

توانایی انجام یک پروژه عملی

هدف‌های رفتاری

پس از مطالعه این واحد کار از فراگیر انتظار می‌رود که:

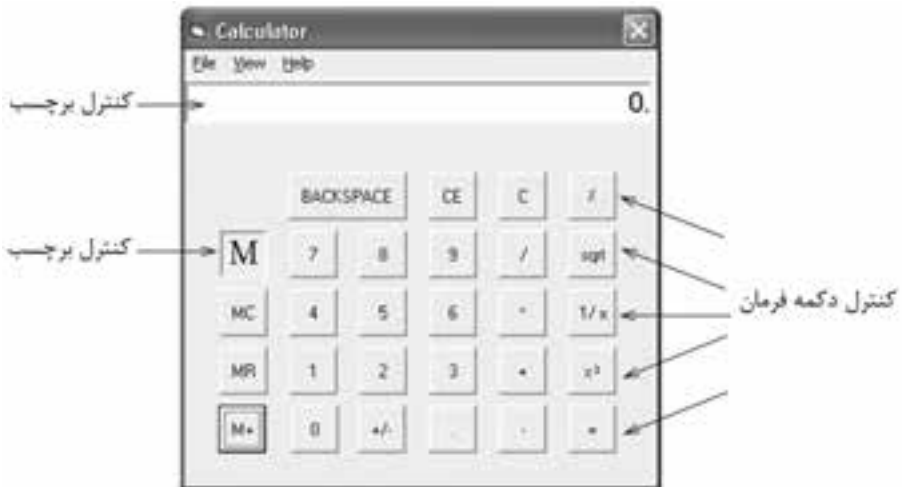
۱ - توانایی انجام یک پروژه برنامه‌نویسی را داشته باشد.

در این واحد کار یک پروژه برنامه‌نویسی واقعی را به صورت عملی انجام خواهید داد تا آنچه را که قبلاً فرا گرفته‌اید به کار بگیرید. یک ماشین حساب مطابق شکل ۱-۱۳ ایجاد کنید که توانایی انجام محاسبات چهار عمل اصلی، محاسبه مجذور و جذر، قرینه اعداد، درصد و معکوس اعداد را مانند یک ماشین حساب الکترونیکی داشته باشد. برای این کار عملیات زیر را به ترتیب انجام دهید:



شکل ۱-۱۳

۱- برنامه ویژوال بیسیک را اجرا کرده، یک پروژه از نوع Standard EXE ایجاد کنید که شامل یک فرم، کنترل دکمه‌های فرمان، دو برچسب مطابق شکل ۲-۱۳ و یک کنترل زمان‌سنج باشد، سپس خصوصیات آن‌ها را مطابق جداول ۱-۱۳ و ۲-۱۳ تنظیم کنید.



شکل ۲-۱۳

دکمه‌های فرمان دارای عرض و ارتفاع ۵۰۰ بوده و عنوان، نام و خصوصیات Top و Left آن‌ها مطابق شکل ۲-۱۳ است و خصوصیت Cancel تمام آن‌ها را مساوی False و خصوصیت Default آن‌ها (بجز کنترل =) را نیز روی False تنظیم کنید.

جدول ۱-۱۳ خصوصیات فرم

مقدار	خصوصیت
frmmain	Name
1-Fixed Single	BorderStyle
Calculator	Caption

جدول ۲-۱۳ خصوصیات کنترل‌ها

کنترل خصوصیت	Label	Label	Timer
Name	lblvalue	lblmemory	tmrdelay
Alignment	1	2	-
BorderStyle	1	1	-
Enabled	True	True	False
Interval	-	-	150
AutoSize	False	False	-
Caption	0	بدون عنوان	-

به بخش تعاریف فرم بروید و متغیرهای مورد نیاز برنامه‌ها را به صورت زیر تعریف کنید:

Dim dblno1 As Double, dblno2 As Double

Dim dblresult As Double, dblmemory As Double

Dim strop As String, strno As String

می‌بینید که متغیرهای مختلفی در بخش تعاریف، تعریف می‌شوند. متغیرهای dblno1 و dblno2 از نوع Double و به ترتیب برای ذخیره‌سازی عددی که قبل از انتخاب عملگر و عددی که بعد از انتخاب عملگر وارد می‌شوند، به کار می‌روند و متغیر dblresult از نوع Double و برای نگهداری نتیجه محاسبات روی متغیرهای dblno1 و dblno2 استفاده می‌شوند.

به علاوه متغیر dblmemory به عنوان حافظه ماشین حساب و متغیر رشته‌ای strop برای ذخیره کردن عمل ریاضی که کاربر انتخاب کرده، استفاده می‌شوند و به عنوان آخرین متغیر، یک متغیر رشته‌ای دیگر به نام strno عدد وارد شده به ماشین حساب را به صورت یک رشته در خود قرار می‌دهد تا در جریان اجرای برنامه با تبدیل آن به مقدار عددی در متغیرهای dblno_۱ و dblno_۲ ذخیره شود.

برای جلوگیری از تکرار دستورات در رویه‌های دیگر لازم است دو رویه را قبل از سایر رویه‌ها ایجاد نمایید. بنابراین اولین رویه را با نام dellblvalue و به صورت زیر تعریف کنید:

```
Private Sub dellblvalue()
```

```
    If (strop = "*" Or strop = "/" Or strop = "+" Or strop = "-") And strno = "" Then
```

```
        lblvalue.Caption = ""
```

```
    End If
```

```
End Sub
```

این رویه بررسی می‌کند که آیا قبل از ورود یک رقم، یکی از دکمه‌های چهار عمل اصلی فشرده شده است یا خیر. چون در صورت انتخاب یک عملگر، ورود هر یک از ارقام به عنوان عدد دوم محسوب می‌شود، در نتیجه باید عدد وارد شده که قبل از انتخاب عملگر مورد نظر در کنترل برچسب نمایش داده می‌شود، پاک شود. این عمل با دستور lblvalue.Caption = "" انجام خواهد شد. پس از تعریف رویه dellblvalue که در رویدادهای Click ارقام ۱ الی ۹ و نقطه اعشار اجرا می‌شود، رویه دوم را با نام inputnumber و به صورت زیر تعریف کنید. این رویه وظیفه دریافت ارقام و تشکیل عدد مورد نظر کاربر را به عهده دارد.

```
Private Sub inputnumber (strdigit As String)
```

```
    If strdigit <> "." Then
```

```
        If Left(strno, 1) = "0" And Mid(strno, 2, 1) <> "." Then
```

```
            strno = Mid(strno, 2) + strdigit
```

```
        Else
```

```
            strno = strno + strdigit
```

```
        End If
```

```
Else
```

```
    If InStr(1, strno, ".") = 0 Then
```

```
        If Len(lblvalue.Caption) = 0 Then strno = "0." Else _
```

strno=strno + "."

End If

End If

lblvalue.Caption = strno

End Sub

این رویه یک آرگومان به نام `strdigit` از نوع رشته‌ای دارد که در زمان فراخوانی با توجه به این که روی کدام دکمه کلیک شده است، رقم متناظر با آن را به رویه می‌دهد. پس از کلیک روی یکی از ارقام یک تا ۹ یا نقطه اعشار، رویه فراخوانی می‌شود و کاراکتر متناظر با رقم مربوطه در آرگومان `strdigit` ذخیره می‌شود. سپس با استفاده از یک دستور `If` مقدار این آرگومان بررسی می‌شود و در صورتی که مقدار آن مخالف کاراکتر نقطه باشد به این معنی است که دکمه مربوطه یکی از ارقام یک تا ۹ می‌باشد، در نتیجه `If` موجود در بخش `Then` اجرا می‌شود و رشته `strno` را که مجموعه ارقام وارد شده می‌باشد، مورد بررسی قرار می‌دهد.

با استفاده از توابع `Left` و `Mid` رشته `strno` بررسی می‌شود. اگر اولین و دومین کاراکتر رشته `strno` برابر با صفر و نقطه باشند ("0.") به این معنی است که عدد بین صفر و یک است، بنابراین رقم `strdigit` به رشته "0" اضافه می‌شود. اما در صورت نادرست بودن شرط رقم `strdigit` به وسیله دستور `strno=strno + strdigit` به مجموعه ارقام قبلی در رشته `strno` اضافه می‌شود. در واقع این `If` برای کنترل ارقام صفری است که قبل از نقطه اعشار و سایر ارقام غیرصفر وارد می‌شوند. به این صورت از دریافت رقم صفر قبل از سایر ارقام غیرصفر جلوگیری به عمل می‌آید.

اما اگر دکمه کلیک شده، نقطه اعشار باشد `If` موجود در بخش `Else` اولین `If` اجرا می‌شود. این `If` از ورود نقطه اعشار اضافه جلوگیری می‌کند. با استفاده از تابع `InStr` وجود نقطه اعشار در ارقامی که تاکنون وارد شده‌اند یعنی متغیر `strno` بررسی می‌شود. اگر رشته `strno` فاقد نقطه اعشار باشد `If` موجود در آن اجرا می‌شود و با استفاده از تابع `Len` طول رشته بررسی می‌شود و اگر این مقدار برابر صفر باشد به این معنی است عدد با مقدار 0 شروع می‌شود، در غیر این صورت نقطه اعشار با دستور `strno=strno+"0"` به ارقام قبلی اضافه خواهد شد. در پایان این رویه، ارقامی که تاکنون وارد شده‌اند با استفاده از دستور `lblvalue.Caption=strno` در کنترل برچسب نمایش داده می‌شوند. به این ترتیب می‌توان انواع اعداد کوچک‌تر یا بزرگ‌تر از یک را وارد کرد.

در این مرحله امکان ورود اعداد را فراهم کرده، رویداد Click دکمه‌های رقمی ۱ تا ۹ و نقطه اعشار را به صورت زیر تنظیم کنید:

```
Private Sub cmd1_Click()
```

```
    Call dellblvalue
```

```
    Call inputnumber ("1")
```

```
End Sub
```

```
Private Sub cmd2_Click()
```

```
    Call dellblvalue
```

```
    Call inputnumber ("2")
```

```
End Sub
```

```
Private Sub cmd3_Click()
```

```
    Call dellblvalue
```

```
    Call inputnumber ("3")
```

```
End Sub
```

```
Private Sub cmd4_Click()
```

```
    Call dellblvalue
```

```
    Call inputnumber ("4")
```

```
End Sub
```

```
Private Sub cmd5_Click()
```

```
    Call dellblvalue
```

```
    Call inputnumber ("5")
```

```
End Sub
```

```
Private Sub cmd6_Click()
```

```
    Call dellblvalue
```

```
    Call inputnumber ("6")
```

```
End Sub
```

```
Private Sub cmd7_Click()
```

```
    Call dellblvalue
```

```
    Call inputnumber ("7")
```

```
End Sub
```

```
Private Sub cmd8_Click()  
    Call dellblvalue  
    Call inputnumber ("8")  
End Sub
```

```
Private Sub cmd9_Click()  
    Call dellblvalue  
    Call inputnumber ("9")  
End Sub
```

```
Private Sub cmddot_Click()  
    Call dellblvalue  
    Call inputnumber (".")  
End Sub
```

۵ - در این مرحله رویداد دکمه فرمان رقم صفر را به صورت زیر تنظیم کنید:

```
Private Sub cmd0_Click()  
If strop="*" Or strop = "+" Or strop="-" Or Stop="/" Then  
    If Left(strno, 1)<>"0" Or Mid(strno, 2, 1) = "." Then  
        strno = strno + "0"  
    Else  
        strno = ""  
        lblvalue.Caption = ""  
    End If  
Else  
    If Left(strno,1)<> "0" Or Mid(strno, 2,1) = "." Then strno=strno+"0"  
End If  
lblvalue.Caption = strno  
End Sub
```

این رویداد از یک دستور If تشکیل شده است که در بخش Then و Else آن نیز مجدداً یک دستور If به کار رفته است. در اولین دستور If فشرده شدن یکی از دکمه‌های مربوط به چهار عمل اصلی بررسی می‌شود. در صورتی که هیچ یک از آن‌ها استفاده نشده باشند نتیجه شرط نادرست بوده و If موجود در بخش Else اجرا می‌شود. در این If نیز با استفاده از تابع Left و Mid بررسی می‌شود که آیا اولین کاراکتر در متغیر strno صفر یا دومین کاراکتر در آن نقطه اعشار است یا خیر؟

۶ - اکنون رویداد Click مربوط به چهار عمل اصلی را به صورت زیر تنظیم کنید:

```
Private Sub cmdmultiply_Click()
```

```
    If strop = "" Then
```

```
        strop = "*"
```

```
        dblno1 = Val(strno)
```

```
        strno = ""
```

```
    Else
```

```
        If strno <> "" Then
```

```
            dblno2 = Val(strno)
```

```
            Call Compute
```

```
            dblno1 = Val(strno)
```

```
            strno = ""
```

```
        End If
```

```
        strop = "*"
```

```
    End If
```

```
End Sub
```

```
Private Sub cmddivision_Click()
```

```
    If strop = "" Then
```

```
        strop = "/"
```

```
        dblno1 = Val(strno)
```

```
        strno = ""
```

```
    Else
```

```
        If strno <> "" Then
```

```
            dblno2 = Val(strno)
```

```
            Call Compute
```

```
            dblno1 = Val(strno)
```

```
            strno = ""
```

```
        End If
```

```
        strop = "/"
```

```
    End If
```

```
End Sub
```

```
Private Sub cmdplus_Click()
```

```
    If strop = "" Then
```



```
strop = "+"  
dblno1 = Val(strno)  
strno = ""  
Else  
If strno <> "" Then  
    dblno2 = Val(strno)  
    Call compute  
    dblno1 =Val(strno)  
    strno = ""  
End If  
strop = "+"  
End If  
End Sub
```

```
Private Sub cmdminus_Click()  
If strop = "" Then  
    strop = "-"  
    dblno1 = Val(strno)  
    strno = ""  
Else  
    If strno <> "" Then  
        dblno2 = Val(strno)  
        Call compute  
        dblno1 =Val(strno)  
        strno = ""  
    End If  
    strop = "-"  
End If  
End Sub
```

دستورات این چهار رویداد مشابه یکدیگر هستند. در این رویدادها ابتدا با استفاده از دستور If مقدار متغیر strop بررسی می‌شود تا مشخص شود تاکنون عملگری انتخاب شده است یا خیر (چون ممکن است عبارتی که به ماشین حساب داده می‌شود مشابه $2*3+4$ یا مانند عبارت $3=2*$ باشد). در صورتی که شرط درست باشد (یعنی عملگری تاکنون انتخاب نشده است) ابتدا کاراکتر متناسب با دکمه‌ای که کلیک شده است در متغیر strop

ذخیره می‌شود، مثلاً اگر روی دکمه * کلیک کنید کاراکتر "*" در strop ذخیره می‌شود، سپس عددی که قبل از کلیک روی چهار عمل اصلی وارد شده است و در متغیر strno نگهداری می‌شود با استفاده از تابع Val به نوع عددی تبدیل و در متغیر dblno1 به عنوان عدد اول ذخیره می‌شود و مجدداً مقدار "" در strno ذخیره می‌شود تا مقدار متغیر رشته‌ای strno برای دریافت عدد دوم از بین برود.

اما اگر قبلاً یکی از چهار عمل اصلی انتخاب شده باشند دیگر مقدار strop برابر رشته "" نیست و دستورات بخش Else اجرا می‌شوند. در این جا یک If دیگر استفاده می‌شود و اگر مقداری در متغیر strno موجود باشد به این معنی است که عدد دوم نیز وارد شده است. بنابراین در صورتی که عباراتی مانند $3+4*2$ یا مشابه آن‌ها وارد شوند به این معنی است که قبل از دریافت عدد ۴ باید محاسبه $3*2$ انجام شود، پس عدد دوم در متغیر dblno2 ذخیره خواهد شد و با فراخوانی رویه compute محاسبات انجام می‌شود. رویه compute را نیز به صورت زیر در ماژول فرم ایجاد کنید.

Private Sub compute()

Select Case strop

Case "*": dblresult = dblno1 * dblno2

Case "/": dblresult = dblno1 / dblno2

Case "+": dblresult = dblno1 + dblno2

Case "-": dblresult = dblno1 - dblno2

Case "power": dblresult = dblno1 ^ dblno2

End Select

lblvalue.Caption = dblresult

strno = dblresult

End Sub

با فراخوانی این رویه و با استفاده از یک دستور Select Case، مقدار متغیر strop بررسی می‌شود تا با توجه به عمل ریاضی انتخاب شده اعمال ضرب، تقسیم، جمع یا تفریق روی اعداد dblno1 و dblno2 انجام شده و نتیجه در متغیر dblresult ذخیره شود. سپس مقدار این متغیر در خصوصیت Caption برچسب ذخیره می‌شود تا نتیجه محاسبات در پنجره ماشین حساب دیده شود و به همین صورت این مقدار برای انجام محاسبات بعدی در متغیر strno ذخیره می‌شود.

به این ترتیب پس از خاتمه اجرای رویه compute نتیجه محاسبات که در متغیر strno

ذخیره شده در متغیر `dblno1` نیز قرار داده می‌شود تا در محاسبات بعدی به عنوان اولین عدد مورد استفاده قرار گیرد و پس از اجرای `strno=""` و حذف نتیجه محاسبات از متغیر `strno` دستورات `If` موجود در `Else` تمام می‌شود و به عنوان آخرین دستور در این بخش، کاراکتر متناسب با عمل ریاضی انتخاب شده در متغیر `strop` برای استفاده در مراحل بعدی ذخیره می‌شود. به این ترتیب یک دوره کامل از ورود اعداد و انجام محاسبات، کدنویسی و آماده شده است.

۷- اکنون باید ماشین حساب را به‌گونه‌ای تنظیم کنید که در صورت استفاده از عباراتی مانند $12.5+6=$ نیز محاسبات به درستی انجام شود، بنابراین رویداد `Click` دکمه فرمان تساوی را به صورت زیر تنظیم کنید:

```
Private Sub cmdequal_Click()  
    If strop="/" Or strop="+" Or strop="-" Or strop="*" Then  
        dblno2 = Val(strno)  
        Call compute  
    End If  
    strop = ""  
End Sub
```

در این رویداد نیز ابتدا مقدار متغیر `strop` بررسی می‌شود تا در صورت کلیک روی یکی از دکمه‌های چهار عمل اصلی محاسبات انجام شود، بنابراین در صورت درست بودن شرط موجود در دستور `If` ابتدا عدد دوم در متغیر `dblno2` ذخیره می‌شود، سپس رویه `compute` فراخوانی می‌شود تا محاسبات مربوطه انجام شود و محتویات متغیر `strop` برای مراحل بعدی خالی شود. ذکر این نکته ضروری است که عدد اول نیز در این حالت در هنگام کلیک روی یکی از چهار عمل اصلی در متغیر `dblno1` ذخیره می‌شود.

۸- در این مرحله رویداد `Click` دکمه `BACKSPACE` را به صورت زیر تنظیم کنید:

```
Private Sub cmdbackspace_Click()  
    If Len (lblvalue.Caption) = 1 Then  
        lblvalue.Caption = "0"  
        strno=""  
    Else  
        lblvalue.Caption = Left(lblvalue.Caption, Len (lblvalue.Caption)-1)  
        strno = lblvalue.Caption  
    End If  
End Sub
```

می بینید که این رویه نیز با یک دستور If آغاز می شود و با بررسی طول رشته ای که در کنترل برچسب نمایش داده می شود عمل حذف کاراکترها را از مقدار موجود، انجام می دهد. بنابراین اگر طول این رشته برابر یک باشد شرط درست بوده و مقدار خصوصیت Caption کنترل برچسب روی صفر تنظیم شده سپس دستور "strno =" برای اعمال تغییرات در محاسبات نیز اجرا می شود. در غیر این صورت با استفاده از توابع Len و Left یک کاراکتر از سمت راست رشته حذف می شود و مقدار خصوصیت Caption در متغیر strno نیز ذخیره می شود تا تغییرات انجام شده در محاسبات نیز اعمال شود.

۹ - اکنون رویداد مربوط به دکمه درصد را به صورت زیر تنظیم کنید:

```
Private Sub cmdpercent_Click()  
    lblvalue.Caption = Val(lblvalue.Caption) * 0.01  
    strno = lblvalue.Caption  
    strop = ""  
End Sub
```

در این رویداد نیز ابتدا با تبدیل خصوصیت Caption کنترل برچسب به عدد و ضرب آن در عدد ۰/۰۱ مقدار را براساس درصد محاسبه کرده و سپس مقدار به دست آمده در متغیر strno برای محاسبات بعدی ذخیره می شود و با اجرای دستور "strno =" از ایجاد تداخل و اشتباه که ممکن است در اثر انتخاب چند عملگر رخ دهد، جلوگیری به عمل می آید.

۱۰ - مانند مرحله ۸ رویداد Click مربوط به دکمه های به توان رسانی (X^2)، جذر (sqrt)، قرینه سازی (+/-) و معکوس سازی ($1/x$) را به این صورت تنظیم کنید. این چهار رویه عملکردی مشابه رویه رویداد Click دکمه درصد دارند.

```
Private Sub cmdpower_Click()  
    lblvalue.Caption = Val(lblvalue.Caption) ^ 2  
    strno = lblvalue.Caption  
    strop = ""  
End Sub
```

```
Private Sub cmdsign_Click()  
    If strno <> "" Then  
        lblvalue.Caption = -Val(lblvalue.Caption)  
        strno = lblvalue.Caption  
    End If  
End Sub
```

```
Private Sub cmdsqrt_Click()  
    lblvalue.Caption = Sqr(Val(lblvalue.Caption))  
    strno = lblvalue.Caption  
    strop= ""
```

End Sub

```
Private Sub cmdinverse_Click()  
    If Val(lblvalue.Caption) <> 0 Then lblvalue.Caption=1/ Val(lblvalue.Caption)  
    strno = lblvalue.Caption  
    strop= ""
```

End Sub

۱۱ - اکنون رویه‌های رویداد مربوط به دکمه‌های M+, MR و MC را به صورت زیر تنظیم کنید. دکمه M+ برای ذخیره اعداد و نتیجه محاسبات در حافظه ماشین حساب، دکمه MR برای استفاده از محتویات حافظه ماشین حساب در عملیات موردنظر و دکمه MC برای پاک کردن حافظه ماشین حساب به کار می‌روند.

```
Private Sub cmdm_Click()  
    If strop = "/" Or strop = "+" Or strop = "-" Or strop="*" Then  
        dblno2 = Val(strno)  
        Call compute  
    End If  
    dblmemory = dblmemory + Val(lblvalue.Caption)  
    lblmemory.Caption = "M"  
    strop= ""  
End Sub
```

```
Private Sub cmdmc_Click()  
    dblmemory = 0  
    lblmemory.Caption = ""  
End Sub
```

```
Private Sub cmdmr_Click()  
    lblvalue.Caption = str(dblmemory)  
    If dblmemory=0 Then  
        strno= ""  
    Else
```

strno=str(dbلمemory)

End If

End Sub

رویه () cmdm_Click برای ذخیره مقادیر در حافظه ماشین حساب به کار می‌رود و با یک دستور If شروع می‌شود تا در صورت استفاده از اعمال ریاضی قبل از کلیک روی دکمه M+، ابتدا محاسبات با فراخوانی رویه compute انجام شود، سپس نتیجه عملیات به محتویات قبلی حافظه ماشین حساب که در متغیر dbلمemory قرار دارد، اضافه می‌شود و با نمایش کاراکتر M در کنترل برچسب مربوطه، کاربر از ذخیره شدن اطلاعات در حافظه ماشین حساب مطلع می‌شود. البته در صورت عدم انتخاب یک عمل ریاضی دستورات بخش If اجرا نشده و طبیعی است که محاسباتی نیز انجام نمی‌شود، اما عدد موجود در نمایشگر ماشین حساب به حافظه اضافه خواهد شد. آخرین دستور در این رویه یعنی "strop=" نیز همان کاربرد قبلی در سایر رویه‌های مشابه را دارد. رویه () cmdmc_Click برای حذف مقدار ذخیره شده در حافظه ماشین حساب به کار می‌رود و با اجرای آن ابتدا مقدار dbلمemory صفر می‌شود سپس حرف M از کنترل برچسب مربوط به حافظه برداشته می‌شود.

رویه () cmdmr_Click نیز برای استفاده از مقدار ذخیره شده در حافظه در زمان انجام محاسبات به کار می‌رود و با تبدیل مقدار متغیر dbلمemory به مقدار رشته‌ای، آن را در کنترل برچسب ماشین حساب نمایش داده و بعد دستور If را برای اعمال تغییرات در محاسبات اجرا می‌کند.

۱۲ - در این مرحله رویداد دکمه‌های C و CE را به صورت زیر تنظیم کنید. همان‌طور که می‌دانید دکمه C تمام اعداد ورودی، عملگرهای انتخاب شده و محتویات حافظه ماشین حساب را پاک می‌کند و دکمه CE عمل ریاضی انتخاب شده پس از ورود اولین عدد را از بین می‌برد، اما عدد اول وارد شده را حذف نمی‌کند.

Private Sub cmdc_Click()

strno = ""

lblvalue.Caption = "0"

strop = ""

dblno1 = 0

dblno2 = 0

dblresult = 0

dbلمemory = 0

```
lblmemory.Caption = ""  
End Sub  
Private Sub cmdce_Click()  
If strop="/" Or strop="+" Or strop="-" Or strop="*" Then  
    lblvalue.Caption=dblno1  
    strno=dblno1  
    strop=""  
End If  
End Sub
```

رویداد () cmdce_Click مربوط به دکمه C می‌شود و همان‌طور که می‌بینید کلیه متغیرهای عددی، رشته‌ای، حافظه و نمایشگر ماشین حساب را پاک می‌کند.

رویداد () cmdce_Click نیز نحوه انتخاب عملگرهای ریاضی را بعد از ورود عدد اول بررسی می‌کند و در صورت درست بودن شرط، متغیر dblno1 را در خصوصیت Caption کنترل برچسب ماشین حساب قرار داده و متغیر strop را خالی می‌کند تا عملگر انتخاب شده از بین رفته و عملیات به مرحله بعد از ورود عدد اول بازگردد.

۱۳- اکنون باید رویدادهای صفحه کلید را نیز به گونه‌ای تنظیم کنید که در صورت استفاده از صفحه کلید نیز ماشین حساب قابل استفاده باشد. برای این کار ابتدا خصوصیت KeyPreview را برای فرم پروژه روی مقدار True و خصوصیت Style کلیه دکمه‌های فرمان را روی مقدار 1-Graphical تنظیم کنید، سپس رویداد KeyPress فرم را به صورت زیر تنظیم نمایید:

```
Private Sub Form_KeyPress(KeyAscii As Integer)  
    Select Case KeyAscii  
        Case 8:  
            Call cmdbackspace_Click  
            cmdbackspace.BackColor = vbRed  
            tmrdelay.Enabled = True  
        Case 48:  
            Call cmd0_Click  
            cmd0.BackColor = vbRed  
            tmrdelay.Enabled = True  
        Case 49:  
            Call cmd1_Click  
            cmd1.BackColor = vbRed
```

tmrdelay.Enabled = True

Case 50:

Call cmd2_Click

cmd2.BackColor = vbRed

tmrdelay.Enabled = True

Case 51:

Call cmd3_Click

Cmd3.BackColor = vbRed

tmrdelay.Enabled = True

Case 52:

Call cmd4_Click

cmd4.BackColor = vbRed

tmrdelay.Enabled = True

Case 53:

Call cmd5_Click

cmd5.BackColor = vbRed

tmrdelay.Enabled = True

Case 54:

Call cmd6_Click

cmd6.BackColor = vbRed

tmrdelay.Enabled = True

Case 55:

Call cmd7_Click

cmd7.BackColor = vbRed

tmrdelay.Enabled = True

Case 56:

Call cmd8_Click

cmd8.BackColor = vbRed

tmrdelay.Enabled = True

Case 57:

Call cmd9_Click

cmd9.BackColor = vbRed

tmrdelay.Enabled = True

Case 43:

Call cmdplus_Click


```
cmdplus.BackColor = vbRed
```

```
tmrdelay.Enabled = True
```

Case 45:

```
Call cmdminus_Click
```

```
cmdminus.BackColor = vbRed
```

```
tmrdelay.Enabled = True
```

Case 46:

```
Call cmddot_Click
```

```
Cmddot.BackColor = vbRed
```

```
tmrdelay.Enabled = True
```

Case 42:

```
Call cmdmultiply_Click
```

```
cmdmultiply.BackColor = vbRed
```

```
tmrdelay.Enabled = True
```

Case 47:

```
Call cmddivision_Click
```

```
cmddivision.BackColor = vbRed
```

```
tmrdelay.Enabled = True
```

Case 37:

```
Call cmdpercent_Click
```

```
cmdpercent.BackColor = vbRed
```

```
tmrdelay.Enabled = True
```

Case 67, 99:

```
Call cmdc_Click
```

```
cmdc.BackColor = vbRed
```

```
tmrdelay.Enabled = True
```

Case 69, 101:

```
Call cmdce_Click
```

```
cmdce.BackColor = vbRed
```

```
tmrdelay.Enabled = True
```

Case 80, 112:

```
Call cmdpower_Click
```

```
cmdpower.BackColor = vbRed
```

```
tmrdelay.Enabled = True
```

Case 83, 115:

```
Call cmdsqrt_Click
```

```
cmdsqr.BackColor = v
```

```
bRed
```

```
tmrdelay.Enabled = True
```

Case 73, 105:

```
Call cmdinverse_Click
```

```
cmdinverse.BackColor = vbRed
```

```
tmrdelay.Enabled = True
```

Case 68, 100:

```
Call cmdmc_Click
```

```
cmdmc.BackColor = vbRed
```

```
tmrdelay.Enabled = True
```

Case 77, 109:

```
Call cmdm_Click
```

```
cmdm.BackColor = vbRed
```

```
tmrdelay.Enabled = True
```

Case 82, 114:

```
Call cmdmr_Click
```

```
cmdmr.BackColor = vbRed
```

```
tmrdelay.Enabled = True
```

Case 71, 103:

```
Call cmdsign_Click
```

```
cmdsign.BackColor = vbRed
```

```
tmrdelay.Enabled = True
```

Case 61:

```
Call cmdequal_Click
```

```
cmdequal.BackColor = vbRed
```

```
tmrdelay.Enabled = True
```

End Select

End Sub

مشاهده می‌کنید که با استفاده از یک دستور Select Case مقدار آرگومان KeyAscii رویداد KeyPress بررسی می‌شود و با توجه به کلید فشرده شده رویداد Click متناظر با کلیدی که قبلاً طراحی شده است، فراخوانی می‌شود. این کار سبب می‌شود تا از تکرار دستورات و شلوغ شدن ماژول فرم جلوگیری شود و در هر Case رویداد Click مربوط به دکمه‌ای که فشرده شده است فراخوانی شود. پس از خاتمه اجرای رویه رویداد کلید

فشرده شده، با استفاده از خصوصیت BackColor کنترل مربوطه، رنگ کنترل به قرمز تغییر داده می‌شود. سپس خصوصیت Enabled کنترل زمان‌سنج روی True تنظیم می‌شود تا زمان‌سنج شروع به کار کند و پس از گذشت زمان کوتاهی رنگ کنترل را به حالت اول بازگرداند و خصوصیت Enabled خود را False می‌کند تا برای دفعه بعد آماده شود. به این صورت با فشردن کلیدهای صفحه کلید روی پنجره ماشین حساب، فشرده شدن کلیدها با تغییر رنگ قابل مشاهده است. سپس رویداد Timer کنترل زمان‌سنج را به صورت زیر تنظیم کنید:

Private Sub tmrdelay_Timer()

```
cmddot.BackColor = &H8000000F
cmd0.BackColor = &H8000000F
cmd1.BackColor = &H8000000F
cmd2.BackColor = &H8000000F
cmd3.BackColor = &H8000000F
cmd4.BackColor = &H8000000F
cmd5.BackColor = &H8000000F
cmd6.BackColor = &H8000000F
cmd7.BackColor = &H8000000F
cmd8.BackColor = &H8000000F
cmd9.BackColor = &H8000000F
cmdplus.BackColor = &H8000000F
cmdminus.BackColor = &H8000000F
cmdmultiply.BackColor = &H8000000F
cmddivision.BackColor = &H8000000F
cmdbackspace.BackColor = &H8000000F
cmdc.BackColor = &H8000000F
cmdce.BackColor = &H8000000F
cmdpercent.BackColor = &H8000000F
cmdpower.BackColor = &H8000000F
cmdsqrt.BackColor = &H8000000F
cmdinverse.BackColor = &H8000000F
cmdmc.BackColor = &H8000000F
cmdm.BackColor = &H8000000F
cmdmr.BackColor = &H8000000F
```

```
cmdsign.BackColor = &H8000000F
cmdequal.BackColor = &H8000000F
tmrdelay.Enabled = False
```

End Sub

۱۴- رویداد Load فرم را نیز به صورت زیر تنظیم کنید تا مقادیر متغیرها مقداردهی

اولیه شود:

```
Private Sub Form_Load()
    cmdc_Click()
```

End Sub

۱۵- نوار منو و گزینه‌های موجود در آن را مطابق شکل ۱۳-۳ و ۱۳-۴ ایجاد کنید.

روی کادر علامت Enabled گزینه Black و White کلیک کنید و آن‌ها را از حالت

انتخاب خارج کنید. به علاوه روی کادر علامت Checked گزینه Right Alignment کلیک

کنید و آن‌را انتخاب نمایید.



شکل ۱۳-۳



شکل ۱۳-۴

۱۶- اکنون به ماژول فرم بروید و رویداد Click گزینه‌های Clear و Exit را به صورت

نشان داده شده در صفحه بعد تنظیم کنید. گزینه Clear همان عملکرد دکمه C را دارد و

انتخاب گزینه Exit سبب نمایش یک کادر پیغام خواهد شد که در صورت کلیک روی

دکمه OK برنامه خاتمه یافته و در صورت کلیک روی دکمه Cancel کاربر به پنجره ماشین

حساب باز می‌گردد.

```
Private Sub mnuclear_Click()
    Call cmdc_Click
```

End Sub

```
Private Sub mnuexit_Click()  
    Dim intanswer As Integer  
    intanswer = MsgBox("Do You Want To Exit ?",vbOKCancel _  
        +vbInformation, "Exit")  
    If intanswer = vbOK Then Unload Me  
End Sub
```

۱۷- در این مرحله رویداد مربوط به گزینه‌های زیرمنوهای ForeGround و Background

را برای تغییر رنگ قلم و زمینه نمایشگر ماشین حساب، به صورت زیر تنظیم کنید:

```
Private Sub mnublack_Click()  
    lblvalue.ForeColor = vbBlack  
    mnublack.Enabled = False  
    mnublue.Enabled = True  
    mnured.Enabled = True  
End Sub
```

```
Private Sub mnublue_Click()  
    lblvalue.ForeColor = vbBlue  
    mnublack.Enabled = True  
    mnublue.Enabled = False  
    mnured.Enabled = True  
End Sub
```

```
Private Sub mnured_Click()  
    lblvalue.ForeColor = vbRed  
    mnublack.Enabled = True  
    mnublue.Enabled = True  
    mnured.Enabled = False  
End Sub
```

```
Private Sub mnuwhite_Click()  
    lblvalue.BackColor = vbWhite  
    mnuwhite.Enabled = False  
    mnugreen.Enabled = True  
    mnumagenta.Enabled = True  
End Sub
```

Private Sub mnugreen_Click()

lblvalue.BackColor = vbGreen

mnuwhite.Enabled = True

mnugreen.Enabled = False

mnumagenta.Enabled = True

End Sub

Private Sub mnumagenta_Click()

lblvalue.BackColor = vbMagenta

mnuwhite.Enabled = True

mnugreen.Enabled = True

mnumagenta.Enabled = False

End Sub

۱۸- اکنون رویداد Click مربوط به زیرمنوی Alignment را تنظیم کنید. برای این کار رویه‌های دو گزینه این زیر منو را به صورت زیر تنظیم کنید:

Private Sub mnuleft_Click()

lblvalue.Alignment = vbLeftJustify

mnuleft.Checked = True

mnuright.Checked = False

End Sub

Private Sub mnuright_Click()

lblvalue.Alignment = vbRightJustify

mnuleft.Checked = False

mnuright.Checked = True

End Sub

در این صورت با انتخاب هر یک از این گزینه‌ها، چک مارک در کنار گزینه‌ای که کلیک شده است، قرار داده می‌شود و چک مارک گزینه دیگر حذف می‌شود. به علاوه با توجه به گزینه کلیک شده، خصوصیت Alignment نمایشگر ماشین حساب نیز تنظیم می‌شود. ۱۹ - در این مرحله لازم است تا برای گزینه About در منوی Help دستورات لازم را تایپ کنید. هدف از این گزینه ارایه شماره نگارش نرم‌افزار به کاربر است، بنابراین یک فرم با مشخصات ارایه شده در جدول ۳-۱۳ و مطابق با شکل ۵-۱۳ به پروژه اضافه کنید.



شکل ۵-۱۳

جدول ۳-۱۳

کنترل خصوصیت	Image	Label	Command Button	Form
Name	imgcalculate	lblversion	cmdok	frmabout
Caption	-	Calculator Version 1-1383	&OK	About
Stretch	True	-	-	-
BorderStyle	-	۱	-	۴
Alignment	-	۲	-	-

۲۰ - پس از اضافه کردن فرم جدید آن را با نام About ذخیره کنید و سپس رویداد

Click دکمه OK و Unload فرم About را به صورت زیر تنظیم کنید:
 Private Sub cmdok_Click()

 Unload Me

End Sub

Private Sub Form_Unload(Cancel As Integer)

 frmmain.Enabled = True

End Sub

۲۱ - در این مرحله رویه رویداد Click گزینه About را از منوی Help، به صورت زیر

تنظیم کنید. اولین دستور سبب نمایش فرم About می‌شود و دستور دوم، فرم اصلی را غیرفعال می‌کند و تا زمانی که کاربر روی دکمه OK فرم About کلیک نکند یا آن را نبندد نمی‌تواند از ماشین حساب استفاده کند.

Private Sub mnuabout_Click()

 frmabout.Show

frmmain.Enabled = False

End Sub

۲۲ - پروژه و فرم اصلی را با نام calculator ذخیره کنید، سپس آنرا اجرا کرده و برای حالت‌های مختلف آزمایش کنید.

۲۳ - به اجرای پروژه خاتمه داده و پنجره ویژوال بیسیک را ببندید.