (hizalama).</i>
BackColor	<i>Etiket'in background(artalan) rengini düzenler.</i>
ForeColor	<i>Caption metninin rengini düzenler.</i>
BorderStyle	<i>Label sınırlarının stilini(tipini) düzenler.</i>
Left	<i>Formun sol kenarından, etiketin sol kenarına kadar olan uzaklık.</i>

Top	<i>Formun üst noktasından etiketin üst noktasına kadar olan uzaklık.</i>
Width	<i>Twip birimi ile etiketin genişliği</i>
Height	<i>Twip birimi ile etiketin yüksekliği</i>
Visible	<i>(run modunda) etiketin form üzerinde görünüp görünmemesini belirler</i>

Örnek

Visual Basic'i çalıştırın. Yeni bir projeye başlayın. Form üzerine bir label koyun. İsterseniz bunu yeniden boyutlandırıp hareket ettirin. Caption özelliğini düzenleyin. Değişik fontları deneyin