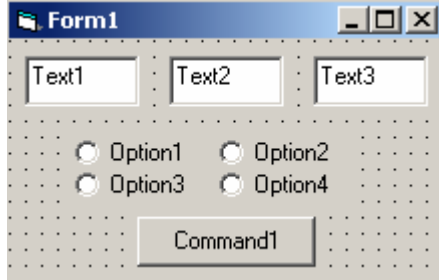
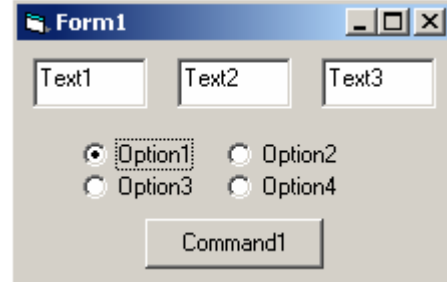


## OPTION BUTTON ve ELSEIF UYGULAMALARI

**OPTION** düğmeleri, Muhtelif seçeneklerin olduğu ve bu alternatif seçeneklerden birisinin seçilmesi gerektiği durumlarda kullanılan nesnelere denir. Daire şeklindeki bu option nesnesi, fare ile tıklanarak seçildiklerinde içinde siyah bir nokta belirir. Birden çok option nesnesi özel olarak gruptandırılmadan kullanılmışsa, nokta, hangi option nesnesi tıklanmışsa ona geçer, bir önceki nokta içeren option nesnesindeki nokta kaybolur.



Option nesnesi kullanarak tasarlanmış bir form örneği



Aynı form örneğinin çalıştırılmış (RUN edilmiş) örneği

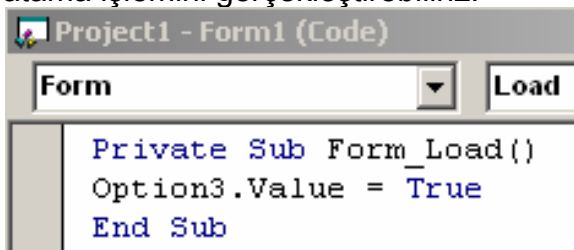
Yukarıdaki örneklerde **Text** kutuları, **Command** düğmesi ve dört adet **Option** nesnesi kullanarak tasarlanmış form örnekleri görünmektedir. Sağdaki örnek, soldaki tasarımın çalışır durumdaki halini göstermekte olup, bu örnekte Option1 nesnesi seçili durumdadır. Eğer, program algoritması içinde formun yüklenmesi sırasında herhangi bir alternatif seçenek varsayılan (default) değer olarak belirtilmemiş ise daima 1. Option nesnesi varsayılan olarak seçili konumda gelir.

**Option** nesnelerinin seçili olup olmadıklarının kontrolü veya seçili duruma program algoritması içinden atanmaları, **Value** (değer) özellikleri kontrol edilerek yapılabilir. **Value** özelliği **TRUE** ve **FALSE** olmak üzere iki seçeneğe sahiptir. Eğer Value özelliği True ise (veya True olarak atanmışsa) nesne seçili durumdadır.

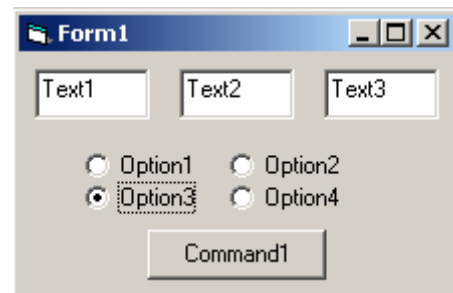
Örneğin Option1 nesnesinin seçili duruma getirilmesi algoritma içinden yapılacaksa, ilgili komut satırı;

**Option1.Value = True** şeklinde yazılır.

Yukarıdaki örnekte herhangi bir tanımlama yapılmadığı için Option1 nesnesi seçili konumda gelmekteydi. Biz Program ilk çalıştığında Option3 nesnesinin varsayılan seçim olarak gelmesini istiyorsak, bu durumda tasarım **Formunun Load** özelliğinde bu atama işlemini gerçekleştirebiliriz.



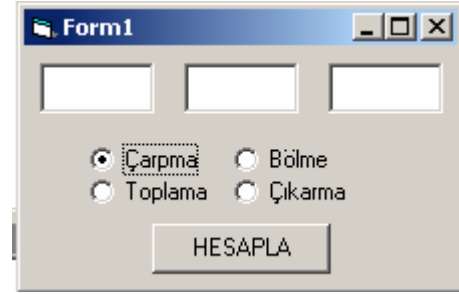
Option3 nesnesinin formun yüklenmesi sırasında varsayılan seçim olarak atanması



Option3 nesnesinin seçili nesne olarak atanmış olduğu Formun çalışır durumdaki görünümü

Yukarıdaki tasarımı basit bir 4 işlem hesap makinesi tasarımına dönüştürmek istediğimizi ve yapılacak aritmetik işlemini de Option nesnelere seçerek belirlemek istediğimizi varsayalım. Tasarımdaki nesnelere başlıklarını (Caption özellikleri) Formun yüklenmesi sırasında atayalım. Bu durumda ilgili Form yükleme algoritması ve form görünümü aşağıdaki şekilde olacaktır.;

```
Project1 - Form1 (Code)
Form
Load
Private Sub Form_Load()
Text1.Text = ""
Text2.Text = ""
Text3.Text = ""
Option1.Caption = "Çarpma"
Option2.Caption = "Bölme"
Option3.Caption = "Toplama"
Option4.Caption = "Çıkarma"
Command1.Caption = "HESAPLA"
End Sub
```



Formun yüklenmesi sırasında tasarım nesnelere atanan Caption özellikleri

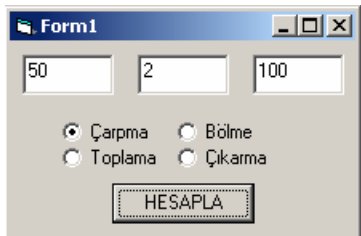
Form.Load özelliğini kullanımı

Şimdi bu tasarımda, Text1 ve Text2 kutularına girilen sayıların, Option nesnelere göre birisinin seçili olmasına göre "HESAPLA" komut düğmesinin tıklanmasıyla işlem sonucunun Text3 kutusuna aktarılmasını istediğimizi varsayalım. Bu durumda HESAPLA (Command1) düğmesi için yazılması gereken algoritma;

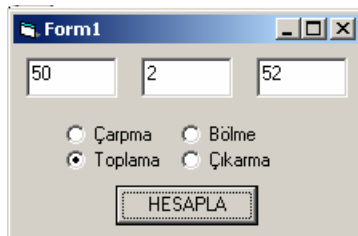
#### Private Sub Command1\_Click()

```
If Option1.Value = True Then Text3.Text = Val(Text1.Text) * Val(Text2.Text)
If Option2.Value = True Then Text3.Text = Val(Text1.Text) / Val(Text2.Text)
If Option3.Value = True Then Text3.Text = Val(Text1.Text) + Val(Text2.Text)
If Option4.Value = True Then Text3.Text = Val(Text1.Text) - Val(Text2.Text)
End Sub
```

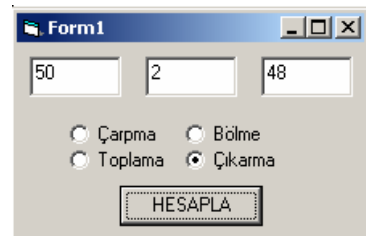
Şeklinde olacaktır. Bu algoritma çalıştırılarak farklı option nesnesi seçildikten sonra Komut düğmesi tıkladığında; aşağıdaki sonuçlar ve ekran görüntüleri elde edilir.



Option1 nesnesi seçiliyken HESAPLA düğmesi tıkladığında elde edilen sonuç



Option3 nesnesi seçiliyken HESAPLA düğmesi tıkladığında elde edilen sonuç



Option4 nesnesi seçiliyken HESAPLA düğmesi tıkladığında elde edilen sonuç

Bu örnekte, işlemlerin yapılabilmesi için, HESAPLA (Command1) düğmesi tıkladığında her bir Option nesnesi klasik **If ... Then ...** koşul cümlesi uygulaması ile test edilerek uygun koşulun sağlandığı seçenek gerçekleştirilmektedir. Bilindiği üzere

“If” koşul cümleleri **IF Bloğu** olarak tanımlanan ve koşullar ile alternatifleri tek satır yerine çok satırda tanımlandığı;

```
If koşul(lar) Then  
.....  
Yapılacak işler  
.....  
Else  
.....  
Alternatif işler  
.....  
End If
```

Şeklinde de kullanılabilir. Herhangi bir işin yapılabilmesi bir tek **If** satırı ile ifade edilebilecek koşul(lar) veya **Else** kullanılarak iki olasılıklı koşul(lar) söz konusu olduğunda daha önce tanımlanan **If ..Then..Else...** komut düzeni veya **standart If bloğu** kullanımı yeterli olabilir. Ancak, bir iş (veya işlerin) yapılabilmesi ikiden daha çok olasılığın kontrolüne bağlı ise, standart If bloğu kullanımında yer alan **Else** ifadesi **ElseIf** olarak ek koşul tanımlamada kullanılır.

Örneğin, yukarıda örneklenen basit dört işlem hesap makinesi programı tasarımında 4 adet Option nesnesinin her biri, klasik tek satır **If..Then..** cümlesi ile denetlenmişti. Bu örneği, yukarıda tanımlanan ve ikiden daha çok koşulun bir if bloğu içinde ElseIf kullanılarak ardışıklı olarak kontrol edildiği bir algoritmaya dönüştürürsek, HESAPLA (Command1) düğmesi için tanımlanan program kod satırları;

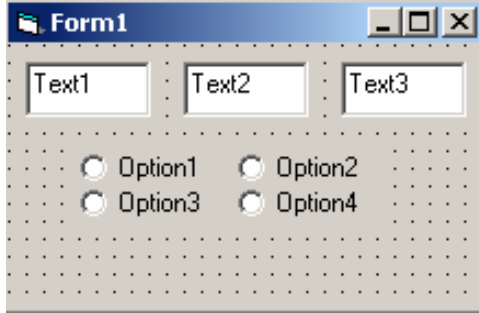
```
Private Sub Command1_Click()  
If Option1.Value = True Then  
    Text3.Text = Val(Text1.Text) * Val(Text2.Text)  
ElseIf Option2.Value = True Then  
    Text3.Text = Val(Text1.Text) / Val(Text2.Text)  
ElseIf Option3.Value = True Then  
    Text3.Text = Val(Text1.Text) + Val(Text2.Text)  
ElseIf Option4.Value = True Then  
    Text3.Text = Val(Text1.Text) - Val(Text2.Text)  
End If  
End Sub
```

şeklinde olacaktır. Bu kodun çalıştırılması ile elde edilecek sonuç, yukarıdaki şekillerde (ekran görüntülerinde) olduğu gibi elde edilecektir.

Buraya kadar anlatılan örnekte, işlem sonucunun elde edilebilmesi için, Text kutularına sayı girişinden sonra, sırası ile önce işlem tercihinin Option nesnesi ile belirlenmesi, daha sonra da HESAPLA komut düğmesinin tıklanması gerekmektedir. Tüm bu Option nesnesi Value kontrolleri, **Command1** düğmesinin **Click** özelliği altında tanımlanmış bulunmaktadır.

Şimdi, aynı örnek üzerinde değişiklik yaparak, söz konusu basit hesap makinesi programını, Komut (Command) düğmesi kullanmadan, doğrudan Option nesnesinin seçilmesi durumuna bağlı olarak çalışır ve sonuç verir şekle getirmek istediğimizi varsayalım. Bu durumda Command Düğmesi kullanmayacağımıza göre, Option nesnelere ait Value kontrollerinin ve değer atamalarının gerçekleştirilmesi için,

Command düğmesi yerine, yine **Option** düğmesinin **Click** özelliğinden yararlanmamız gerekecektir. Bu durumda ilgili formu, HESAPLA düğmesini kaldırarak yeniden tasarladığımızda ve Form nesnelерinin başlıklarını (Caption) **Formun Load** özelliği ile atadığımızda, form görünümü ve ilgili program kod satırları aşağıdaki gibi olacaktır.



Komut düğmesi kaldırılmış Form tasarımının görünümü

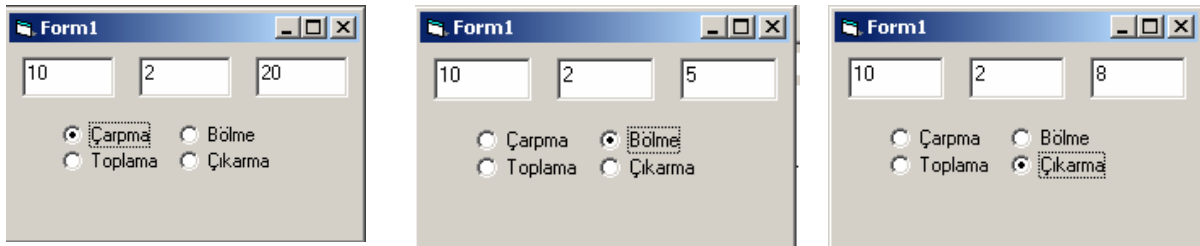
```
Project1 - Form1 (Code)
Form Load
Private Sub Form_Load()
Text1.Text = ""
Text2.Text = ""
Text3.Text = ""
Option1.Caption = "Çarpma"
Option2.Caption = "Bölme"
Option3.Caption = "Toplama"
Option4.Caption = "Çıkarma"
End Sub
```

Form.Load özelliği ile Caption ataması

Bu form tasarımı (program) üzerinde Option nesnelерinin Click özelliklerine bağlı işlem yaptırabilmek için yazılması gereken kod satırları ;

```
Private Sub Option1_Click()
Text3.Text = Val(Text1.Text) * Val(Text2.Text)
End Sub
Private Sub Option2_Click()
Text3.Text = Val(Text1.Text) / Val(Text2.Text)
End Sub
Private Sub Option3_Click()
Text3.Text = Val(Text1.Text) + Val(Text2.Text)
End Sub
Private Sub Option4_Click()
Text3.Text = Val(Text1.Text) - Val(Text2.Text)
End Sub
```

şeklinde olacaktır. Bu algoritma çalıştırıldığında ve uygun değerler verilerek işlem seçenekleri tıklandığında aşağıdaki ekran görüntüleri elde edilir.

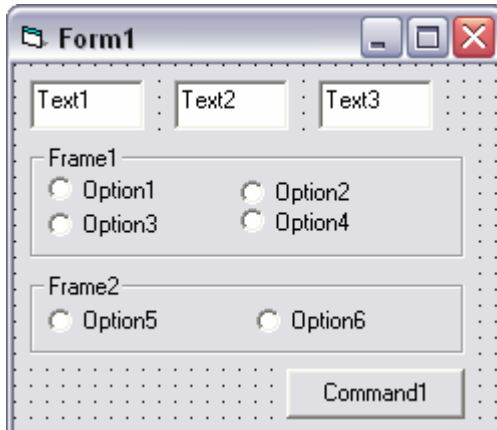


Çeşitli işlem seçeneklerinde program ekran görüntüleri

## Option Nesnelerinin Gruplandırılması : Frame Kullanımı

Yukarıda anlatıldığı şekilde bir form üzerinde yer alan Option nesnelerinin yalnızca bir tanesi seçimli olabilmekteydi. Birisi tıklandığında seçili konuma gelmekte, bir önceki seçili olanı seçim dışı kalmaktaydı. Bazen aynı anda birden çok option düğmesinin farklı amaçlar için seçimli olması arzu edilebilir. Bu nedenle, option nesnelerinin gruplandırılması gerekir. Bu en basit şekliyle option nesnelerinin, Frame nesnelere içine yerleştirilmeleri ile sağlanabilir. Kaç grup yapılmak istendiği belirlendikten sonra, form üzerine o kadar Frame nesnesi yerleştirilir. Her Frame nesnesi içine de, o frame ile kaç özellik kontrol edilecekse o kadar option nesnesi yerleştirilir. Her bir nesne için gerekli kodları yazılabilir. Her frame içinde yer alan option nesnelere, diğer frame içindeki nesnelere farklı davranır. Bir frame içinde bir option nesnesi seçili iken, diğer frame içinde de bir başka option nesnesi seçili olabilir.

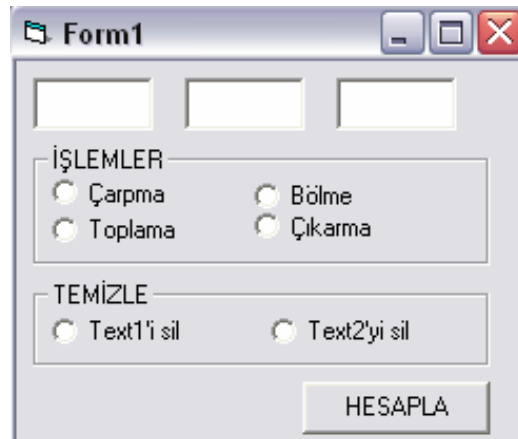
Yukarıda denenmiş olan basit hesap makinesi örneğimizi geliştirerek, mevcut 4 option nesnemizi gruplandırarak ayrıca yine farklı grup yapılmış 2 option nesnesi daha ekleyelim ve bu iki option nesnesinin **click** özelliklerini kullanarak Text1 ve Text2 kutularının içerisini silmek istediğimizi varsayalım. Bu silme işlemini yaparken, mevcut seçili aritmetik işlem seçeneği option nesnemizde değişiklik olmasın. Bu işlem için formumuza eklene yeni nesnelere ile formumuz;



Yandaki görünümü almıştır. Eklenen 2 adet **Frame** içlerine ayrı ayrı option nesnelere eklenmiştir.

Bu tasarım üzerindeki, tüm nesnelere başlıklarını (caption) form yükleniş sırasında değiştirmek istediğimizde, gerekli kod yazılımını ve programın çalışır durumdaki görünümü;

```
Private Sub Form_Load()  
Text1.Text = ""  
Text2.Text = ""  
Text3.Text = ""  
Frame1.Caption = "İŞLEMLER"  
Option1.Caption = "Çarpma"  
Option2.Caption = "Bölme"  
Option3.Caption = "Toplama"  
Option4.Caption = "Çıkarma"  
Command1.Caption = "HESAPLA"  
Frame2.Caption = "TEMİZLE"  
Option5.Caption = "Text1'i sil"  
Option6.Caption = "Text2'yi sil"  
End Sub
```



Yukarıdaki şekilde olacaktır. Çalışır durumda option nesneleri seçilir ve değerler girilirse ;

Frame1 içindeki nesnelere seçili, işlem yapılmış, ancak Frame2 seçimi henüz yapılmamış form

Frame1 içindeki nesnelere seçili, işlem yapılmış, Frame2 seçimi yapılarak Text2 silinmiş, ancak, Frame1 içindeki seçim korunmuş form

Frame2 içinde yer alan option nesnelere için kullanılan (gerekli) kodlar;

---

```
Private Sub Option5_Click()  
    Text1.Text = ""  
End Sub
```

---

```
Private Sub Option6_Click()  
    Text2.Text = ""  
End Sub
```

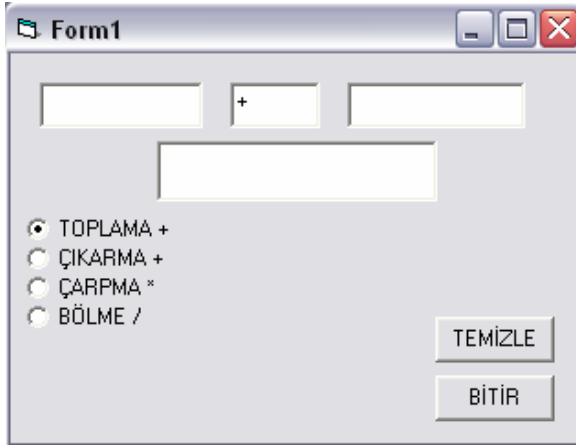
---

Şeklinde yazılmıştır. Bu örnekte, Frame2 içinde yer alan yeni option nesnelere için yazılan bu kodlar dışında, bir önceki örneğe göre, kullanılan diğer nesnelere kodlarında bir değişiklik olmayıp, yalnızca yeni eklenen nesnelere tanımlanan caption özellikleri için Formun Load özelliğine yukarıda da gösterildiği şekilde yeni satırlar eklenmiştir.

## ÖRNEK UYGULAMA

Seçili option nesnelere bağlı olarak 1. ve 3. metin kutularına bilgi (sayı) girildikçe, bu metin kutularının değişim özelliğine bağlı olarak 2. metin kutusunda yapılan işlemin işaretini, 4. metin kutusunda da işlem sonucunu gösteren bir form tasarlayarak program kodlarını yazınız. Ayrıca, kullanılacak bir komut düğmesi yardımıyla da 1. ve 3. metin kutularının içeriğini silinmesini bir diğer komut düğmesi yardımıyla da programın sonlandırılmasını sağlayınız.

### ÇÖZÜM :



Yukarıda tanımlanan problem için yandaki şekilde tasarlanmış bir form düzenlenebilir. Form üzerinde yer alan Option nesnelere ve komut düğmelerinin başlıkları (Caption) Programda **Formun Load** özelliğinde değiştirilmiştir. Program, seçili Option nesnesine bağlı olarak Text1 ve Text3 kutularına sayı girildikçe sonucu doğrudan Text4. kutusuna aktarmaktadır.

Söz konusu tasarım nesnelerinin caption özelliklerinin tanımlanması;

```
Private Sub Form_Load()  
Text1.Text = ""  
Text3.Text = ""  
Text2.Text = "+"  
Option1.Caption = "TOPLAMA +"  
Option2.Caption = "ÇIKARMA +"  
Option3.Caption = "ÇARPMA *"  
Option4.Caption = "BÖLME /"  
Command1.Caption = "TEMİZLE"  
Command2.Caption = "BİTİR"  
Option1.Value = True  
End Sub
```

kodları kullanılarak yapılmıştır. Bu kod ile, başlangıçta Text1 ve Text3 kutuları boşaltılmış, Text2 kutusuna varsayılan ilk seçenek olarak "+" tanımlanmış ve Option1 nesnesinin Value özelliği buna bağlı olarak True tanımlanmıştır. (Bunlar tercihe bağlı olup, programın ilk çalışmasında bir seçenek aktif olarak görünsün düşüncesi ile böyle yapılmıştır.) Ayrıca Option nesnelere ve komut düğmelerine başlıklar tanımlanmıştır.

Problemin tanımı gereği, hesaplama işlemleri, hem Text1 kutusunda hem de Text2 kutusunda bir değişiklik oldukça gerçekleştirilecektir. Bu nedenle, her iki metin kutusu için de aynı program kodlarının yazılması gerekecektir. Böylece, 1. veya 2. metin kutusunda bir değişiklik oldukça, seçili Option nesnesinin türüne bağlı olarak

gerçekleştirilecek işlemin sonucu 4. metin kutusuna, işlemin aritmetik işareti 2. metin kutusuna yazılacaktır.

Text1 ve Text2 kutuları için kod satırları;

```
Private Sub Text1_Change()  
If Option1.Value = True Then  
    Text2.Text = "+"  
    Text4.Text = Val(Text1.Text) + Val(Text3.Text)  
ElseIf Option2.Value = True Then  
    Text2.Text = "-"  
    Text4.Text = Val(Text1.Text) - Val(Text3.Text)  
ElseIf Option3.Value = True Then  
    Text2.Text = "*"  
    Text4.Text = Val(Text1.Text) * Val(Text3.Text)  
ElseIf Option4.Value = True Then  
    Text2.Text = "/"  
    If Val(Text3.Text) = 0 Then  
        Text4.Text = "Çözümsüz"  
        GoTo 100  
    End If  
    Text4.Text = Val(Text1.Text) / Val(Text3.Text)  
100 End If  
End Sub
```

---

```
Private Sub Text3_Change()  
If Option1.Value = True Then  
    Text2.Text = "+"  
    Text4.Text = Val(Text1.Text) + Val(Text3.Text)  
ElseIf Option2.Value = True Then  
    Text2.Text = "-"  
    Text4.Text = Val(Text1.Text) - Val(Text3.Text)  
ElseIf Option3.Value = True Then  
    Text2.Text = "*"  
    Text4.Text = Val(Text1.Text) * Val(Text3.Text)  
ElseIf Option4.Value = True Then  
    Text2.Text = "/"  
    If Val(Text3.Text) = 0 Then  
        Text4.Text = "Çözümsüz"  
        GoTo 100  
    End If  
    Text4.Text = Val(Text1.Text) / Val(Text3.Text)  
100 End If  
End Sub
```

Olarak yazılmıştır. Burada Option nesnelere If bloğu içinde ElseIf kullanılarak ardışıklık olarak denetlenmektedir. Her bir koşul için, koşulun uygunluğuna bağlı olarak işlem yaptırılmakta, ayrıca Text2 kutusuna işlem işareti atanmaktadır (yazdırılmaktadır). Burada dikkat edilecek bir özellik olarak, her iki metin kutusu için bölme işlemi seçildiğinde, bölen değer yer aldığı metin kutusunun (Text3) içeriğinin "0" olup olmadığının denetlenmesi, eğer "0" ise işlem yapılmadan "Çözümsüz" ibaresinin sonuç kutusuna yazdırılarak işlemin sonlandırılmasıdır. Bu kısım, 0'a bölme olasılığı ve buna bağlı programın hata üreterek kesikliğe uğramasını engellemek amacıyla konulmuş bir kontrol ifadesidir. "Çözümsüz" ifadesi yazıldıktan sonra program 100 numaralı satıra yönlendirilerek, arada kalan;



**Text4.Text = Val(Text1.Text) / Val(Text3.Text)**

İfadesi atlatılmakta ve kod sonlandırılmaktadır.

Burada kullanılan If, ElseIf ve End If ifadelerine dikkat edilecek olursa, 100 numaralı satır olarak gösterilen End If, yukarıda ibirinci If koşulu ile bunu takip eden Else If koşullarını sonlandırmaktadır. 0'a bölme işleminin kontrol edildiği kısımaki If ve End If ise, Ana If bloğu içinde ikincil If bloğu olarak yer almaktadır.

Eğer Text3 kutusuna girilen değer 0'dan farklı ise, söz konusu if koşulu sağlanmadığı için, bu ikincil if bloğu devre dışı kalarak program;

**Text4.Text = Val(Text1.Text) / Val(Text3.Text)**

Satırında belirtilen işi gerçekleştirmektedir.

If ve End If ifadelerinin birbirleri ile ilişkilerini çizgi ile birleştirecek olursak;

```
Private Sub Text1_Change()  
If Option1.Value = True Then  
    Text2.Text = "+"  
    Text4.Text = Val(Text1.Text) + Val(Text3.Text)  
ElseIf Option2.Value = True Then  
    Text2.Text = "-"  
    Text4.Text = Val(Text1.Text) - Val(Text3.Text)  
ElseIf Option3.Value = True Then  
    Text2.Text = "*"  
    Text4.Text = Val(Text1.Text) * Val(Text3.Text)  
ElseIf Option4.Value = True Then  
    Text2.Text = "/"  
    If Val(Text3.Text) = 0 Then 1  
        Text4.Text = "Çözümsüz"  
        GoTo 100  
    End If 2  
Text4.Text = Val(Text1.Text) / Val(Text3.Text) 3  
100 End If  
End Sub
```

Görünümünü elde ederiz. Burada 1 numara ile gösterilen kutu içinde yer alan koşul cümlesi, iki satır üstündeki;

**ElseIf Option4.Value = True Then**

cümlesinde belirtilen koşulun sağlanması durumunda, kontrol edilmesi gereken ikincil bir koşulu göstermektedir. Bu nedenle, takiben 1. kutu içinde gösterilen;

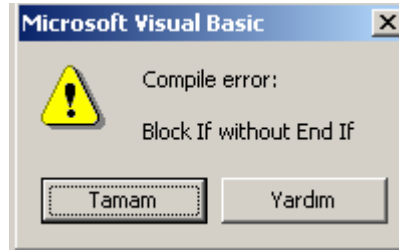
**If Val(Text3.Text) = 0 Then** ifadesi eklenmiştir.

Bu kontrolün yapılması sırasında. **If Val(Text3.Text) = 0** koşulu sağlandığında 2 numaralı kutu içinde verilen görevler yerine getirilecek ve program akışı, 3 numaralı kutudaki görev atlanarak **A** ile gösterilen kalın ok yönünde, 100 numaralı satıra aktarılarak sonlandırılacaktır.

Eğer **If Val(Text3.Text) = 0** koşulu sağlanmıyorsa, bu durumda da **B** ile gösterilen ok doğrultusunda, aradaki satırlar atlanarak, **2** numaralı kutuda ifade edilen görevler yerine getirilmeyerek, program akışı **3** numaralı kutuda ifade edilen görevlerin yapılmasıyla sonlandırılacaktır.

Yukarıda çizgi ve kutularla belirtilen kısımda **2** ve **3** numaralı kutular arasında görünen **End If** ifadesi, arada program akışının durumuna göre hiç uğranılmıyor ve kullanılmıyor gibi görünse de;

**If Val(Text3.Text) = 0 Then** olarak tanımlanan if bloğu cümlesini sonlandırmak amacıyla zorunludur. Eğer söz konusu **End IF** ifadesi, bu aradan silinecek olursa program akışı sırasında hata üreterek;



mesajını verecektir. Bu nedenle yazılmıştır

Program içinde komut düğmelerine verilmesi istenen t-emizleme ve program sonlandırma görevleri için de;

```
Private Sub Command1_Click()  
Text1.Text = ""  
Text3.Text = ""  
End Sub  
Private Sub Command2_Click()  
End  
End Sub
```

kodları yazılabilir.

Bu koda göre, 1 numaralı komut düğmesi tıkladıkça, Text1 ve Text2 temizlenecektir. 2 numaralı komut düğmesi tıkladığında da, program sonlandırılacaktır.