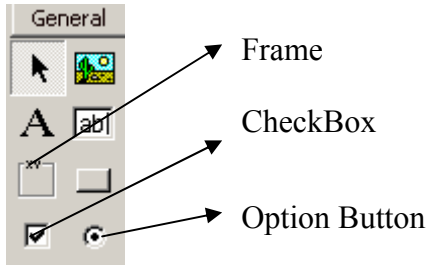


## Check Box kullanımı ve Metin kutusu FONT özellikleri

**Check Box** kutuları **Option** button nesnelere farklı olarak, programın çalışması sırasında farklı tercihlere bağlı olarak aynı anda birden çok seçeneğin seçilebilir olması özelliği ile farklılık gösterir.

Bilindiği üzere, **Option** nesnelere eğer Form üzerinde birden fazlaysa, bu nesnelere ancak yalnızca bir tanesi seçili konuma getirilebilmekteydi. Bu seçme işleminde de seçili Option nesnesi içindeki noktanın tıklanarak yeni seçilen Option nesnesine kayması ile belirlenmekteydi. Eğer aynı anda Option nesnelere iki veya üç (veya daha fazla) grup alternatif görevde, farklı seçimler için kullanılması gerekiyorsa söz konusu Option nesnelere gruplandırılması gerekmekte olup, bu işlemde, **Frame** nesnesi kullanılarak yapılabilmekteydi. Önce oluşturulan Frame nesnesi içine, bu grup için gerektiği kadar Option nesnesi yerleştirilmekte, Daha sonra diğer grup görev için açılan yeni bir Frame nesnesi içinde de o görevle ilgili Option nesnelere yerleştirilmekteydi.



**Check Box** kutuları, Form üzerinde tıklanıldığında içinde ✓ işareti beliren, tekrar tıklanıldığında da bu işaretin kaybolduğu seçim nesnelere dir. Söz konusu işaretin, CheckBox içinde görünür olması, bu seçeneğin seçili durumda olduğunu, işaretin kaldırılmış olması da seçimin iptal edilmiş olduğunu gösterir.

**Check Box** kutularının seçim durumları ilgili kod içinde **Value** özelliklerinin sorgulanması ile test edilirler. Eğer, **Check Box** kutusu seçili ise ve içinde ✓ işareti görünür durumda ise, bu CheckBox nesnesinin **Value** özelliği **1**, eğer seçili değilse (içinde ✓ işareti görünür durumda değilse) **Value** özelliği **0**, değerlerine sahiptir. Bu değerlere bağlı olarak, program kod satırları içinde yaptırılacak sorgulama ile (Value özelliğinin 1 veya 0 olup olmadığı) ilgili seçenek için çeşitli görevler kodlanabilir.

CheckBox nesnesinin kullanımını, bir örnek üzerinde uygulayalım; Bir öğrencinin, arasınavda almış olduğu notun metin kutusuna girilmesiyle, **60** başarı notunu sağlayabilmesi için, genel sınav notunun **kaç olması** gerektiğini (genel sınavda, sınav notu olarak en az **50** almış olmak koşuluyla) hesaplayan basit bir algoritma yazalım. Genel Sınav notunun algoritma içinde kullanılacak değişkeni **GS** olsun. Bu arada, ayrıca Form nesnelere adlarını da ilgili nesne için **properties** pencerelerini kullanarak değiştirelim. Bu kapsamda, Söz konusu hesaplamayı yapabilmek amacıyla;



görünümünde bir form tasarlayalım.

Bu form üzerinde yer alan;

**Text1** metin kutusu **sorgu** adıyla,  
**Text2** metin kutusu **sonuc** adıyla ve  
**Command1** düğmesi de **gshesap** adıyla tanımlanmış olsun

Bu nesnelere kullanarak **sorgu metin kutusu** nesnesine girilen not;

$$GS = (60 - (Val(x\$) * 0.4)) / 0.6$$

formülü kullanılarak alınması gereken başarı notu hesaplanabilir. Bu durumda, yazılması gerekli kod ve veri girilerek sonuç alınmış form görünümü aşağıdaki şekilde olacaktır;

Arasınav notu olarak 50 girilmiş ve sonuç alınmış form görünümü

```
Private Sub gshesap_Click()  
x$ = Val(sorgu.Text)  
GS = (60 - (Val(x$) * 0.4)) / 0.6  
sonuc.Text = GS  
End Sub
```

ilgili kod yazılımı

Ancak bu örneğe ait temel koşullardan bir tanesi genel sınav notunun en az 50 olması şartı ile sonuç not 2. metin kutusuna yazdırılacaktır. Bu nedenle yukarıdaki kod satırlarına hesap sonucu bulunan notun 50'den az olması durumunda **en az 50 alması gerektiği** açıklamasını, 50 ve üzeri notlar için ise yalnızca sonuç notu yazdıracak koşul cümleleri ekleyelim. Bu durumda ilgili kod;

```
Private Sub gshesap_Click()  
x$ = Val(sorgu.Text)  
GS = (60 - (Val(x$) * 0.4)) / 0.6  
If GS < 50 Then  
sonuc.Text = "En az 50 gerekli"  
Else  
sonuc.Text = GS  
End If  
End Sub
```

olacaktır.

Bu örnek uygulamanın esas amacı **CheckBox** kullanımını olduğuna göre, bu uygulamada 2. metin kutusuna yazdırılan değerlerin, **CheckBox** seçeneklerine göre normal yazı, **kalin yazı**, *italik yazı*, ve **hem kalın hem italik yazı** olarak değiştirilebilmesini sağlayalım. Bu amaç için mevcut Formumuz üzerine iki adet **CheckBok** yerleştirelim ve **Caption** özelliklerini değiştirelim.

CheckBox eklenmiş Form

CheckBox nesnelerinin Caption özellikleri değiştirilmiş Form

Yukarıda CheckBox Value özelliklerinin, seçili olup olmama durumuna göre, 0 ve 1 olarak değiştiğini ve buna göre sorgulanabileceklerine değinmiştik. Bu durumda, yukarıda tanımlanmış olan kod satırlarına ilgili koşul ifadeleri de eklendiğinde;

```

Project1 - Form1 (Code)
gshesap Click
Private Sub gshesap_Click()
    x$ = Val(sorgu.Text)
    GS = (60 - (Val(x$) * 0.4)) / 0.6
    If GS < 50 Then
        sonuc.Text = "En az 50 gerekli"
    Else
        sonuc.Text = GS
    End If
    If Check1.Value = 1 Then
        sonuc.Font.Bold = True
    Else
        sonuc.Font.Bold = False
    End If
    If Check2.Value = 1 Then
        sonuc.Font.Italic = True
    Else
        sonuc.Font.Italic = False
    End If
End Sub

```

1. metin kutusuna girilen arasınav notundan itibaren 60 başarı notu esasına göre Gerekli Genel Sınav notunun hesaplanması

Hesaplanan notun, 50'den az olup olmamasına göre, 2. metin kutusuna anlamlı sonucun yazdırılması

1. CheckBox seçeneğinin seçili olup olmamasına göre, 2. metin kutusunun içeriğinin **kalin Yazı** özelliğinin değiştirilmesi

2. CheckBox seçeneğinin seçili olup olmamasına göre, 2. metin kutusunun içeriğinin *italikYazı* özelliğinin değiştirilmesi

Kod dizilimi elde edilir.

Bu programın çalıştırılması ile, farklı seçenek durumlarında, aşağıdaki görünüm elde edilir.

Hiçbir CheckBox seçili değilken 2. metin kutusunun görünümü

Kalin Yazı için CheckBox seçiliyken 2. metin kutusunun görünümü

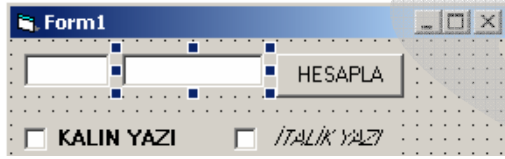
İtalikYazı için CheckBox seçiliyken 2. metin kutusunun görünümü

Hem Kalın hem de İtalik Yazı için CheckBox seçiliyken 2. metin kutusunun görünümü

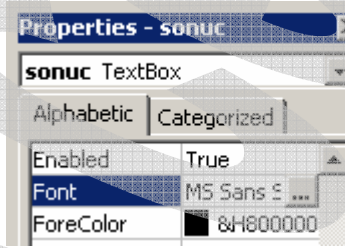
Yukarıda ifade edilen uygulamada, CheckBox seçeneklerine göre yazı görünümündeki değişiklikleri düzenlerken bu arada **FONT (yazı)** özelliklerinin değiştirilmesine de değinmiş olduk.

**FONT özellikleri** yazının yazı tipi, kalın, italik, normal gibi görünümü yanı sıra, yazı ve zemin rengi gibi özellikleri kapsar. Bu özellikler yukarıdaki örnekte ifade edildiği gibi program kodları vasıtasıyla değiştirilebileceği gibi, eğer seçimsiz olursa ve önceden varsayılan değerler olarak baştan düzenlenecekse yine ilgili nesnenin **properties** penceresinden düzenlenebilir.

Yukarıdaki örnekte kullandığımız 2. metin kutusunun (sonuc olarak adlandırılan metin kutusu) söz konusu özelliklerini properties penceresinden değiştirmek istediğimizi varsayalım. Bunun için 2. metin kutusu seçiliyken properties penceresinde değiştirilmek istenen özellik seçilir.

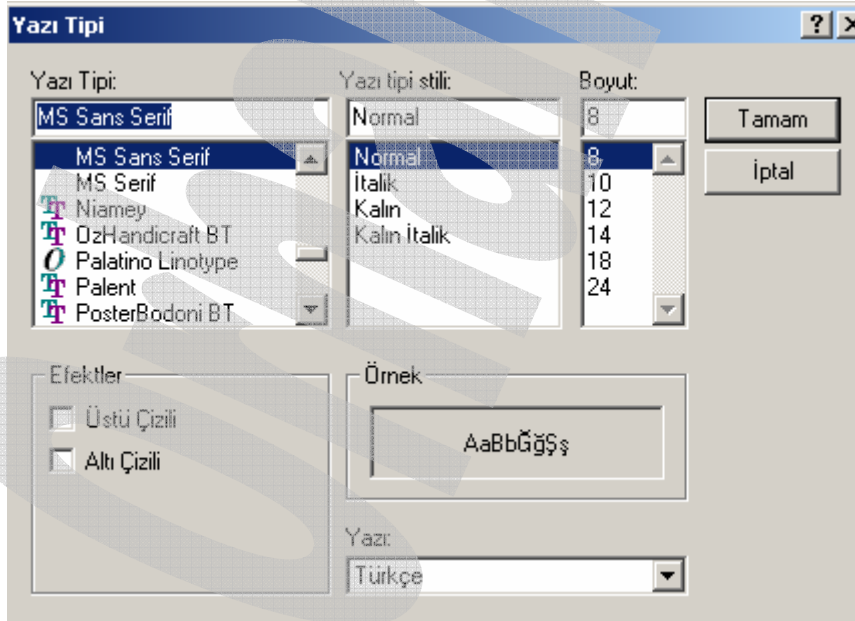


sonuc olarak adlanmış 2. metin kutusunun seçilmesi



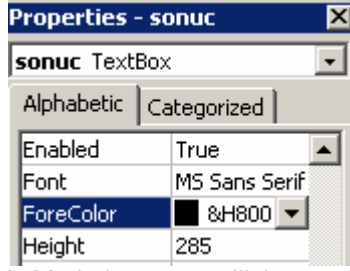
2. metin kutusu seçiliyken **FONT** seçeneğinin seçilmesi. (Bu seçenek seçildiğinde yazı tipi adının yanında, üzerinde üç nokta olan bir düğme belirir. Tıklandığında

aşağıda görünen **Yazı Tipi** diyalog penceresi açılır

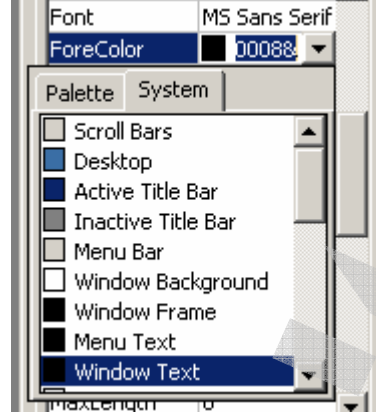


Bu diyalog penceresi vasıtasıyla ilgili nesnenin FONT özelliklerinin büyük bir kısmı değiştirilebilir.

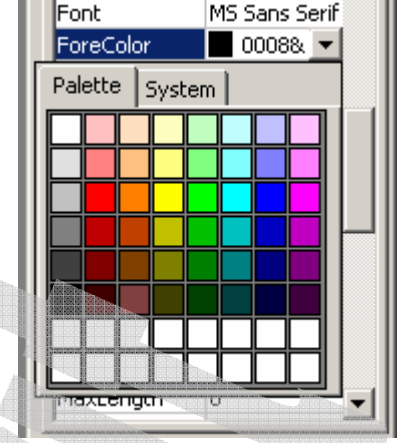
Yazı rengi değiştirilmek istendiğinde de yine properties penceresinde **ForeColor** seçeneği seçilir. Bu senek tıklandığında da yine **Font** seçeneğinde olduğu gibi yanında bir düğme belirir. Bu düğme tıklandığında **System** ve **Palette** adıyla gelen iki ayrı seçenek görünür. Bu **Tabl**ardan gerekli olan seçilerek açılan diyaloglardan istenen renk ayarlaması yapılabilir.



2. Metin kutusu seçiliyken yazı rengi (**ForeColor**) seçilmesi

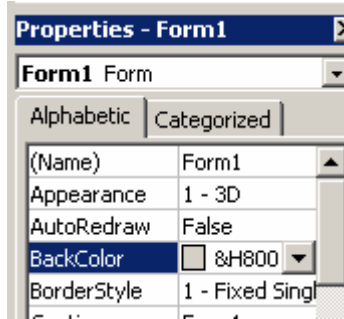


ForeColor özelliğinin System renk seçenekleri



ForeColor özelliğinin Palette renk seçenekleri

Yazı rengi için yapılan ayarlamalar **BackColor** seçeneği kullanılarak ta yapılabilir.



2. Metin kutusu seçiliyken zemin rengi (**BackColor**) seçilmesi

Zemin rengi seçimi işlemleri de yukarıda yazı rengi seçimi ile ilgili olarak anlatılan işlemlerin aynısıdır. Bu seçeneğinde alt menü seçenekleri yazı rengi seçimi alt menü seçeneklerine benzer.

Yukarıda tanımlanan bu özellikler benzer şekilde kod satırları vasıtası ile seçimli olarak da değiştirilebilir. Yukarıdaki öğrenci Genel Sınav notu hesaplama örneğinde de olduğu gibi, **Text1** kutusundaki metin özelliklerini değiştirmek istediğimizi varsayalım. Yaygın kullanılan metin özelliklerini aşağıdaki gibi değiştirebiliriz.

**Text1.Font.Bold = True**

Metin kutusu içeriğini kalın yazı yapar

**Text1.Font.Bold = False**

Metin kutusu içeriğinin kalın yazı özelliğini iptal eder

**Text1.Font.italic = True**

Metin kutusu içeriğini italik yazı yapar

**Text1.Font.italic = False**

Metin kutusu içeriğinin italik yazı özelliğini iptal eder

**Text1.ForeColor = vbRed**

Metin kutusu içeriğini kırmızı renkte yazar

**Text1.ForeColor = vbblack**

Metin kutusu içeriğini siyah renkte yazar

**Text1.BackColor = vbYellow**

Metin kutusu zeminini sarı renge dönüştürür

**Text1.BackColor = vbWhite**

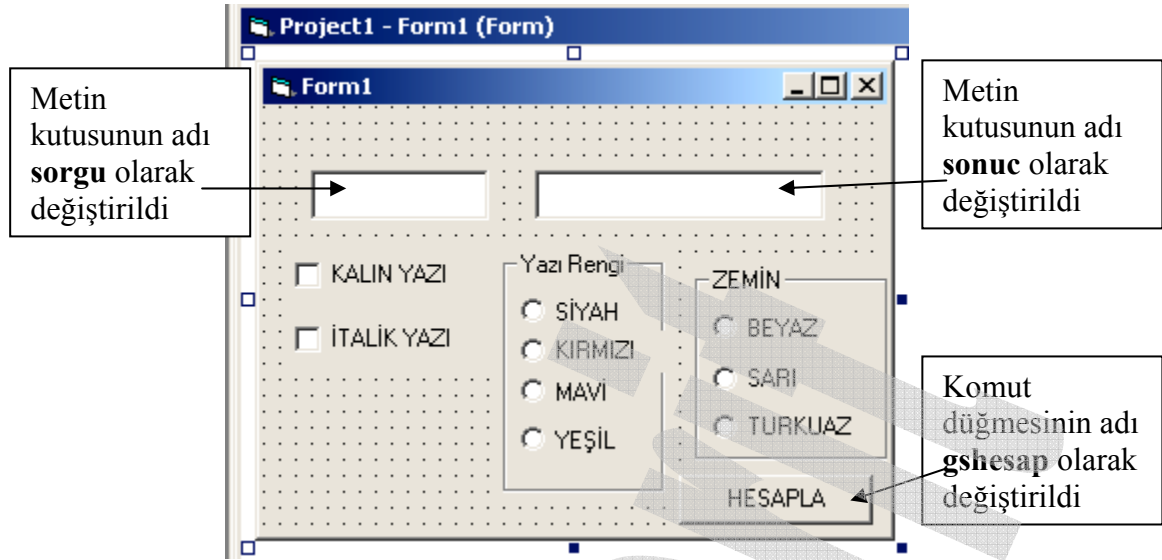
Metin kutusu zeminini beyaz renge dönüştürür

**Text1.FontName = "Arial"**

Metin kutusu içeriğini Arial yazı tipine dönüştürür

**Açıklama :** Renk özelliklerini kullanırken temel renklerin adları önüne **vb** ekleyerek tanımlamak mümkündür. Yazı tipi seçerken de yazı tipinin İngilizce doğru adı tırnak işaretleri arasında yazılmalıdır.

Yukarıda, CheckBox kullanımının anlatımı sırasında örneklendirdiğimiz ve kodlarını yazdığımız, arasınav notuna göre gerekli genel sınav notu hesaplama programı formu üzerinde değişiklik yaparak, 2. metin kutusuna yazdırdığımız sonuc değerini yazı ve zemin renklerini, gruplandırılmış Option nesnelere kullanarak değiştirelim.



Yukarıda tanımlanan örneğe uygun yazı rengi ve Zemin rengi seçenekleri eklenmiş Form tasarımı

```

If Option1.Value = True Then sonuc.ForeColor = vbBlack
If Option2.Value = True Then sonuc.ForeColor = vbRed
If Option3.Value = True Then sonuc.ForeColor = vbBlue
If Option4.Value = True Then sonuc.ForeColor = vbGreen
If Option5.Value = True Then sonuc.BackColor = vbWhite
If Option6.Value = True Then sonuc.BackColor = vbYellow
If Option7.Value = True Then sonuc.BackColor = vbCyan

```

Form örneğine eklenmiş olan Yazı ve Zemin rengi seçenekleri için düzenlenen (eklenen) program kod satırları

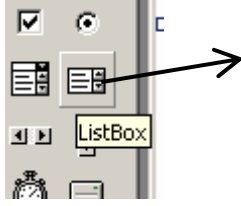
Sonuç olarak yukarıda görülen form tasarımı için gerekli nihai program kodu:

```

Private Sub gshesap_Click()
x$ = Val(sorgu.Text)
GS = (60 - (Val(x$) * 0.4)) / 0.6
If GS < 50 Then
    sonuc.Text = "En az 50 gerekli"
Else
    sonuc.Text = GS
End If
If Check1.Value = 1 Then
    sonuc.Font.Bold = True
Else
    sonuc.Font.Bold = False
End If
If Check2.Value = 1 Then
    sonuc.Font.Italic = True
Else
    sonuc.Font.Italic = False
End If
If Option1.Value = True Then sonuc.ForeColor = vbBlack
If Option2.Value = True Then sonuc.ForeColor = vbRed
If Option3.Value = True Then sonuc.ForeColor = vbBlue
If Option4.Value = True Then sonuc.ForeColor = vbGreen
If Option5.Value = True Then sonuc.BackColor = vbWhite
If Option6.Value = True Then sonuc.BackColor = vbYellow
If Option7.Value = True Then sonuc.BackColor = vbCyan
End Sub

```

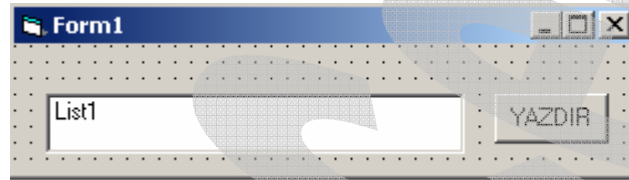
## LIST BOX Kullanımı



Visual Basic araç çubuğunda ListBox nesnesi seçeneği

**ListBox** Visual Basic uygulamalarında düzenli basit listeler, tablolar oluşturmak amacıyla kullanılan form nesnelerinden birisidir. Özellikle Text Kutusu gibi sınırlı kutular içine yazılmayacak kadar uzun ve ardışıklı olarak yazılacak çoklu satırlar için kullanılır.

**ListBox** içine yazdırılacak ifadeler **AddItem** komutu ile yazdırılırlar. Örneğin bir adet ListBox ve 1 adet Command button kullanılarak;



şeklinde tasarlanmış bir form içine "Orhan Cerit" ve "Cumhuriyet Üniversitesi" ifadelerini yazdırmak istediğimizi varsayalım. Bunun için gerekli kod;

```
Project1 - Form1 (Code)
Command1 Click
Private Sub Command1_Click()
List1.AddItem "Orhan Cerit"
List1.AddItem "Cumhuriyet Üniversitesi"
End Sub
```

şeklinde olacaktır. Bu kodun çalıştırılması ile ListBox içeriği aşağıda görüldüğü gibi olur;



Örneğin, bir önceki uygulamada, öğrencinin ara sınav notuna karşılık, başarılı olması için gerekli olan genel sınav notu hesabı uygulamasında, hesaplanmış olan Genel Sınav notlarının, her yeni veri girişinde kaybolmasını önlemek ve sonucun bir liste kutusunda sıralı olarak yazdırılmak istendiğini varsayalım. Bu amaçla form tasarımına bir de ListBox ekleyelim. Bu durumda program algoritmasının son satırında **End Sub** ifadesinden önce;

**List1.AddItem GS** yazılması yeterlidir

ListBox eklenmiş Form Tasarımı

ListBox eklenmiş Formun **List1.AddItem GS** kodu eklenmiş ve çalıştırılmış görünümü

Bu örnekte, girilen arasınav notuna karşılık hesaplanan genel sınav notları ListBox içine sırayla listelenmektedir. Ancak, Girilen arasınav notları gösterilmemektedir. Eğer Arasınav notlarının da görünmesi arzu ediliyorsa, (veya birden çok nesne tek ifade olarak yazdırılmak isteniyorsa) bu ifadeler birleştirilerek tek bir karakter dizisi (String) haline dönüştürülerek yine AddItem komutu ile ListBox içine atanabilirler.

Hatırlanacağı gibi, Karakter diziler "+" işareti kullanılarak toplanabilmektedirler. Örneğin;

**X\$ = "Orhan"**

**Y\$ = "Cerit"** ifadeleri

**Z\$ = X\$ + Y\$** şeklinde toplanırlarsa; bunun anlamı;

**Z\$ = "OrhanCerit"** ile eşdeğerdir. Eğer sayısal değerler de söz konusu ise ve

sayısal değerlerin Karakter dizileri ile bir cümle oluşturması isteniyorsa; Val fonksiyonunun tersi diyebileceğimiz **Str\$** fonksiyonunu kullanarak, sayısal değerleri, sayısal anlamı olmayan karakter dizilerine dönüştürerek, karakter dizilerine yamamak (eklemek) mümkündür. Örneğin Yukarıdaki **Z\$** ifadesi ile **123456** sayısını karakter dizisine dönüştürerek tüm bu ifadeyi **A\$** dizisinde göstermek istediğimizi düşünelim;

**A\$ = Z\$ + Str\$(123456)** ifadesi bu işlemi gerçekleştirecektir.



Bu durumda, yukarıdaki öğrenci not hesaplama örneğimizi, öğrencinin arasınava notunu da ifadeye alarak ListBox kutusu içinde listelemek istediğimizde, ListBox içine atmadan önce, Arasınava, genel sınav notlarını ve açıklama ifadelerini tek bir tümceye toplamamız ve bundan sonra atamamız gerekir. Bu durumda, yine program algoritmasında, **End Sub** ifadesinden önce;

**A\$ = "Arasınava notu " + X\$ + " için Genel SINAV " + Str\$(GS) + " olmalıdır  
List1.AddItem A\$**

Satırları programa eklenmelidir. Program çalıştırıldığında, veriler girildikçe sonuçlar birer cümle halinde, ListBox içine satır satır eklenir. Listelenen satırlar, ListBox görünüm alanına sığmaz ise otomatik olarak düşey kaydırma çubuğu ayrıca belirir.



Listbox eklenmiş Formun yeni kod eklenmiş ve çalıştırılmış görünümü

Sonuç olarak yukarıda görülen form tasarımı için gerekli nihai program kodu:

**Private Sub gshesap\_Click()**

X\$ = Val(sorgu.Text)

GS = (60 - (Val(X\$) \* 0.4)) / 0.6

If GS < 50 Then

sonuc.Text = "50 alsin isi bitirsin"

Else

sonuc.Text = GS

End If

If Check1.Value = 1 Then

sonuc.Font.Bold = True

Else

sonuc.Font.Bold = False

End If

If Check2.Value = 1 Then

sonuc.Font.Italic = True

Else

sonuc.Font.Italic = False

End If

If Option1.Value = True Then sonuc.ForeColor = vbBlack

If Option2.Value = True Then sonuc.ForeColor = vbRed

If Option3.Value = True Then sonuc.ForeColor = vbBlue

If Option4.Value = True Then sonuc.ForeColor = vbGreen

If Option5.Value = True Then sonuc.BackColor = vbWhite

If Option6.Value = True Then sonuc.BackColor = vbYellow

If Option7.Value = True Then sonuc.BackColor = vbCyan

A\$ = "Arasınava notu " + X\$ + " için Genel SINAV " + Str\$(GS) + " olmalıdır"

List1.AddItem A\$

**End Sub**

ListBox Kullanımını Ayrıca Form Tasarım örneği aşağıda verilen ve değerleri Metin kutularına girilen, Belirli bir yarıçap değerinden, yine belirli bir yarıçap değerine kadar, isteğe bağlı adım değerlerinde değişmek üzere, dairenin alan ve çevre değerlerini hesaplayarak bu yarıçap değerinin karşısına, ListBok içine listeleyen bir program yazalım.

Çok sayıda daire alan ve çevre hesabı form tasarımı örneği

Burada 1. Metin kutusuna başlangıç yarıçap değerini 10, 2. metin kutusuna nihai (son) yarıçap değerini 100 ve 3. metin kutusuna da adım değerini (kaçar kaçar hesaplanacak) girelim;

Çok sayıda daire alan ve çevre hesabı yapılmış Form

Bu işlemin yapılması için Command1 düğmesine atanan kod:

```
Private Sub Command1_Click()  
For R = Val(Text1.Text) To Val(Text2.Text) Step Val(Text3.Text)  
a = R * R * 3.14  
c = 2 * R * 3.14  
X$ = "YarıÇap:" + Str$(R) + " Alan:" + Str$(a) + " Çevre:" + Str$(c)  
List1.AddItem X$  
Next R  
End Sub
```

Aynı işlemler döngü değişkeni için değerler başlangıçta atanarak da yapılabilir;

```
Private Sub Command1_Click()  
bs = Val(Text1.Text)  
bt = Val(Text2.Text)  
adim = Val(Text3.Text)
```

```
For R = bs To bt Step adim
a = R * R * 3.14
c = 2 * R * 3.14
X$ = "YarıÇap:" + Str$(R) + " Alan:" + Str$(a) + " Çevre:" + Str$(c)
List1.AddItem X$
Next R
End Sub
```

Her iki kodun da sonuç işlevi aynıdır.

Orhan Cerit