

VISUAL BASIC 6.0

Visual Basic karmaşık mantıklara ve biçimlere sahip, Windows uyumlu uygulamaların tasarımlarını kolaylaştırıcı bir görsel dildir. Bu dilde program geliştirmenin ilk adımı, kullanıcının ekranda görmek istediklerini saptamak ve ekran tasarımını gerçekleştirmektir. Bu programda her şey nesne olarak tanımlanmıştır (form, yazı kutusu, komut düğmesi v.b.). Nesnelere, kendilerine iliştilmiş olan olayları (farenin tıklanması, farenin çift tıklanması gibi) tanıyabilir ve yazılmış olan komutlara göre tepki verebilir. Olaylar gerçekleştiğinde işlemler yerine getirilir, ve pencereler güncelleştirilir. Sistem daha sonra yeni olayları bekler.

TEMEL KAVRAMLAR

Form: Windows pencerelerinin Visual Basic'deki karşılığıdır. Uygulama çalıştırıldığında formlar pencerelere dönüşür. Her uygulama en az bir form içermek zorundadır.

Prosedür: Komut cümlelerinden oluşan program parçalarıdır.

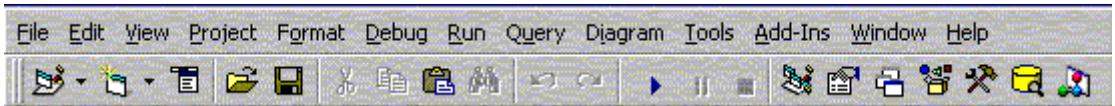
Modül: Tek bir dosyada toplanan ve programın herhangi bir yerinden çağırılabilen prosedürler ve tanımlamalar topluluğu.

Özellik (Property): Nesne karakteristiklerini (boyut, renk, konum, ...) ve nesne davranışlarını tanımlayan nitelikler.

Olay (Event): Nesnelere algılayabildiği eylemlerin adıdır.

ARAÇ ÇUBUĞU VE KISAYOL TUŞLARI

Şekil-1'de araç çubuğu ve üzerindeki kısayol tuşları görüntülenmektedir.



Şekil-1. Araç Çubuğu

Araç çubuğu üzerindeki kısayol tuşlarının işlevleri soldan sağa aşağıdaki gibidir:

- Standart EXE Proje Ekleme - File/Add Project (File Menüsünden Add Project seçeneği tıklanır)
- Form Ekleme - File/New Form
- Menü Tasarımı - Window/Menu Design - Ctrl+M
- Proje Aç - File/Open Project
- Proje Kaydet - File/Save Project
- Kes - Edit/Cut - Ctrl+X
- Kopyala - Edit/Copy - Ctrl+C
- Yapıştır - Edit/Paste - Ctrl+V
- Bul - Edit/Find - Ctrl+F
- Geri Al - Edit/Undo - Ctrl+Z
- Tekrar - Edit/Redo
- Başlat - Run/Start - F5
- Ara Ver - Run/Break - Ctrl+Break
- Sona Erdir - Run/End
- Proje Gezgini - View/Project Explorer
- Özellikler Penceresi - View/Properties Window - F4
- Form Görüntü Penceresi - View/Form Layout Window
- Nesne Gezgini (tüm nesnelere gösterir) - View/Object Browser
- Araç Kutusu (kontrol nesnelere içerir) - View/Toolbox
- Veri tabanı Bağlantıları - View/Dataview Window

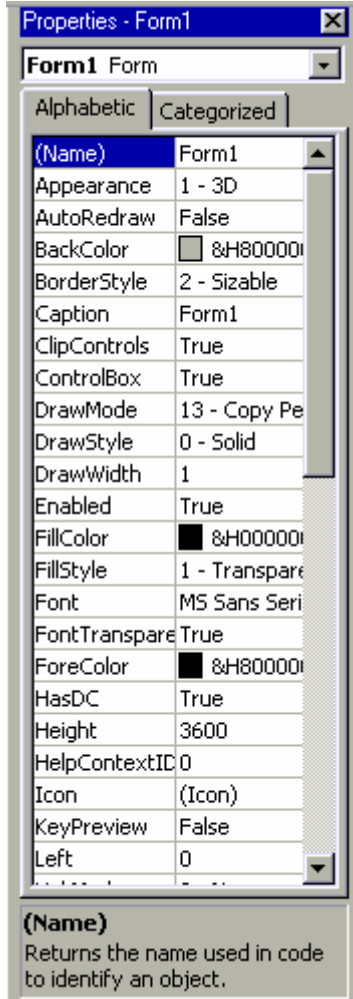
Proje ve Form Penceresi

Tüm Visual Basic bileşenleri bir proje içerisinde bulunur. Proje yaratıldıktan sonra, proje penceresinde, o uygulamada yer alan formlar, modüller ve özel denetçiler listelenir. Listedenden istenilen form seçildikten sonra View/Form ile formun kendisi ya da View/Code ile o forma ait kod görüntülenebilir. Formun üzerinde yer alan noktalara grid denir. Bu noktalar form üzerine yerleştirilecek nesnelere hizalanmasını sağlar.

ÖZELLİKLER PENCERESİ

Bir formun konumunu değiştirmek onun özelliğini değiştirmek anlamına gelir. Özellikler penceresi açıldığında Şekil-2'deki gibi görüntülenir. Başlık çubuğu altındaki

nesne kutusunda (Object Box) yer alan açıklama nesnenin adını ve ne tip bir nesne olduğunu belirtir. Örneğin Şekil-2'deki "Form1 Form" açıklaması seçili nesnenin bir form olduğunu ve adının Form1 olduğunu göstermektedir. Nesne kutusunu altındaki ayarlar kutusunda (Settings Box) ise pencerede o anda seçilmiş olan ayarın değeri görüntülenir ve değiştirilebilir. Seçeneklerin çoğunluğu kullanıcıya düşen liste olarak sunulur. Bu liste renkler için paleti görüntülerken, pencerenin görünür olma özelliği için ise doğru/yanlış (True/False) olarak sunulur.

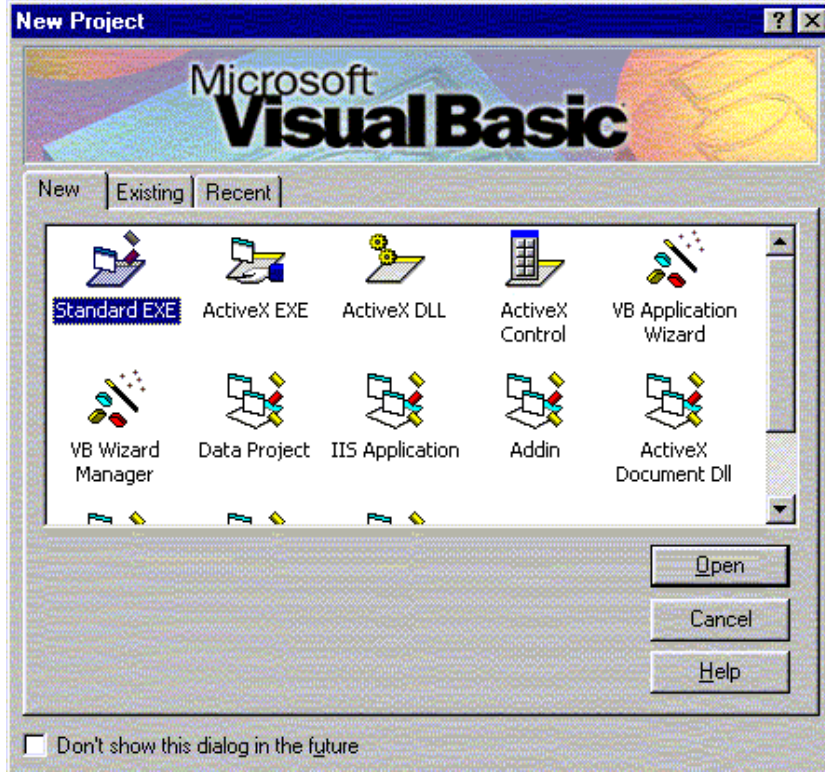


Bazı Form Özellik Tanımları
(Name): İsim
Appearance: Formun düz ya da 3-boyutlu görünmesini sağlar.
BackColor: Artalan rengi
BorderStyle: Kenar çizgisi biçimi
Caption: Form Başlığı
ControlBox: Pencere kontrollerinin olup olmayacağını belirler.
Font: Yazı tipini belirler.
Height: Formun yüksekliğini belirtir.
Left: Formun sol koordinatını belirler.
Top: Formun üst koordinatını belirler
Visible: Formu gizlemek ya da gizli formu görüntülemek için kullanılır.
Width: Formun genişliğini belirtir.

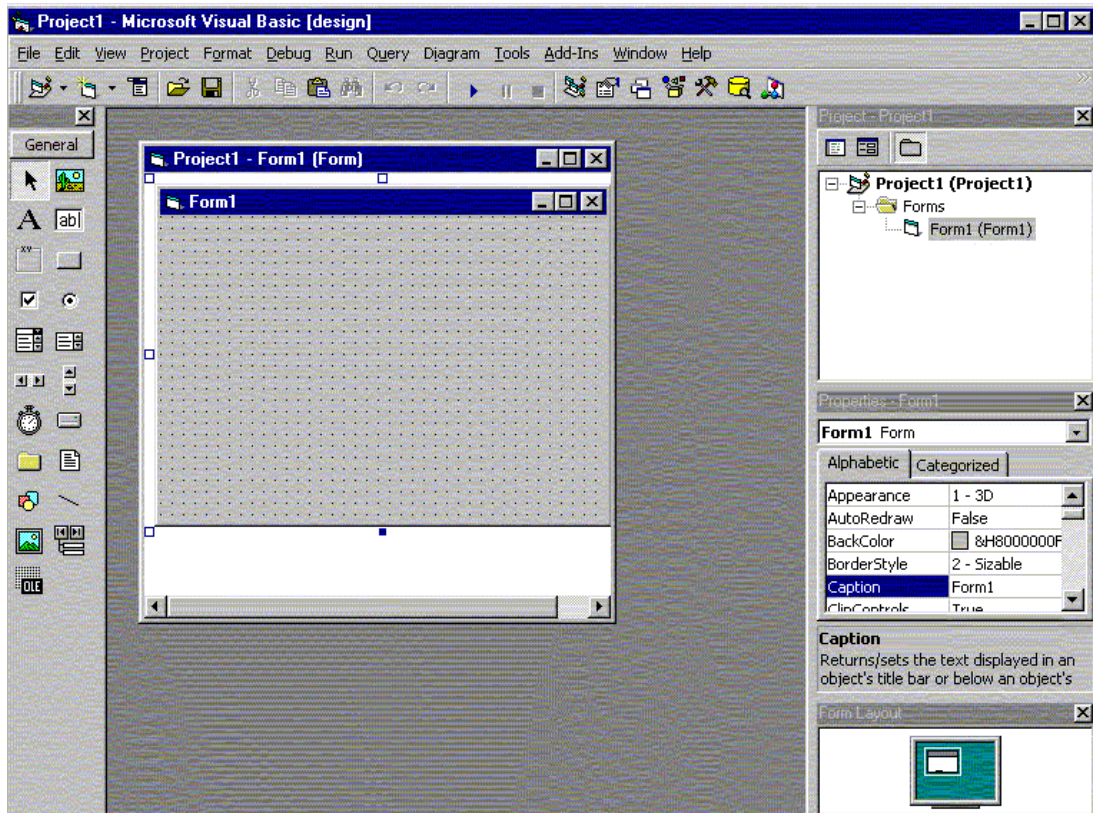
Şekil-2. "Properties" Özellikler Penceresi

Programı Başlatma

Visual Basic programını başlattığınızda ekrana Şekil-3'deki Yeni Proje Penceresi gelir. Eğer Visual basic daha önceden çalışıyor ise aynı pencereyi File/New project seçeneği ile de açabilirsiniz. Bu pencere seçenekleri ile proje tipini belirleyebilirsiniz. Normal proje için Standard.EXE, İnternet ile ilgili projeler için Activex.EXE kullanınız.



Şekil-3. Yeni Proje Diyalog Penceresi



Şekil-4. Proje Tasarım Penceresi

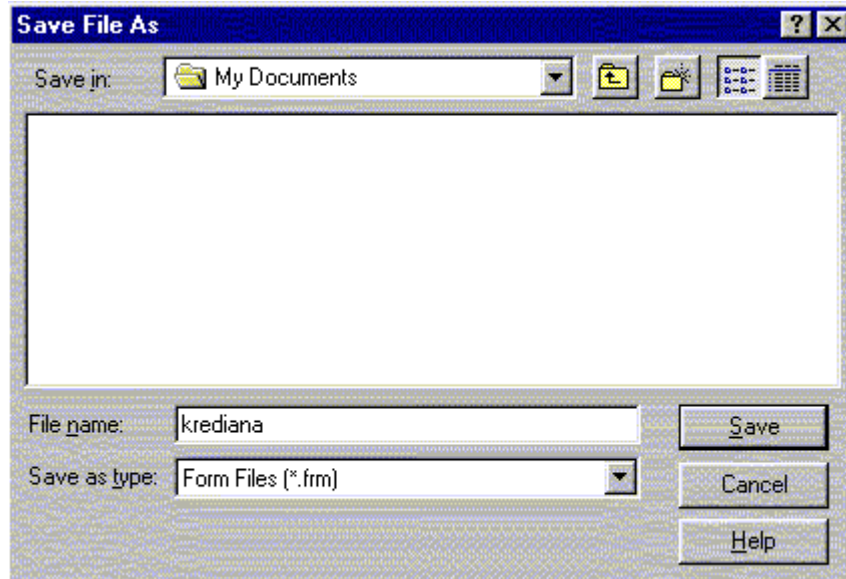
Proje açıldıktan sonra proje tasarım penceresi Şekil-4'deki gibi ekrana gelir. Bu pencerenin başlığı Project1: Microsoft Visual Basic (Design) olarak görüntülenir. Daha sonra bu program çalıştırıldığında Design-Run olarak değişecektir.

Bu yeni proje bir form ile başlar. Genelde bu form programınızın başlangıç sayfası olur. Form penceresi dışında Tasarım Penceresi üzerinde projede yer alan formları gösteren "Project1" adlı proje penceresi, formun ekran üzerindeki boyutunu gösteren "Form Layout" penceresi ile nesne özelliklerini gösteren "Properties" penceresi yer alır. Tasarım penceresinin sol tarafında ise nesne yaratımı için gerekli bir araç çubuğu bulunur.

Projeyi Saklama

Projenin saklanması için öncelikle projenin bileşenleri (form, kod modülü, vs.) ayrı ayrı ait oldukları dosyalara saklanmalı, daha sonra projenin kendisi proje dosyası olarak saklanmalıdır.

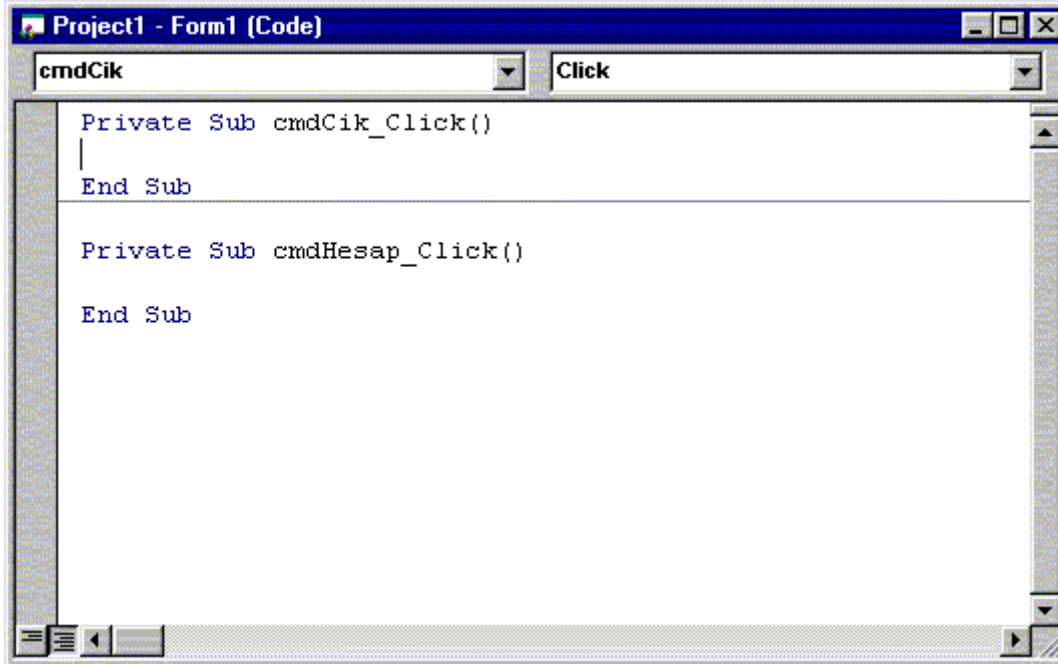
Projeyi saklamak için File/Save Project menü seçeneği ya da araç çubuğu üzerindeki Save Project düğmesi tıklanır. Her proje bileşeni için ardı ardına Şekil-5'deki Yeni Adla Proje Saklama Diyalog kutusu ekrana gelecektir. Bu kutu üzerinde her dosyanın adını ve yerini belirlemeniz gerekmektedir.



Şekil-5. Yeni Adla Proje Saklama Penceresi

Formları Programlama

Formlar 20'den fazla olaya tepki verebilirler. Bu tepkinin ne olacağını kodlama ile belirtebilirsiniz. Örneğin, kullanıcı form üzerindeki herhangi bir bölgeyi tıkladığında bir mesaj görüntüleyebilirsiniz. Kodlamayı yapabilmek için formu çift tıklamak ya da View/Code komutu kullanmak yeterlidir. Bu komut çalıştırıldığında Şekil-6'daki pencere ekrana gelir. Bu pencerenin üst sağındaki "Object" kısmında formda yer alan tüm nesnelerin listesi, soldaki "Proc" kısmında ise nesnenin tanıyacağı olaylar bulunur.



Şekil-6. Form Kod Penceresi

Form tıklandığında "Fareyi tıkladınız" mesajını vermek için gerekli kod aşağıdadır:

```
Sub Form_Click  
    Print "Fareyi tıkladınız."  
End Sub
```

Örnek Form Olayları	
Activate	Formun etkin pencere olması
Click	Form üzerinde farenin tıklanması
DblClick	Form üzerinde farenin çift tıklanması
Deactivate	Formun edilgen olması
GotFocus	Form denetçilerinin edilgen, formun etkin olması
KeyDown	Odak form üzerinde iken klavye tuşuna basılması
KeyPress	Odak form üzerinde iken klavye tuşuna basılıp bırakılması
KeyUp	Odak form üzerinde iken herhangi bir tuşa basılması yada bırakılması
Load	Form penceresinin açılması
LostFocus	Odağın formdan başka bir nesneye kayması
MouseDown	Farenin bir tuşuna basılması
MouseMove	Farenin hareket ettirilmesi
MouseUp	Farenin basılan tuşunun bırakılması
Unload	Formun kapatılması

Örnek Form Yöntemleri	
Circle	Form üzerine daire, elips ya da yay çizer
Cls	Form üzerindeki tüm yazı ve resimleri siler
Hide	Formu gizler fakat kapatmaz
Line	Form üzerinde çizgi ya da dikdörtgen çizer
Move	Formu hareket ettirir
Print	O andaki renk ve yazı boyunu kullanarak form üzerine yazı yazar
PrintForm	Formu ekranda görüldüğü gibi yazıcıya yollar
Refresh	Formun kendini yeniden çizmesini sağlar
Scale	Formun koordinat sistemlerini tanımlar
SetFocus	Odağın forma kaymasını sağlar
Show	Formun görüntülenmesini sağlar
TextHeight	Yazı boyunu belirler
TextWidth	Yazı uzunluğunu belirler

Kullanıcı Etkileşimi

Kullanıcıdan bilgi almak için kontrol nesneleri kullanılır. Bu kontrol nesneleri Şekil-7'de görüntülenmektedir.



Gösterici (Pointer)	Resim Çerçevesi (PictureBox)
Etiket (Label)	Yazı Kutusu (TextBox)
Çerçeve (Frame)	Komut Düğmesi (CommandButton)
İşaret Kutusu (CheckBox)	Seçenek Düğmesi (OptionButton)
Çekme Liste (ComboBox)	Liste (ListBox)
Yatay Kaydırma Çubuğu (HScrollBar)	Düşey Kaydırma Çubuğu (VScrollBar)
Zamanlayıcı (Timer)	Sürücü Listesi (DriverListBox)
Dizin Listesi (DirListBox)	Dosya Listesi (FileListBox)
Şekil (Shape)	Çizgi (Line)
Resim (Image)	Veri Denetçisi (Data)
OLE Denetçisi (OLE)	

Şekil-7. Kontrol Nesnelерinin Yer Aldığı Araç Çubuğu

Bir Nesnenin Taşınması ve Boyutlandırılması

Bir nesnenin taşınma ve boyut değiştirme işlemleri grafik nesnelерinki ile aynıdır. İstenirse birden fazla nesne Ctrl tuşu basılı tutularak seçilir ve grup halinde taşıma ya da boyut değiştirme işlemi gerçekleştirilebilir.

Etiket

Kullanıcıya bilgileri ve yazıları görüntülemek için kullanılır, burada bilginin kullanıcı tarafından değiştirilmesi olası değildir.

Etiket Ekleme

Yazı kutularının işlevini belirlemek için her yazı kutusu ile birlikte bir adet etiket kutusuna gereksinim vardır.

- Araç çubuğu üzerinde etiket kutusu düğmesini tıklayınız.
- Farenin imlecini form üzerine, kutuyu koyacağınız yerin bir köşesine götürünüz.
- Fareyi tıklayarak parmağınızı bırakmadan kutunun karşı köşesi olacak yere kadar sürükleyiniz, ve fare düğmesini bırakınız.

Etiket kutusunun özellikleri "Properties" penceresinde görüntülenir. Nesnenin adı "Name" ile belirlenir, etiketin ekranda gözükecek yazısı ise "Caption" bölümüne yazılır.

Yazı Kutusu

Kullanıcıdan yazılı olarak bilgi almak ve yine bilgileri kullanıcıya göstermek için kullanılır. Yazı kutusunun özellikleri yine "Properties" penceresinden değiştirilir. Yazı kutusuna girilen bilgi sayısal bilgi de olsa karakter dizini olarak kabul edilir. Aritmetik işlem yapılacağı takdirde sayıya çevrilmelidir (Val komutu ile). Yazı kutularının ismi vardır ancak başlık - "Caption" özelliği bulunmaz.

Yazı Kutusu Ekleme

- Araç çubuğu üzerinde yazı kutusu düğmesini tıklayınız.
- Farenin imlecini form üzerine, kutuyu koyacağınız yerin bir köşesine götürünüz.
- Fareyi tıklayarak parmağınızı bırakmadan kutunun karşı köşesi olacak yere kadar sürükleyiniz, ve fare düğmesini bırakınız.

Komut Düğmesi

Kullanıcının tıklaması ile çeşitli işlevlerin çalıştırmasını sağlayan düğmelerdir. Komut düğmelerinin üzerine komutun işlevini belirten yazılar başlık-Caption özelliği ile iliştilir. Başlık yazısı düğmenin ortasına yazılır. Başlık düğmenin standart boyundan geniş ise boyutlandırma otomatik olarak yapılmaz. Bunu sizin yapmanız gerekir. Başlık üzerinde bir karakterin altını çizmek için o karakterin önüne & işareti konur, ve o karakter altı çizili olarak yazılır. Böylece komut Alt tuşu ile altı çizili karakterin klavyeden basılması ile de çalıştırılır.

&Son -- Son -- Alt+S ile Son komutu gerçekleşir ve program sona erer

Komut Düğmesi Ekleme

- Araç çubuğu üzerinde komut düğmesi düğmesini tıklayınız.
- Farenin imlecini form üzerine, düğmeyi koyacağınız yerin bir köşesine götürünüz.
- Fareyi tıklayarak parmağınızı bırakmadan düğmenin karşı köşesi olacak yere kadar sürükleyiniz, ve fare düğmesini bırakınız.

PROGRAMLAMA TEMELLERİ

Cümleler

Bir cümleyi yazıp Enter tuşuna bastığınızda o komut cümlesi değerlendirilir ve yazım yanlışları saptanır. Büyük küçük harf ayrımı yapılmaz. Komutun ilk harfi otomatikman büyütülür. Komut olmayan açıklama satırları için tek tırnak, Rem ya da Remark kullanılır.

Rem Yardım Düğmesi

' Bu düğme tıklanınca çalışır.

Print "Yardım yok" 'sadır arasına da açıklama koyabilirsiniz

Değişkenler

Kullanıcıdan alınan bilgiler ve hesaplama sonuçları değişken denilen geçici bölgelerde tutulur. Değişken adları en fazla 255 karakter olabilir. Değişken adı bir alfabe karakteri ile başlamalıdır. Değişkenin özellikleri ise aşağıda verilen listedeki gibi belirlenir:

s -String	:	Karakter dizisi
n - Integer	:	Tamsayı (-32768 +32767)
l - Long	:	Uzun Tamsayı (yaklaşık +/- 2 milyar)
f - Single	:	Tek Duyarlı Ondalıklı Sayı (floating)(1E-45, 3E38)
d - Double	:	Çift Duyarlı Ondalıklı Sayı (floating)(5E-324)1.8E308
c - Currency	:	Para değeri (\$) (xxxxxxxxxxxxxxxxxxxx.xxxx)
b - Boolean	:	Doğru/Yanlış (true-false 0/-1)
v- Variant	:	Değişken
d - Date	:	Gün ve saat bilgisi (1/1/100 den 12/31/9999 a kadar)

Değişken Tanımları

Bir formda tanımlanan değişken o formda yaratılacak tüm prosedürlerde kullanılabilir. Bir prosedürde Dim ya da Private ile yerel olarak tanımlanan değişken ise sadece o prosedür için tanımlı kalır. Genel tanımların kısıtlanması kod hatalarının ayıklanması açısından kolaylık getirir. Bu nedenle yalnızca prosedür içinde kullanılacak değişkenler yerel olarak tanımlanmalıdır. Heryerde (formlar arası) kullanılacak değişken tanımlaması ise Public ile tanımlanmalıdır.

Dim değişkenadı [As deęişkentipi] [, deęişkenadı2 [As deęişkentipi2]]
Private deęişkenadı [As deęişkentipi] [, deęişkenadı2 [As deęişkentipi2]]
Public deęişkenadı [As deęişkentipi] [, deęişkenadı2 [As deęişkentipi2]]

Buna göre dolar olarak gösterilecek para deęişkeni tanımı Őu olabilir:

```
Dim cPara As Currency
```

Atama ve Özellik Belirleme

Deęişkenlere deęer ataması için "=" ya da "Let" deyimi kullanılır. Deęişkenin adı eđitlięin saę tarafında, deęeri ise sol tarafında olmalıdır.

```
Faiz = .6
```

```
Let Faiz = .6
```

```
Yardım.Caption = "&Yardım" 'Yardım düęmesinin bađlığını belirler
```

```
Yardım.Left = Yardım.Left-200 ' Yardım düęmesini 200 piksel sola alır
```

```
Yardım.Visible = False ' Yardım düęmesini gizler
```

```
Yardım.TabStop = False ' Tab ile bu düęmeye geçilmez
```

```
' Alttaki komut yardım düęmesi gizli ise görüntüler, deęilse gizler
```

```
Yardım.Visible = Not(Yardım.Visible)
```

Sayılar ve İşlemler

Tamsayı (Integer) deęişkenlere reel sayı atandığında tamsayıya yuvarlanarak atanır. Deęişken tipine uymayan bir atama yapılırsa ise sistem uyarı verir.

```
Dim Sayi As Integer
```

```
Dim Yil As String
```

```
Sayi = 50000 ' Yanlıđ, en fazla 32768.
```

```
Sayi = 32.2 ' 32 olarak atanır
```

```
Sayi=Val("1994") ' 1994 deęeri atanır
```

```
Yil = Str$(1994) ' "1994" deęeri atanır
```

Aritmetik işlemlerde parantezlerin öncelięi vardır. Parantez yok ise yukarıda belirtilen öncelik sıraları uygulanır. Aynı öncelik sırasına sahip olan işlemler soldan saęa doęru gerçekleştirilir.

Aritmetik Operatörler ve Öncelik Sırası		
1	^	Üs alma
2	-	Negatif sayıları gösterme
3	*	Çarpma
3	/	Bölme
4	\	Tamsayı bölme
5	Mod	Kalanı bulma ($5 \text{ Mod } 2 = 1$) (Tamsayı)
6	+	Toplama
6	-	Çıkartma

Farklı tipte iki sayıya işlem yapıldığında sonuç sayılardan daha duyarlı olan tipte olur. Duyarlılık sırası Integer, Long, Single, Double ve Currency'dir.

Komut İşlemleri Kodlama

Visual Basic'de nesnelere hangi durumlarda hangi işlemleri yapacağı kodlama ile belirtilir. Bunlar nesnenin üzerine farenin gelmesi, nesnenin tıklanması ya da nesnenin çift tıklanması gibi durumlar olabilir.

Bir komut düğmesini kodlamak için:

- Düğmeyi çift tıklayınız.
- Kod penceresine gerekli kodu giriniz. İlk olarak ekrana gelen prosedür tıklama için olacaktır. Değişik olaylar için sağdaki prosedür düşen listesini tıklayarak diğer durumları seçip değişik kodlar yazabilirsiniz.
- İlk satırdaki Private Sub xxxx_Click() komutu bu prosedürün xxxx komut düğmesi tıkladığında çalışacak prosedür olduğunu belirtir. En alt satırdaki End Sub komutu ise prosedürün bittiğini göstermektedir.

Bir diğer komut düğmesine kod yazmak için kod penceresini kapattıktan sonra, o düğmeyi çift tıklayarak, ya da açık olan kod penceresinin sağdaki nesne düşen listesini tıklayarak ve istenilen nesneyi seçerek de prosedürü yazabiliriz.

Diyalog Pencereleri

Yazı kutusu dışında kullanıcıdan bilgi almak için diyalog pencereleri kullanılabilir. Bunun için kullanılan fonksiyon InputBox (kullanıcının girdiği biçimde atanır) ya da InputBox\$ (karakter olarak atanır) dır.

```
InputBox$ (Açıklama$[,Başlık$][,İlkDeğer$][,x%,y%])
```

Sub Form_Load

'Form görüntülendiğinde adınızı soran diyalog penceresi ekrana gelir.

```
Ad = InputBox$("Lütfen adınızı giriniz:", "Ad", "Damla Buket")
```

End Sub

Prosedür Kodu

Bu kısımda program için gerekli hesaplamalar yazılacaktır. Programın içine yazılacak olan açıklamalar için satırın en soluna tek tırnak işareti koymak yeterlidir. Yazı kutusundan gelen değerlerin sayı değerine çevrilmesi için gerekli komut Val komutudur.

"txtSure" yazı kutusunun değerini bir değişkene atamak için aşağıdaki komut kullanılabilir:

```
' Sure değerini değişkene atar  
nSure = Val(txtSure.Text)
```

Programın Çalıştırılması

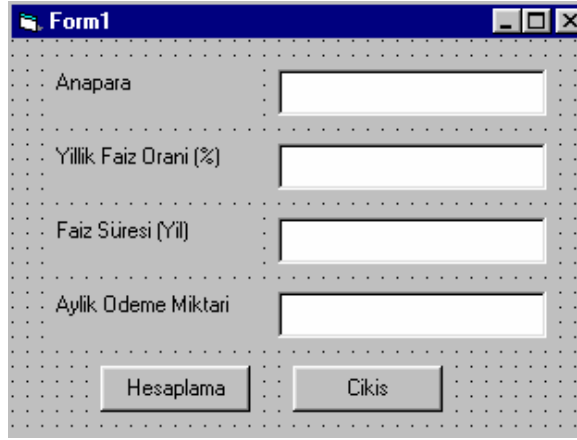
Bir program 3 şekilde çalıştırılabilir:

1. Visual Basic araç çubuğu üzerindeki "Start" düğmesine basınız.
2. Run/Start menü seçeneğini tıklayınız.
3. Klavyede F5 tuşuna basınız.

Hata olduğu takdirde bunlar bir pencerede görüntülenecektir.

KREDİ GERİ ÖDEME PLANI PROJESİ

Bu projenin amacı Şekil-8'deki gibi bir form oluşturmak, kullanıcıdan anapara, yıllık faiz oranı, ve faiz süresi (yıl bazında) bilgilerini alarak, aylık ödemeleri hesaplamaktır. Bu projede kullanıcıdan anapara bilgileri "Anapara", Yıllık faiz oranı bilgileri "Yıllık Faiz Oranı", ve yıl bazında faiz süresi bilgisi "Faiz Süresi (Yıl)" yazı kutuları ile alınacaktır. Hesaplama işlemi "Hesaplama" komut düğmesi ile başlatılacaktır. Sonuç yine "Aylık Odeme Miktari" yazı kutusu ile görüntülenecektir. Programdan çıkış için "Cikis" komut düğmesi kullanılacaktır.

The image shows a screenshot of a Windows application window titled "Form1". The window has a standard Windows XP-style title bar with minimize, maximize, and close buttons. The main area of the form is a grid with a dotted background. It contains four text input fields stacked vertically, each with a label to its left: "Anapara", "Yıllık Faiz Oranı (%)", "Faiz Süresi (Yıl)", and "Aylık Odeme Miktari". At the bottom of the form, there are two buttons: "Hesaplama" on the left and "Cikis" on the right.

Şekil-8. Kredi Geri Ödeme Planı Projesi

Projeyi Kaydetme

"Kredi Geri Ödeme Planı" projesinin saklanması için gerekli düğmeye basıldığında öncelikle form daha sonra da proje için diyalog kutuları ekrana gelecektir. Bu işlemde form için dosya adı olarak "krediana" (uzantısı .frm olacaktır) kullanınız ve dosyayı "My Documents" altında "krediproje" adlı bir klasör yaratarak, bu klasöre saklayınız. Daha sonra ekrana gelen proje diyalog kutusunda ise dosya adı için "krediproje" (uzantısı .vbp olacaktır) yazınız ve Save tuşuna basınız (dosya aynı klasöre saklanacaktır).

Yazı Kutusu Ekleme

Proje için gerekli olan 4 yazı kutusunu formun solunda etiketlere yer bırakacak şekilde yarattıktan sonra bu kutulara anapara için "txtAnapara", yıllık faiz oranı için "txtFaiz", faiz süresi için "txtSure", ve ödemeler için "txtOdeme" isimlerini "Name" seçeneği ile veriniz. "Text" seçeneğinde yer alan "text" yazısını siliniz. Bu alan kutunun bir yazı ile dolu olarak gelmesi istenirse kullanılır.

Etiket Ekleme

Proje için gerekli olan 4 etiket kutusunu formun sağına ve yazı kutularının soluna gelecek şekilde yarattıktan sonra bu kutulara anapara için "lblAnapara", yıllık faiz oranı için "lblFaiz", faiz süresi için "lblSure", ve ödemeler için "lblOdeme" isimlerini veriniz. Ekran görüntüsü için ise "Caption" özelliklerine anapara için "Anapara", yıllık faiz oranı için "Yıllık Faiz Oranı (%)", faiz süresi için "Faiz Süresi (Yıl)", ve ödemeler için "Aylık Ödeme Miktarı" değerlerini giriniz.

Komut Düğmesi Ekleme

Hesaplama ve programdan çıkış için gerekli 2 komut düğmesi yaratınız. Hesaplama düğmesinin "Name" özelliğine "cmdHesap", "Caption" özelliğine ise "Hesaplama" giriniz. Çıkış düğmesinin "Name" özelliğine "cmdCik", "Caption" özelliğine ise "Çıkış" giriniz.

Komut İşlemleri Kodlama

Projenin "Çıkış" düğmesini çift tıklayınız. Private Sub cmdCik_Click() komutu ile End Sub komutu arasına programdan çıkış için gerekli "end" komutunu yazınız.

Proje için gerekli değişken tanımlarını aşağıdaki gibi yapınız:

```
Dim cAnapara As Currency
Dim fFaiz As Single
Dim nSure As Integer
Dim cOdeme As Currency
```

Prosedür Kodu

Projede cAnapara değeri txtAnapara yazı kutusundan alınacaktır.

```
' Anapara değerini değişkene yükle
cAnapara = Val(txtAnapara.Text)
```

Diğer hesaplama kodları ise aşağıdaki gibidir:

```
'Yüzde faiz oranını ondalık değere çevir
fFaiz = Val(txtFaiz.Text) / 100
'Yıllık faiz oranını aylığa çevir
fFaiz = fFaiz / 12
'Toplam yıl süresini aya çevir
nSure = Val(txtSure.Text) * 12
```

'Aylık ödemeleri hesapla

$cOdeme = cAnapara * (fFaiz / (1 - (1+fFaiz)^{-nSure}))$

`txtOdeme.Text = Format(cOdeme,"Fixed")`

Programın Çalıştırılması

Start" düğmesini tıklayarak projeyi çalıştırınız. Anapara için 128000, faiz oranı için 9.75, süre için ise 30 değerlerini giriniz. Aylık ödemeler 1099.72 olmalıdır.

FORMAT:

Format (1234, "\$#,###.00")

Biçime 0 konduğunda rakam olmasa bile 0 mutlaka görüntülenir.

\$1,234.00

Format (1234, "\$#,###.##")

Biçime # konduğunda rakam olmazsa görüntülenmez.

\$1,234

Format (234, "\$#,###.00")

\$234.00

Format (1234567, "\$#,###.00")

\$1,234,567.00

Format(1234, "Currency")

\$1,234.00 veya 1.234,00 TL (Türkçe)

FormatNumber (1234)

1,234.00 veya 1.234,00 (Türkçe)

FormatDateTime("20:10", vbLongTime)

20:10:00

FORM ÖZELLİĞİ DEĞİŞTİRME PROJESİ (Program Çalışırken)

Bu örnekte formun genişliği ile ilgili bilgi kullanıcıdan alınıp, buna göre form özelliğini değiştiren program yazılacaktır. Form ilk açıldığında ekranın sol üst köşesinde ve boyutları değişmez olarak görüntülenecektir.

Form (Ornek1) üzerinde bir yazı kutusu (txtGenis-Name), bir etiket (Istenilen Genislik-Caption, lblGenis-Name) ve bir komut düğmesi (Formu Genislet-Caption, cmdGenis-Name) olacaktır.

Form Prosedürü

```
Sub Ornek1_Load
```

```
    Ornek1.Top = 0
```

```
    Ornek1.Left = 0
```

```
    Ornek1.BorderStyle = 0
```

```
    Ornek1.ControlBox = 0
```

```
End Sub
```

Komut Düğmesi Prosedürü

```
Sub cmdGenis_Click
```

```
    ' Formun genişliğinin 500'den fazla olmaması için denetleme
```

```
    If Val(txtGenis.Text) > 500 Then Ornek1.Width = 500
```

```
    Else Ornek1.Width = Val(txtGenis.Text)
```

```
    End If
```

```
End Sub
```

SEVİMLİ BİR UYGULAMA

1. Yeni bir proje yaratınız.
2. Form üzerine bir zamanlayıcı ve bir resim denetleyicisi koyunuz.
3. Resmin "Picture" özelliğinin yanındaki düğmeyi tıklayınız.
4. "Load Picture" diyalog pencereci ekrana geldiğinde VisualBasic\Graphics\Icons\Misc dizininden face03.ico resmini seçiniz.
5. Aşağıda verilen kodları gerekli yerlere yazınız.

```
Dim xChange As Integer  
Dim yChange As Integer
```

```
Private Sub Form_Load  
    xChange=100  
    yChange=100  
End Sub
```

```
Private Sub Timer1_Timer()  
    Image1.Left = Image1.Left + xChange  
    Image1.Top = Image1.Top + yChange  
    If Image1.Left > Me.ScaleWidth Then xChange=xChange * -1  
    If Image1.Left < 0 Then xChange=xChange * -1  
    If Image1.Top > Me.ScaleHeight Then yChange=yChange * -1  
    If Image1.Top < 0 yChange=yChange * -1  
End Sub
```

6. "Timer Control" özellikler penceresinde "Interval" özelliğini "True" yapınız.

Programı çalıştırdığımızda, resim ekran üzerinde hareket edecek ve formun bir kenarına çarptığında yön değiştirecektir.

ACCESS'DEN VERİ AKTARIMI ÖRNEĞİ

Veri tabanından bilgi aktarımı için kullanacağımız komutlar OpenDatabase, ve OpenRecordset'dir. Veri tabanı (Db) ve kayıt kümesi (Rs) genel tanımları yapıldıktan sonra, bu tanımlardan veritabanı için OpenDatabase tanımı, kayıt kümesi için de Open Recordset tanımı yapılmalıdır. OpenRecordset tanımında kullanılan "snapshot" dosya açıldığında aktif olan kaydın bir kopyasını alır. Bu örnekte tek kayıtlık bir veritabanı kullanılacak ve o bilgi görüntülenecektir.

Tüm kayıtların görüntülenmesi işlemi bir döngü ile aktif kayıt ilerletilerek yapılmalıdır.

Visual Basic Programı'na Access veritabanından veri aktarımı gerçekleştirmek için öncelikle bir veri tabanı hazırlamamız gereklidir.

Access Veri tabanı hazırlama işlemleri:

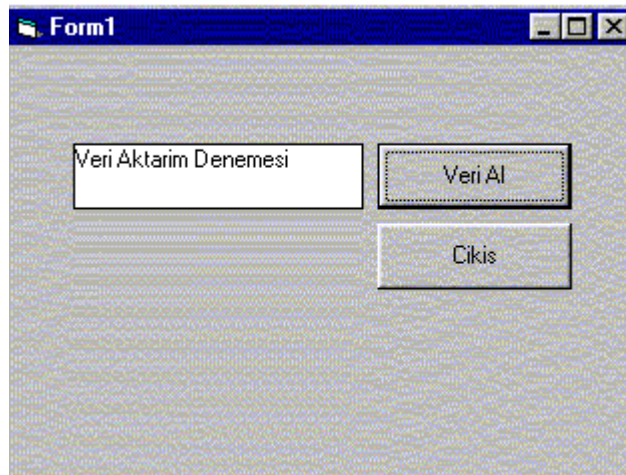
1. Access'de "c:\MyDocuments" dizini altına "ornek.mdb" dosyası açınız.
2. Tables/New/Design seçenekleri ile tablo tasarımı penceresini aktif hale getiriniz.
3. "Field Name" kolonuna "Yazi" yazınız.
4. "Data Type" kolonunda "Text" seçiniz.
5. Tabloyu kapatınız ve saklama penceresi geldiğinde tabloyu "tblornek" adı ile saklayınız.
6. Ana anahtar seçimi için "No" yanıtlayınız.
7. Tablonuzu "Open" ile açınız. İlk satıra "Veri Aktarım Denemesi" yazınız.
8. Access'den çıkınız.

Visual Basic'de ise aşağıdaki işlemleri gerçekleştiriniz:

1. Yeni bir proje açınız. Projeyi "ornek" olarak yine "c:\MyDocuments" dizinine saklayınız.
2. Form_Load Prosedürü'ne eklenecek kod:
Dim Db as Database
Dim Rs as Recordset
3. Formun üzerine bir etiket ve 2 komut düğmesi yerleştiriniz.
4. Etiket özellikleri:
Caption = (Boş olacak)

- Name = Label1 (Aynı kalacak)
BorderStyle = 1
Appearance = 0
5. Birinci komut düğmesi özellikleri:
Caption = Veri Al
Name = Command1 (Aynı kalabilir)
 6. İkinci komut düğmesi özellikleri:
Caption = Çıkis
Name = Command2 (Aynı kalabilir)
 7. Birinci komut düğmesi Click_Prosedürü
Set Db = OpenDatabase("c:\MyDocuments\ornek.mdb")
Set Rs = Db.OpenRecordset("tblornek",dbopensnapshot)
Label1.Caption = Rs!Yazi
 8. İkinci komut düğmesi Click_Prosedürü:
End
 9. Project/References menü seçeneğinden sonra liste üzerinde "Microsoft DAO 3.51 Object Library" seçeneğini işaretleyiniz.
 10. Programı saklayınız.

Programı çalıştırdıktan sonra "Veri Al" düğmesine bastığınızda Şekil-9'daki görüntü oluşacaktır.



Şekil-9 Veri Tabanından Veri Aktarım Örneği

VERİ TABANINDA ARAMA YAPMA ÖRNEĞİ

Bir kayıt bilgisinin veri tabanında olup olmadığını belirlemek için arama işlemi yapmanız gerekmektedir. Bu işlem için öncelikle aranacak bilginin bir yazı kutusu ile kullanıcıdan alınması daha sonra da veri tabanında arama yapılması gerekmektedir. Bilginin bulunup bulunmadığı bilgisi yine yazı kutusu ile kullanıcıya verilebilir.

Arama için yapacağınız işlemler aşağıdaki gibidir:

1. "ornek.mdb" veritabanına "deneme" ve "veri aktarım" olmak üzere iki adet kayıt daha giriniz.
2. "ornek.frm" üzerine bir yazı kutusu ekleyiniz, "Text" özelliği boş olsun.
3. "ornek.frm" üzerine bir komut düğmesi ekleyiniz, "Caption" özelliği ara olsun.

Bu komuta aşağıdaki prosedürü ekleyiniz.

```
Private Sub Command3_Click()  
Set Db = OpenDatabase("c:\bg\telif\vb\ornek.mdb")  
Set Rs = Db.OpenRecordset("tblornek", dbOpenSnapshot)  
Do While Not Rs.EOF  
    If Text1.Text = Rs!Yazi Then  
        Text1.Text = "Bulundu"  
    Else:  
        Rs.MoveNext  
    End If  
Loop  
If Text1.Text <> "Bulundu" Then Text1.Text = "Bulunamadi"  
End Sub
```

4. Programı çalıştırdıktan sonra yazı kutusuna "Deneme" girip "Ara" düğmesini tıklarsanız yazı kutusu üzerine "Bulundu" yazısı görüntülenecektir. Eğer "Ornek" girip yine "Ara" düğmesini tıklarsanız bu kez de "Bulunamadi" yazısı görüntülenecektir.

VERİ RAPORU YARATMA

Veri raporu yaratma işlemi için öncelikle veri alanı tanımlamaları yapılmalıdır.

Veri Bağlantısı Tanımlamaları

1. Project/Add Data Environment (Project/Components/Designers/Data Environment ya da Project/More Activex Designers/Data Environment) tıklanır.
2. Açılan "Data Environment" penceresindeki "Connection1" üzerinde farenin sağ tuşu tıklanarak "Properties" seçilir.
3. "Data Link Properties" diyalog kutusu üzerinde "Microsoft Jet 3.51 OLE DB Provider" tıklanır.
4. "Next" düğmesi tıklanarak "Connection" kısmına geçilir.
5. "Use data source name" yazı kutusu yanındaki (...) düğmesi tıklanarak Access veri tabanı seçildikten sonra kullanacağınız veri tabanı seçilir.
6. "Test connection" tıklanarak veri tabanına erişim yapıp yapılamadığı denetlenir.

Veri Tabanı Tanımlamaları

1. "Connection1" üzerinde farenin sağ tuşu tıklanarak "Add Command" seçilir.
2. Eklenen "Command1" üzerinde sağ tuş tıklanarak "Properties" seçilir. Açılan pencerenin "General" kısmında "Command Name" ile bir isim atanır, "Connection" ile istenilen bağlantı seçilir (birden fazla bağlantı olduğu durumlarda kullanılmalıdır), "Database Object" ile verinin tablo olduğu belirtilir, "Object Name" üzerinden tablo ismi seçilir. (Hastane veri tabanı üzerinde kimlik tablosu gibi)
3. Eğer bu veri tabanına ilişkili bir veri tabanı var ise bu "Add Child Command" ile gerçekleştirilir. Diğerleri gibi isim ve tablo ismi tanımlamaları yapıldıktan sonra "Relation" ile ilişki tanımı yapılır. (Hastane veri tabanı üzerinde vizite tablosu, kimlik-hastakod ile vizite-hastakod ilişkisi gibi)

Veri Raporu Yaratma

1. Project/Add Data Report seçilir. Ekranı "Access" benzeri bir rapor ortamı gelir.
2. "Properties" penceresi üzerinde "Data source" için önceden tanımlanan "Data Environment", "Data Member" için ise tanımlanan veri tabanı girilir. İstenirse raporun ismi değiştirilebilir.
3. "Data Report" penceresi üzerinde farenin sağ tuşu tıklanarak "Retrieve Structure" seçilir. Bu işlem otomatik olarak veri alanlarını tanımlar.
4. Daha sonra "Data Environment" penceresi üzerinden alanlar "Data Report" üzerine tıklanıp sürüklenerek eklenir ve istenilen biçimde yerleştirilir.

Yaratılan veri raporları bir komut düğmesine eklenecek **rapor_ismi.show** ile çağırılabilir. Gösterilen rapor üzerinde yazdırma düğmesi bulunduğu için ayrıca yazdırma komutu eklemeye gerek yoktur.

HASTANE VERİ TABANI UYGULAMASI

HASTANE VERİ TABANI

Veri tabanı üzerinde Data1 (hastane-kimlik) ve Data2 (hastane-vizite) nesneleri "visible.false" olarak bulunmaktadır.

The screenshot shows a Windows-style application window titled "Hastane". It features a grid of input fields and buttons. The "Hasta Kodu" field contains "1". The "Adı" field contains "Tayfun". The "Soyadı" field contains "Özturan". Below these are fields for "Protokolno:" (with values "1" and "3"), "Tarih:" (with values "01/01/99" and "02/02/99"), and "Tani:" (with value "Reçete Kontrol"). A "Soyada göre bul" button is present. At the bottom, there are navigation buttons "Geri", "Baş", "Son", and "İleri". A search field contains "soyad = Özturan" and a dropdown menu shows "1".

(General Declaration)
Public son1 As Integer
Dim son2 As Integer
Public sayi As Integer

Sub ilkform()

```
cmdileri.Visible = True  
cmdgeri.Visible = True  
cmdbas.Visible = True  
cmdson.Visible = True  
cmddekle.Visible = True  
cmdrapor.Visible = True  
cmdguncelle.Visible = True  
cmdbul.Visible = True  
cmdcikis.Visible = True  
cmdiptal.Visible = False
```

End Sub

Public Sub goster()

```
txtad.Text = Data1.Recordset("ad")  
txtsoyad.Text = Data1.Recordset("soyad")  
txthastakod.Text = Data1.Recordset("hastakod")  
Data2.Recordset.MoveFirst  
liste(0).Clear  
liste(1).Clear
```

```

liste(2).Clear
endx = 0
Do While endx = 0
  Do While Not Data2.Recordset.EOF
  If Data1.Recordset("hastakod") <> Data2.Recordset("hastakod") Then
    Data2.Recordset.MoveNext
  Else
    endx = 1
    varx = Data2.Recordset("hastakod")
    vary = Data2.Recordset("hastakod")
    Do While varx <> 0
    Do While vary = varx
    liste(0).AddItem (Data2.Recordset("protokolno"))
    liste(1).AddItem (Data2.Recordset("tarih"))
    liste(2).AddItem (Data2.Recordset("tani"))
    Data2.Recordset.MoveNext
    If Data2.Recordset.EOF = True Then
      varx = 0
      vary = 1
    Else
      vary = Data2.Recordset("hastakod")
    End If
    Loop
    varx = 0
    Loop
  End If
Loop
End Sub

```

Sub ekledugkaldir()

```

cmdileri.Visible = False
cmdgeri.Visible = False
cmdbas.Visible = False
cmdson.Visible = False
cmdekle.Visible = False
cmdrapor.Visible = False
cmdguncelle.Visible = False
cmdbul.Visible = False
cmdcikis.Visible = False
cmdiptal.Visible = True
End Sub

```

Private Sub bul_Click()

```

Data1.Recordset.MoveFirst
sayi = 1
endx = 0
Do While endx = 0
  If Data1.Recordset("hastakod") <> bul.Text Then

```

```

        Data1.Recordset.MoveNext
        sayi = sayi + 1
    Else
        endx = 1
        goster
    End If
Loop
End Sub

```

Private Sub cmdbas_Click()

```

Data1.Recordset.MoveFirst
goster
sayi = 1
End Sub

```

Private Sub cmdbul_Click()

```

soyadx = txtsoyad.Text
bulx = 0
bul.Clear
Data1.Recordset.MoveFirst
Do While Not Data1.Recordset.EOF
    If Data1.Recordset("soyad") = soyadx Then
        bul.AddItem (Data1.Recordset("hastakod"))
        Data1.Recordset.MoveNext
        bulx = 1
    Else
        Data1.Recordset.MoveNext
    End If
Loop
If bulx = 0 Then
MsgBox ("bulamadim.")
Else
    bul.Text = bul.List(0)
    bul.Visible = True
    txtbul.Visible = True
    txtbul.Text = "soyad =" + soyadx
End If
End Sub

```

Private Sub cmdcikis_Click()

```

End
End Sub

```

Private Sub cmddegtamam_Click()

```

Data1.Recordset.Edit
Data1.Recordset("Ad") = txtad.Text
Data1.Recordset("Soyad") = txtsoyad.Text
Data1.Recordset.Update
Data1.Refresh
cmddegtamam.Visible = False

```

```
Data1.Recordset.Move (sayi - 1)
ilkform
goster
End Sub
```

```
Public Sub cmdekle_Click()
Ekle.Show
Hastane.Hide
son1 = son1
End Sub
```

```
Private Sub cmdgeri_Click()
If sayi <> 1 Then
    Data1.Recordset.MovePrevious
    goster
    sayi = sayi - 1
Else
    MsgBox ("Kayit basina gelindi!")
End If
End Sub
```

```
Private Sub cmdguncelle_Click()
ekledugkaldır
cmddegtamam.Visible = True
End Sub
```

```
Private Sub cmdileri_Click()
If sayi <> son1 Then
    Data1.Recordset.MoveNext
    goster
    sayi = sayi + 1
Else
    MsgBox ("Kayit sonuna gelindi!")
End If
End Sub
```

```
Private Sub cmdiptal_Click()
goster
cmdekletamam.Visible = False
cmddegtamam.Visible = False
ilkform
goster
End Sub
```

```
Private Sub cmdrapor_Click()
DataReport1.Show
End Sub
```

```
Private Sub cmdson_Click()
Data1.Recordset.MoveLast
```

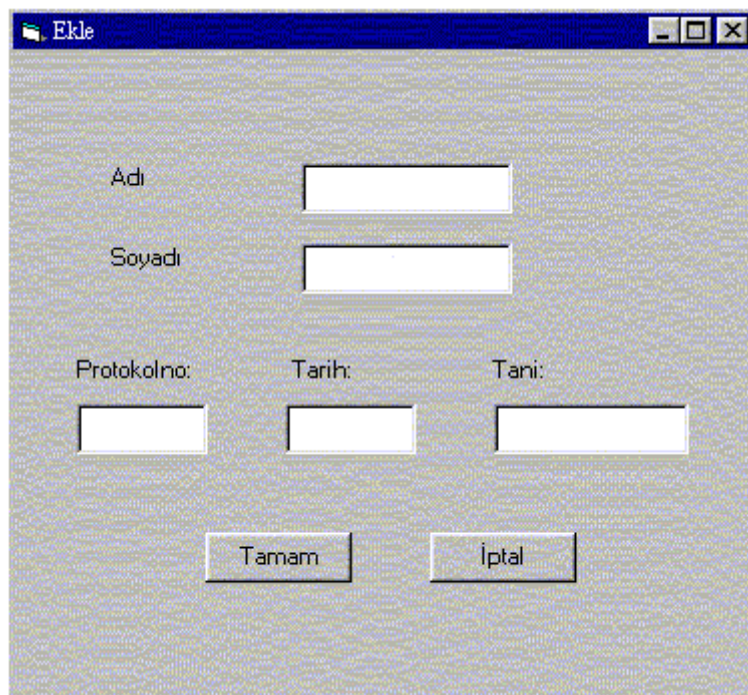


```
goster  
sayi = son1  
End Sub
```

Private Sub Form_Load()

```
Data1.Refresh  
Data2.Refresh  
Data1.Recordset.MoveLast  
son1 = Data1.Recordset.RecordCount  
Data2.Recordset.MoveLast  
son2 = Data1.Recordset.RecordCount  
Data1.Recordset.MoveFirst  
sayi = 1  
goster  
End Sub
```

EKLE FORMU



The image shows a screenshot of a Windows application window titled "Ekle". The window has a blue title bar with standard Windows window controls (minimize, maximize, close). The main area of the window is light gray and contains several text input fields and two buttons. The fields are arranged as follows: "Adi" followed by a text box; "Soyadı" followed by a text box; "Protokolno:" followed by a text box; "Tarih:" followed by a text box; and "Tani:" followed by a text box. Below these fields are two buttons: "Tamam" and "İptal".

Public Sub cmdekletamam_Click()

```
Hastane.Data1.Recordset.AddNew  
Hastane.Data1.Recordset("Ad") = txtad.Text  
Hastane.Data1.Recordset("Soyad") = txtsoyad.Text  
Hastane.son1 = Hastane.son1 + 1  
Hastane.Data1.Recordset("Hastakod") = Hastane.son1  
Hastane.Data2.Recordset.AddNew  
Hastane.Data2.Recordset("Hastakod") = Hastane.son1  
Hastane.Data2.Recordset("protokolno") = txtprot.Text  
Hastane.Data2.Recordset("tarih") = txttarih.Text  
Hastane.Data2.Recordset("tani") = txttani.Text  
Hastane.Data1.Recordset.Update  
Hastane.Data2.Recordset.Update
```

```
Hastane.Data1.Refresh  
Hastane.Data2.Refresh  
Hastane.goster  
Hastane.sayi = 1  
Hastane.Show  
Ekle.Hide  
End Sub
```

```
Private Sub cmdiptal_Click()  
Hastane.Show  
Ekle.Hide  
End Sub
```

```
Private Sub Form_Load()  
txtad.Text = ""  
txtsoyad.Text = ""  
txtprot.Text = ""  
txttani.Text = ""  
txttarih.Text = ""  
End Sub
```