

توانایی استفاده از دستور شرطی IF و عملگرهای مقایسه‌ای و منطقی

هدف‌های رفتاری

- پس از مطالعه این واحد کار از فراگیر انتظار می‌رود که:
- ۱- بتواند از دستور If با حالت یک دستوری و چند دستوری استفاده کند.
 - ۲- عملگرهای مقایسه‌ای و منطقی و نحوه استفاده از آنها را توضیح دهد.
 - ۳- اولویت اجرای عملگرهای مقایسه‌ای و منطقی را بیان کند.
 - ۴- یک فرم جدید را به پروژه اضافه کرده، از کنترل دکمه فرمان استفاده کند و خصوصیات مربوط به کنترل دکمه فرمان را توضیح دهد.
 - ۵- بتواند با متدهای Show و Hide فرم کار کند.
 - ۶- به کمک کنترل‌های کادر تصویر و Image تصاویر مورد نظر خود را نمایش دهد.
 - ۷- بتواند از کادرهای پیام و کادرهای ورود داده استفاده کند.

کلیات

در طراحی پروژه‌های برنامه‌نویسی معمولاً نیاز به کنترل روند برنامه خواهید داشت، چرا که یک برنامه باید در شرایط مختلف عملکرد مناسبی را از خود نشان دهد و توانایی تصمیم‌گیری را با توجه به حالات و رویدادهای متفاوتی که در هنگام اجرای برنامه رخ می‌دهند، داشته باشد. در چنین شرایطی استفاده از ساختارهای تصمیم‌اجتناب‌ناپذیر است. زبان برنامه‌نویسی ویژوال بیسیک امکانات لازم را برای کنترل روند اجرای برنامه در اختیار برنامه‌نویسان قرار می‌دهد.

۱-۵ نحوه استفاده از ساختار IF و عملگرهای مقایسه‌ای در برنامه‌ها

ساختارهای تصمیم به شما اجازه می‌دهند که نحوه اجرای برنامه را به میل خود تغییر داده و تنظیم کنید. یکی از انواع دستورات کنترلی در ویژوال بیسیک دستور If است. این دستور با بررسی شرط یا شرط‌های تعیین شده و با توجه به درست یا نادرست بودن نتیجه بررسی‌ها، روند اجرای برنامه را معین می‌کند. به‌طور کلی می‌توانید از دستور If به دو صورت یک دستوری و چند دستوری استفاده کنید. شکل کلی دستور If در حالت یک دستوری به این صورت است:

یک دستور Else یک دستور Then شرط‌ها If

در این دستور، ابتدا شرط و مقایسه مورد نظری که پس از دستور If قرار می‌گیرد، بررسی می‌شود و در صورتی که نتیجه بررسی شرط درست باشد دستوری که پس از Then قرار گرفته است، اجرا می‌شود سپس دستوری که پس از If قرار گرفته اجرا می‌شود. اما در صورتی که نتیجه بررسی شرط نادرست باشد از اجرای دستوری که بعد از Then قرار گرفته است صرف نظر کرده و دستوری که پس از Else قرار گرفته است، اجرا می‌شود. سپس دستور خطی که پس از If قرار گرفته، اجرا می‌شود. البته در صورت تمایل می‌توانید از بخش Else صرف نظر کرده و از آن استفاده نکنید، در نتیجه دستور پس از Then فقط در زمان درست بودن شرط اجرا می‌شود و در غیر این صورت بدون اجرای آن، کنترل برنامه به دستوری که پس از If قرار گرفته، منتقل می‌شود. با این حال گاهی لازم است از دستور If با حالت چند دستوری استفاده کنید. شکل کلی دستور If در این حالت در

ادامه آمده است:

If (شرط(ها)) Then

⋮

دستور(ات)

Else

⋮

دستور(ات)

End If

در این حالت نیز مانند حالت یک دستوری اگر نتیجه شرطی که پس از دستور If قرار گرفته است، درست باشد دستورات بین Then و Else اجرا می‌شوند و سپس دستوری که پس از End If قرار گرفته است، اجرا می‌شود. اما در صورتی که نتیجه بررسی شرط نادرست باشد از اجرای دستورات بین Then و Else صرف نظر می‌شود و دستورات بین بخش Else و End If اجرا می‌شوند.

اکنون دستوری که پس از End If قرار دارد، اجرا می‌شود. البته در این حالت نیز استفاده از بخش Else اختیاری است و می‌توانید از این دستور به صورت زیر نیز استفاده کنید:

If شرط‌ها Then

⋮

دستور(ات)

End If

در این حالت اگر نتیجه بررسی شرط درست باشد، دستورات بین Then و End If اجرا می‌شوند و سپس دستوری که پس از End If قرار دارد، اجرا می‌شود و اگر نتیجه این بررسی نادرست باشد از اجرای دستورات موجود بین Then و End If صرف نظر شده و اجرای برنامه به دستور پس از End If منتقل می‌شود.



مثال ۱: پروژه‌ای طراحی کنید که یک عدد دلخواه را دریافت کرده و زوج یا فرد

بودن آن را مشخص کند. برای این منظور عملیات زیر را به ترتیب انجام دهید:

۱- برنامه ویژوال بیسیک را اجرا کنید و یک پروژه جدید از نوع Standard EXE به همراه یک فرم و کنترل‌های آن را مطابق شکل ۱-۵ ایجاد کنید و خصوصیت‌های آن‌ها را مطابق جدول‌های

۵-۱ و ۵-۲ تنظیم نمایید:



شکل ۵-۱

جدول ۵-۱ خصوصیات فرم

مقدار	خصوصیت
Frmeven or odd	Name
Even or Odd	Caption

جدول ۵-۲ خصوصیات کنترل‌ها

کنترل خصوصیت	Label	Label	Text Box	Command Button
Name	lblno	lblresult	txtno	cmdevenodd
Caption	Enter No:	---	---	&Even or Odd

۲- در این مرحله رویداد Click دکمه فرمان cmdevenodd را به صورت زیر تنظیم کنید:

```
Private Sub cmdevenodd_Click()
    Dim intno As Integer
    Dim R As Integer
    Intno = Val (txtno. Text)
    R= intno Mod 2
    If R= 0 Then
        lblresult. Caption = " Your Number Is Even."
    Else
        lblresult. Caption = "Your Number Is Odd."
    End Sub
```

در این رویداد ابتدا رشته عددی که در کنترل کادر متن txtno تایپ می شود با استفاده از تابع Val به عدد تبدیل شده و در متغیر ذخیره می شود سپس با استفاده از عملگر Mod باقیمانده تقسیم متغیر intno را بر عدد ۲ محاسبه می کند و در متغیر R قرار می دهد.

در این مرحله دستور If شرط $R=0$ را بررسی می‌کند اگر مقدار R برابر با صفر باشد عدد زوج است و نتیجه بررسی شرط درست (True) می‌شود بنابراین پیام "Your Number Is Even." نمایش داده خواهد شد اما اگر عدد فرد باشد مقدار R برابر با ۱ خواهد شد بنابراین نتیجه بررسی شرط نادرست (False) می‌شود و پیام "Your Number Is Odd" نمایش داده می‌شود.

۳ - فرم و پروژه را با نام even or odd ذخیره کنید.

۴ - پروژه را اجرا کنید و برای اعداد زوج و فرد آن را آزمایش کنید.

۵ - اجرای پروژه را متوقف کرده و به پنجره ویژوال بیسیک بازگردید.


تمرین:



پروژه مثال ۱ را به‌گونه‌ای تنظیم کنید تا برای اعداد منفی و صفر با نمایش یک پیام خطا کاربر را مطلع نماید.

۲-۵ کنترل کادر تصویر (Picture Box)

یکی دیگر از ویژگی‌های زبان ویژوال بیسیک امکان استفاده آسان از تصاویر، فایل‌های گرافیکی و عکس‌ها در برنامه‌هاست. برای نمایش تصاویر می‌توانید از کنترل کادر تصویر استفاده کنید. این کنترل می‌تواند انواع فایل‌های گرافیکی مانند bmp, gif, jpg, ico و cur را نمایش دهد.

مهم‌ترین خصوصیت کنترل کادر تصویر Picture است. این خصوصیت مسیر و نام فایل گرافیکی یا تصویر را مشخص می‌کند. برای این کار کافی است روی دکمه  در روبه‌روی خصوصیت مزبور کلیک کنید و به‌وسیله کادر محاوره Load Picture تصویر موردنظرتان را انتخاب کنید.

کنترل کادر تصویر دارای خصوصیتی به نام Align می‌باشد. این خصوصیت می‌تواند محل نمایش تصویر را تعیین کند. مقادیری که این خصوصیت می‌تواند کسب کند در جدول ۳-۵ ارائه شده‌اند.

جدول ۳-۵ مقادیر خصوصیت Align

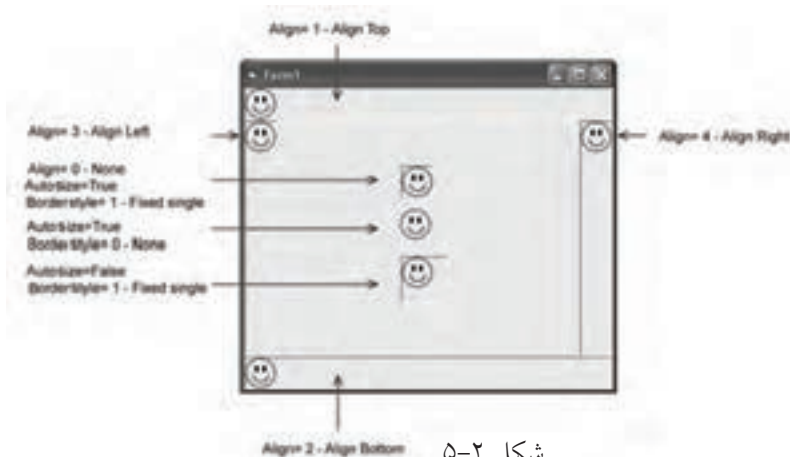
مقدار خصوصیت	توضیح
0 - None	نمایش کنترل با استفاده از مقادیر خصوصیت‌های Top و Left
1- Align Top	نمایش کنترل در بالای فرم
2-Align Bottom	نمایش کنترل در پایین فرم
3 -Align Left	نمایش کنترل در سمت چپ فرم
4 -Align Right	نمایش کنترل در سمت راست فرم

یکی دیگر از خصوصیت‌های کادر تصویر خصوصیت AutoSize است و می‌تواند مقدار True یا False را کسب کند. در صورتی که مقدار این خصوصیت را روی True تنظیم کنید اندازه کنترل کادر تصویر با اندازه تصویر به صورت خودکار یکسان می‌شود اما اگر مقدار این خصوصیت روی False تنظیم شود اندازه کنترل کادر تصویر با توجه به مقدار خصوصیت‌های Height و Width تنظیم می‌شود. به عبارت دیگر در این حالت تصویر با اندازه خود و کنترل کادر تصویر نیز با اندازه خود نمایش داده می‌شوند.

کنترل کادر تصویر دارای خصوصیت دیگری به نام BorderStyle است. با استفاده از این خصوصیت می‌توانید کنترل کادر تصویر را با یک کادر در اطراف آن یا بدون کادر نمایش دهید. خصوصیت BorderStyle می‌تواند مقادیر ارائه شده در جدول ۴-۵ را کسب کند.

جدول ۴-۵

مقدار خصوصیت	توضیح
0 -None	کنترل کادر تصویر بدون کادر نمایش داده می‌شود.
1 -Fixed Single	کنترل کادر تصویر با کادر نمایش داده می‌شود.



شکل ۲-۵

کنترل کادر تصویر علاوه بر رویدادهای مشترکی که با سایر کنترل‌ها دارد، دارای دو رویداد Change و Resize می‌باشد. رویداد Change وقتی رخ می‌دهد که تصویر نمایش داده شده به وسیله کنترل کادر تصویر تغییر کند و رویداد Resize نیز زمانی اجرا می‌شود که اندازه کنترل تغییر می‌کند.



مثال ۲: پروژه‌ای طراحی کنید که یک تصویر را روی یک فرم نمایش دهد. برای این منظور ابتدا فرم اصلی را طراحی کرده، آن را کامل کنید، سپس عملیات بعد را به ترتیب انجام دهید:


۱ - برنامه ویژوال بیسیک را اجرا کنید و یک پروژه جدید از نوع Standard EXE به همراه یک فرم مطابق شکل ۳-۵ ایجاد کنید که مشخصات فرم آن مطابق جدول ۵-۵ باشد.



شکل ۳-۵

جدول ۵-۵ خصوصیات فرم

مقدار	خصوصیت
frmshow	Name
Show Picture	Caption

۲ - برای نمایش تصویر می‌توانید از کنترل کادر تصویر (PictureBox) استفاده کنید. برای استفاده از این کنترل ابتدا به پنجره طراحی Form1 بروید و در جعبه ابزار روی آیکن PictureBox  دابل کلیک کنید تا کنترل کادر تصویر روی فرم قرار گیرد، سپس خصوصیات کنترل کادر تصویر را مطابق جدول ۶-۵ تنظیم کنید.

جدول ۶-۵

مقدار	خصوصیت
picshow	Name
True	AutoSize

۳ - اکنون باید تصویر نمایشی را برای کنترل کادر تصویر انتخاب کنید، بنابراین روی کنترل کادر تصویر کلیک نمایید و در پنجره خصوصیات، خصوصیت Picture این کنترل را پیدا کنید.


۴ - روی دکمه **...** در روبه روی این خصوصیت کلیک کنید و با استفاده از کادرمحاوره Load Picture فایل EARTH.ICO را از مسیر زیر انتخاب کنید، همان طور که مشاهده خواهید کرد تصویر انتخاب شده روی فرم قابل مشاهده است.

D:\Program Files\Microsoft Visual Studio\Common\Graphics\Icons\Elements

۵ - یک کنترل دکمه فرمان روی فرم قرار داده سپس ویژگی های آن را با توجه به جدول ۷-۵ تنظیم کنید. از این کنترل برای خروج از برنامه استفاده می شود.

جدول ۷-۵

مقدار	خصوصیت
cmdexit	Name
&Exit	Caption

 **نکته** برای آن که یک کلید دسترسی سریع برای یک کنترل ایجاد کنید، کافی است که در خصوصیت Caption آن و قبل از حرف مورد نظر یک کاراکتر & تایپ کنید، در این صورت فشردن کلید ترکیبی Alt به همراه حرفی که کاراکتر & قبل از آن تایپ شده است، معادل کلیک کردن روی کنترل دکمه فرمان خواهد بود.

۶ - اکنون باید رویداد مناسبی را انتخاب کنید تا با استفاده از آن و در زمان کلیک روی دکمه فرمان Exit، برنامه خاتمه یابد. بنابراین در پنجره طراحی فرم روی دکمه فرمان Exit دابل کلیک کنید تا پنجره کد فرم فعال شود و رویداد () cmdexit_Click در اختیار شما قرار گیرد. سپس دستور زیر را در این رویداد تایپ کنید:

Unload frmshow

در زمان اجرا وقتی کاربر روی دکمه Exit کلیک کند رویداد () cmdexit_Click به دنبال آن دستور Unload اجرا خواهد شد، این دستور اطلاعات فرم frmshow را از حافظه خارج می کند، به عبارت دیگر برنامه خاتمه می پذیرد.

۷- برنامه را اجرا کنید و پس از مشاهده فرم و تصویر نمایشی روی دکمه Exit کلیک کنید تا برنامه خاتمه یابد.

۸- پروژه و فرم را با نام showpicture ذخیره کنید.



مثال ۳: اکنون پروژه showpicture را کامل کنید تا در هنگام اجرای برنامه، ابتدا یک کلمه رمز سؤال شود و در صورت صحیح بودن کلمه رمز، اجازه ورود کاربر به برنامه داده شود و فرم اصلی برنامه با تصویر مورد نظر نمایش داده شود. برای این منظور عملیات زیر را انجام دهید:

۱- به پنجره ویژوال بیسیک و پروژه showpicture بروید.

۲- برای اضافه کردن یک فرم جدید، در مکان خالی از پنجره پروژه کلیک راست کنید و گزینه Add را انتخاب کنید، سپس روی گزینه Form کلیک کنید (شکل ۵-۵).



شکل ۵-۴



شکل ۵-۵


کادر محاوره Add Form مطابق شکل ۵-۶ نمایش داده می‌شود که دو زبانه New و Existing در آن دیده می‌شود. برای ایجاد یک فرم جدید می‌توانید از زبانه New و برای اضافه کردن یک فرم که قبلاً ایجاد و در روی دیسک ذخیره شده است، از زبانه Existing استفاده کنید.



شکل ۵-۶

البته می‌توانید فرم‌ها را با یکی از این روش‌ها نیز اضافه کنید:

الف- برای دسترسی به کادر محاوره Add Form می‌توانید گزینه Add Form را از منوی Project در نوار منوی ویژوال بیسیک انتخاب کنید.

ب- برای دسترسی به کادر محاوره Add Form می‌توانید روی علامت مثلثی شکل دکمه  در نوار ابزار استاندارد کلیک کنید و سپس گزینه Form را انتخاب کنید.

۳- در کادر محاوره Add Form ابتدا زبانه New را انتخاب کنید، سپس از کادر لیست موجود در زیرزبانه‌ها، روی آیکن Form و بعد روی دکمه Open کلیک کنید. همان‌طور که مشاهده می‌کنید یک پنجره طراحی دیگر در پنجره ویژوال بیسیک نمایش داده می‌شود به علاوه در پنجره‌های پروژه، خصوصیات و تعیین موقعیت نیز، فرم اضافه شده قابل مشاهده است.

۴- ویژگی‌های فرم جدید را تنظیم کنید و سپس آن‌را با نام Log On در روی دسک‌تاپ ذخیره کنید.

۵- اکنون باید یک کنترل کادر متن، برچسب و دکمه فرمان با مشخصات ارائه شده در جدول ۶-۹ به پنجره frmlogon اضافه کنید.

۶- حال باید عملکرد برنامه را طوری تنظیم کنید تا اگر عبارتی که در کادر متن تایپ می‌شود با کلمه رمز برابر باشد، امکان ادامه اجرای برنامه فراهم شود. برای این کار از رویداد کلیک کنترل دکمه فرمان استفاده می‌شود. این رویداد زمانی اجرا می‌شود که کاربر روی این دکمه کلیک کند یا کلید Enter را بفشارد. بنابراین کنترل cmdok را در پنجره طراحی

فرم frmlogon انتخاب کنید، سپس روی این کنترل دابل کلیک کنید. همان‌طور که مشاهده می‌کنید پنجره کد فرم نمایش داده می‌شود و رویداد cmdok_Click در دسترس شما قرار می‌گیرد. با فرض این که کلمه رمز BASIC باشد، رویداد Click این کنترل را به صورت زیر تنظیم کنید:

```
Private Sub cmdok_Click()
```

```
    If txtpass.Text = "BASIC" Then frmshow.Show
```

```
End Sub
```

در رویداد cmdok_Click از دستور If استفاده شده است. در این دستور کلمه رمزی که کاربر در کنترل کادر متن تایپ کرده و در خصوصیت Text کادر متن نگهداری می‌شود با استفاده از علامت تساوی (=) با عبارت "BASIC" که به عنوان کلمه رمز، در نظر گرفته شده است، مقایسه می‌شود. اگر کلمه رمز تایپ شده با عبارت "BASIC" برابر باشد، دستوری که بعد از کلمه کلیدی Then قرار دارد، اجرا می‌شود و در غیر این صورت یعنی اشتباه بودن کلمه رمز، دستوری که بعد از Then قرار گرفته است، اجرا نمی‌شود و دستور خط بعد از If اجرا خواهد شد.

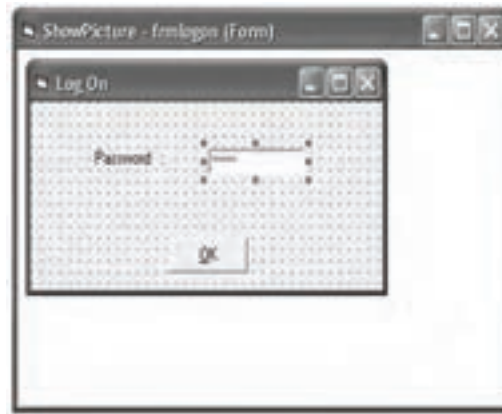
اکنون به بررسی دستور frmshow.Show که بعد از کلمه کلیدی Then قرار گرفته است، می‌پردازیم. Show متدی است که می‌تواند یک فرم را که اطلاعات آن در حافظه بارگذاری شده است، نمایش دهد. البته در صورتی که فرم در حافظه بارگذاری نشده باشد آن را بارگذاری کرده و سپس نمایش می‌دهد.



نکته متد Hide نیز از انواع متدهای فرم است و عملکرد آن برعکس متد Show می‌باشد و از نمایش فرم جلوگیری کرده و آن را مخفی می‌کند، اما اطلاعات فرم را از حافظه خارج نمی‌کند.

۷- می‌دانید که در زمان تایپ کلمه رمز در مکان مربوطه، برای جلوگیری از مشاهده کلمه رمز توسط سایر کاربران، به جای کاراکترهایی که تایپ می‌شوند کاراکتر دیگری نمایش داده می‌شود. به این منظور می‌توانید از خصوصیت PasswordChar کادر متن استفاده کنید. این خصوصیت کاراکتر تعیین شده را به جای کاراکترهایی که در کادر متن تایپ می‌شوند، نمایش می‌دهند، اما محتویات تایپ شده در کادر متن به همان صورت در

خصوصیت Text کادر متن نگهداری می‌شود. بنابراین خصوصیت فوق را پیدا کرده و در کادر متن روبه‌روی این خصوصیت کاراکتر ستاره (*) را تایپ کنید. همان‌طور که مشاهده می‌کنید پنج کاراکتر ستاره در پنجره طراحی فرم و در کادر متن نمایش داده می‌شود. به عبارت دیگر به تعداد کاراکترهای عبارت موجود در خصوصیت Text کادر متن، کاراکتر ستاره نمایش داده می‌شود (شکل ۷-۵).



شکل ۷-۵

۸- در این مرحله برای تعداد کاراکترهای کلمه رمز، محدودیت ایجاد کنید. خصوصیت MaxLength کادر متن، حداکثر تعداد کاراکترهایی را که در این کنترل می‌توان تایپ کرد، معین می‌کند. مقدار پیش فرض این خصوصیت صفر است که تعداد کاراکترهای ورودی نامحدود را پشتیبانی می‌کند. مقدار این خصوصیت را برای کنترل txtpass روی مقدار ۵ تنظیم کنید. در نتیجه هنگام اجرای برنامه می‌توانید حداکثر ۵ کاراکتر را در کادر متن تایپ کنید.

۹- دستور زیر را در رویداد Load فرم frmlogon تایپ کنید تا در هنگام نمایش این فرم کاربر مجبور نباشد مقدار پیش فرض Text_۱ را در کادر متن پاک کند و کلمه رمز را تایپ نماید.

```
txtpass.Text = ""
```

۱۰- در این مرحله تغییرات ایجاد شده در فرم‌ها و پروژه را ذخیره کرده و پروژه را اجرا کنید.


همان‌طور که می‌بینید پنجره اصلی برنامه یعنی فرم frmshow نمایش داده شده و فرم frmlogon مشاهده نمی‌شود. در واقع در هر برنامه می‌توان یک فرم را به عنوان فرمی که

در ابتدای اجرای برنامه نمایش داده می‌شود، معرفی کرد. اگر از چند فرم در پروژه خود استفاده می‌کنید باید فرم مورد نظر را برای نمایش در ابتدای اجرای برنامه معین کنید.

برای این کار باید از کادر محاوره Project Properties استفاده کنید.




شکل ۵-۸

۱۱ - اجرای برنامه را متوقف کنید و از منوی Project گزینه ShowPicture Properties را انتخاب کنید تا کادر محاوره Project Properties نمایش داده شود. در زبانه General و در کادر لیست Startup Object روی دکمه  کلیک کنید (شکل ۵-۸). سپس از کادر لیستی که نمایش داده می‌شود، فرم frmlogon را برگزیده و روی دکمه OK کلیک کنید.

۱۲ - برنامه را اجرا کنید و عبارت BASIC را در کادر متن تایپ کنید، سپس روی دکمه OK کلیک کنید. همان‌طور که می‌بینید فرم frmshow نمایش داده می‌شود و هر دو فرم قابل مشاهده خواهند بود.

۱۳ - پنجره‌های برنامه را ببندید و بار دیگر برنامه را اجرا کنید و این بار کلمه رمز را به صورت basic تایپ کنید و روی دکمه OK کلیک کنید. همان‌طور که مشاهده خواهید کرد فرم برنامه نمایش داده نخواهد شد.

۱۴ - اجرای برنامه را متوقف کنید و به پنجره ویژوال بیسیک بازگردید.

 **نکته** در صورتی که بخواهید یک متغیر در تمام فرم‌های پروژه قابل دسترسی و شناسایی باشد می‌توان آن را با استفاده از کلمه کلیدی Public در بخش تعاریف ماژول فرم تعریف نمود. به این نوع از متغیرها، متغیر عمومی می‌گویند. شکل کلی نحوه تعریف متغیرهای عمومی به صورت زیر است:

نوع داده As نام متغیر Public

به عنوان مثال به این نمونه توجه کنید:

Public curpament As Currency

برای دسترسی به متغیرهای عمومی در سایر فرم‌ها می‌توان نام فرمی را که متغیر در آن تعریف شده است را قبل از نام متغیر قرار داد. به عنوان مثال اگر متغیر curpament در بخش تعاریف فرمی با نام frmmain تعریف شده باشد برای مقداردهی به آن در فرم frmSalary در همان پروژه می‌توان به این صورت عمل کرد:

frmmain. Curpament = 2000

تمرین:



۱- برنامه را به شکلی تنظیم کنید که با کلیک روی دکمه Exit تمام پنجره‌ها به‌طور هم‌زمان بسته شوند.

۲- برنامه را به گونه‌ای تنظیم کنید که در هنگام خروج از برنامه، پنجره کوچکی به همراه دو دکمه Yes و No مبنی بر تأیید برای خروج از نرم‌افزار نمایش داده شود. در صورتی که کاربر روی دکمه Yes کلیک کند، برنامه خاتمه یابد و در صورتی که روی دکمه No کلیک کند، به برنامه بازگردد.

در این جا لازم است با انواع عملگرهای مقایسه‌ای آشنا شوید. برای انجام هرگونه مقایسه می‌توانید از این عملگرها مطابق جدول ۸-۵ استفاده کنید.

جدول ۸-۵ عملگرهای مقایسه‌ای ویژوال بیسیک

عملگر	مفهوم عملگر	مثال	نتیجه مقایسه
=	تساوی	"Ali"="BASIC"	نادرست
<=	کوچک‌تر یا مساوی	"BASIC"<="basic"	درست
>=	بزرگ‌تر یا مساوی	"BASIC">="Basic"	نادرست
>	کوچک‌تر	۱۰ > ۵	نادرست
<	بزرگ‌تر	۱۴/۵ > ۲/۷۵	درست
<>	عدم تساوی یا نامساوی	-۲ <> -۲	نادرست



نکته در مقایسه کاراکترها و عبارات رشته‌ای، از کد اسکی کاراکترها برای انجام مقایسه استفاده می‌شود.

عملگرهای مقایسه‌ای را می‌توانید برای مقایسه هر نوع داده‌ای استفاده کنید، با توجه به این‌که مقادیری که مقایسه می‌شوند از یک نوع باشند.

گاهی اوقات ممکن است از چند عملگر مقایسه‌ای در یک عبارت استفاده شود. در این صورت این سؤال مطرح می‌شود که کدام یک از عملگرها زودتر پردازش می‌شوند. در واقع در یک عبارت مقایسه‌ای با چند عملگر، عملیات با توجه به ترتیب قرار گرفتن عملگرها از چپ به راست عبارت انجام می‌شود و اگر از پرانتز استفاده شود، اولویت و حق تقدم عملیات با عملگرهای موجود در داخل پرانتز خواهد بود. به عنوان مثال این عبارت را در نظر بگیرید:

در عبارت مذکور ابتدا مقایسه $2 > 10$ بررسی می‌شود که نتیجه نادرست یا False را دربر خواهد داشت، سپس مقایسه $0 = 0$ بررسی می‌شود که نتیجه آن درست یا True خواهد بود.

تمرین:



- ۱ - پروژه showpicture را به شکلی تنظیم کنید تا در صورتی که کلمه رمز وارد شده صحیح باشد، فرم frmlogon از روی دسک‌تاپ برداشته شده و سپس فرم frmshow مشاهده شود، هم‌چنین در صورت نادرست بودن کلمه رمز رنگ کاراکترهای تایپ شده قرمز شود و کاربر توانایی ورود کلمه رمز دیگری را نداشته باشد. در ضمن در صورت ورود کلمه رمز به صورت اشتباه دکمه OK غیرفعال شود و رنگ زمینه کادر متن نیز به رنگ آبی درآید.
- ۲ - پروژه‌ای طراحی کنید که کاربر با تایپ اسامی کشورها بتواند سایر مشخصات مانند وسعت، جمعیت، پرچم، مذهب و واحد پول آن‌ها را نمایش دهد و هم‌چنین بتواند برای ورود به نرم‌افزار یک کلمه کاربر و یک کلمه رمز را وارد کند.

۳-۵ نحوه استفاده از عملگرهای منطقی برای ترکیب شرطها

گاهی اوقات ممکن است لازم باشد در دستورات شرطی بیش از یک شرط مورد بررسی قرار گیرد و براساس نتیجه حاصل از بررسی تمام شرطها اجرای برنامه صورت گیرد در این حالت می توان از عملگرهای منطقی مانند AND، OR یا NOT استفاده کرد. این عملگرها امکان ترکیب دو یا چند شرط را با یکدیگر فراهم می کنند.



مثال ۴: یکی دیگر از موارد قابل توجه در مثال قبل، این است که ویژوال بیسیک بین حروف کوچک و بزرگ تفاوت قائل می شود، به عنوان مثال عبارت BASIC را با عبارت basic مساوی در نظر نمی گیرد. البته اگر بخواهید می توانید این حالت را تغییر دهید و به ویژوال بیسیک بگویید که بین حروف کوچک و بزرگ تفاوت قائل نشود. به این منظور باید دستور Option Compare Text را در بخش تعاریف (Declarations Section) در پنجره کد فرم frmlogon بنویسید (شکل ۹-۵).

```
Option Compare Text
Private Sub cmdok_Click()
    If txtpass.Text = "BASIC" Then
        Valued frmlogon
        frmshow.Show
    Else
        txtpass.Locked = True
    End If
End Sub

Private Sub Form_Load()
    txtpass.Text = ""
    cmdok.Enabled = False
End Sub
```

شکل ۹-۵

برنامه را اجرا کنید و کلمه رمز را به صورت basic تایپ کنید و روی دکمه OK کلیک کنید. همان طور که مشاهده خواهید کرد این بار کلمه رمز پذیرفته می شود. روی دکمه Exit کلیک کنید تا برنامه خاتمه یابد، سپس تغییرات را ذخیره نمایید. البته این کار را می توانید به روش دیگری نیز انجام دهید که در این جا به ذکر آن می پردازیم. با استفاده از

عملگرهای منطقی در ویژوال بیسیک می‌توانید ترکیب‌های مختلفی از چند شرط را ایجاد کنید، به عنوان مثال در پروژه قبل برای آن‌که برنامه هر دو نوع عبارت BASIC یا basic را به عنوان کلمه رمز بپذیرد، می‌توانید خط اول از دستور If موجود در رویداد cmdok_Click را به این صورت تنظیم کنید:

```
If txtpass.Text = "BASIC" Or txtpass.Text = "basic" Then
```

سپس دستور Option Compare Text را از بخش تعاریف حذف کنید و پروژه را با دو حالت مختلف کلمه رمز آزمایش کنید و نتیجه را بررسی نمایید. همان‌طور که ملاحظه کردید در هر دو بار کلمه رمز پذیرفته می‌شود.

عملگر Or ("یا" منطقی) یکی از عملگرهای منطقی در ویژوال بیسیک است. از عملگرهای منطقی برای ایجاد ترکیب‌های مورد نظر از چند شرط مختلف، استفاده می‌شود. در صورتی که دو یا چند شرط با استفاده از عملگر منطقی Or با هم ترکیب شوند، نتیجه بررسی تمام ترکیب شرطی، زمانی درست است که حداقل یکی از شرط‌ها درست باشد یا به عبارت دیگر زمانی ترکیب شرطی نتیجه نادرست خواهد داشت که تمام شرط‌ها نادرست باشند. نتیجه ترکیب شرطی که با عملگر Or به وجود می‌آید در

جدول ۵-۹

A	B	A Or B
نادرست	نادرست	نادرست
درست	نادرست	درست
نادرست	درست	درست
درست	درست	درست

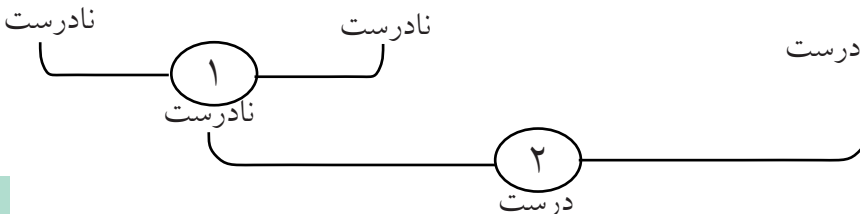
حالت‌های مختلف در جدول ۵-۹ ارایه شده است. منظور از A و B در این جدول عبارت‌های مقایسه‌ای است که با عملگر Or ترکیب شرطی را به وجود آورده و می‌تواند ارزش درست و یا نادرست داشته باشند.

به عنوان مثال این عبارت را در نظر بگیرید:

```
"ALI" = "ali" Or "d" < "B" Or "Tehran" > = "TABRIZ"
```

در عبارت فوق از دو عبارت مقایسه‌ای استفاده شده است که با عملگر منطقی Or با یکدیگر ترکیب شده‌اند، پس برای به دست آوردن نتیجه این ترکیب به صورت زیر عمل می‌کنید:

```
"ALI" = "ali" Or "d" < "B" Or "Tehran" > = "TABRIZ"
```



علاوه بر عملگر منطقی Or دو عملگر منطقی دیگر که معمولاً مورد استفاده برنامه‌نویسان قرار می‌گیرند عبارتند از: عملگر منطقی And ("و" منطقی) و عملگر منطقی Not (نقیض).

نتیجه عبارت شرطی که با And ایجاد می‌شود در جدول ۵-۱۰ ارایه شده است. همان‌طور که ملاحظه می‌کنید فقط زمانی ترکیب شرطی نتیجه درست خواهد داشت که نتیجه هر دو مقایسه و شرط به طور هم‌زمان درست باشد، در غیر این صورت ترکیب شرطی، نتیجه نادرست خواهد داشت.

نتیجه ترکیب شرطی که عملگر Not روی آن انجام می‌شود، مطابق جدول ۵-۱۱ است. همان‌طور که مشاهده می‌کنید عملگر Not می‌تواند ارزش یک شرط یا ترکیب شرطی را معکوس کند.

جدول ۵-۱۰

A	B	A And B
درست	درست	درست
نادرست	درست	نادرست
درست	نادرست	نادرست
نادرست	نادرست	نادرست

جدول ۵-۱۱

A	Not A
درست	نادرست
نادرست	درست

اکنون ارزش درستی این عبارت را به‌دست آورید:

Not ("ali" <= "ALI") Or ("Reza" > "R") And ("Computer" <> "computer")

قبل از این‌که ارزش درستی عبارت قبل را به‌دست آورید، ذکر این نکته ضروری است که عملگرهای منطقی نیز مانند عملگرهای مقایسه‌ای، نسبت به هم اولویت اجرایی دارند و اگر از عملگرهای مشابه در یک عبارت استفاده شود، حق تقدم و اولویت اجرای آن‌ها از چپ به راست خواهد بود، اما اگر از عملگرهای متفاوت استفاده شود، حق تقدم اجرای آن‌ها با توجه به جدول ۵-۱۲ و از بالا به پایین خواهد بود.

جدول ۵-۱۲

Not
And
Or

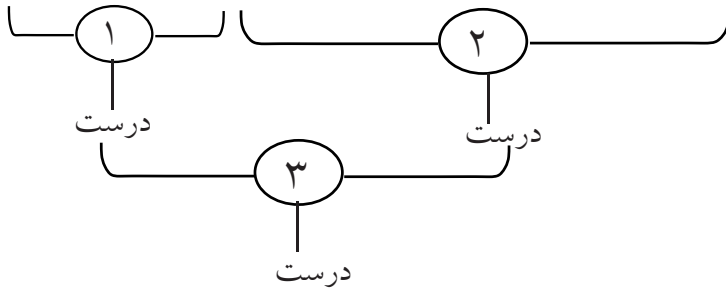
اولویت اجرای عملگرهای مقایسه‌ای از عملگرهای منطقی بالاتر است.



با توجه به مطالب گفته شده نحوه ارزیابی عبارت مورد نظر به این صورت خواهد بود:

Not ("ali" <= "ALI") Or ("Reza" > "R") And ("Computer" <> "computer")

Not (نادرست) Or (درست) And (درست)



تمرین:

در صورتی که مقدار $A=2$ ، $B=-14$ و $C=0$ باشد، ارزش درستی این عبارات را به دست

آورید:

$A < C$ Or Not $B <> C$ AND "A" = "Q" Or $C <= 2$

"6" > "9" AND $A > B$ Or Not $C > B$

۴-۵ کنترل تصویر

ویژوال بیسیک کنترل دیگری به نام تصویر Image دارد که مانند کادر تصویر برای نمایش تصاویر به کار می‌رود. خصوصیات این کنترل مشابه کنترل کادر تصویر است، اما کنترل تصویر (Image) یک خصوصیت به نام Stretch دارد که در کنترل کادر تصویر وجود ندارد. این خصوصیت از نوع منطقی بوده و در صورتی که روی مقدار False تنظیم شود، اندازه کنترل با اندازه تصویر هماهنگ می‌شود. اما اگر این خصوصیت روی مقدار True تنظیم شود اندازه تصویر با توجه به اندازه کنترل تنظیم می‌شود. می‌توانید از این خصوصیت برای بزرگ و کوچک کردن تصاویر استفاده کنید.



مثال ۵: پروژه‌ای طراحی کنید که کاربر بتواند هر فایل گرافیکی مورد نظر خود را

در یک پنجره مطابق شکل ۱۰-۵ نمایش دهد.



شکل ۵-۱۰

۱- برنامه ویژوال بیسیک را اجرا کنید و یک پروژه از نوع Standard EXE ایجاد کنید، سپس کنترل‌های مربوطه را مطابق جدول ۵-۱۳ ایجاد کنید و فرم و پروژه را با نام Image ذخیره نمایید.

جدول ۵-۱۳

کنترل / خصوصیت	Form	TextBox	Label	Command Button	Image	Command Button
Name	frmimage	txtfilename	lblfilename	cmdok	imgshow	cmdexit
Caption	Image	—	Path and File Name:	&OK	—	Exit
Stretch	—	—	—	—	True	—
AutoSize	—	—	True	—	—	—

۲- به پنجره کد فرم بروید و رویداد Load فرم و Change کادر متن را به صورت زیر تنظیم کنید. این دستورات باعث می‌شوند تا زمانی که کاربر، نام و مسیر فایل را در کادر متن تایپ نکند، دکمه OK فعال و قابل استفاده نباشد.

```
Private Sub Form_Load()  
    cmdok.Enabled = False  
End Sub  
Private Sub txtfilename_change()  
    cmdok.Enabled = True  
End Sub
```

۳ - این دستور را در رویداد Click دکمه Exit تایپ کنید:

```
Unload frmimage
```


ذکر این نکته ضروری است که برای خروج اطلاعات فرم فعال از حافظه و پایان دادن به برنامه می‌توانید از دستور Unload Me استفاده کنید، کلمه کلیدی Me به نام فرم فعال اشاره می‌کند.

۴ - برای آن‌که پس از تایپ نام، مسیر فایل و کلیک روی دکمه OK، تصویر مورد نظر نمایش داده شود، رویداد Click دکمه فرمان OK را به این صورت تنظیم کنید:

```
Private Sub cmdok_Click()  
    imgshow.Picture= LoadPicture (txtfilename.Text)  
End Sub
```

با کلیک روی دکمه OK این رویداد اجرا شده و تصویر مورد نظر به وسیله تابع LoadPicture نمایش داده می‌شود. تابع LoadPicture می‌تواند نام و مسیر یک فایل گرافیکی را دریافت کند و آنرا در یک کنترل کادر تصویر (PictureBox) یا تصویر (Image) نمایش دهد. شکل کلی نحوه استفاده از این تابع به صورت زیر است:

("نام و مسیر فایل گرافیکی") Picture = LoadPicture نام کنترل تصویر یا کادر تصویر

 **نکته** باید نام و مسیر فایل را داخل کاراکتر " قرار دهید. اگر از تابع فوق به صورت ("") Load Picture استفاده کنید تصویر نمایش داده شده در کنترل کادر تصویر یا کنترل تصویر، برداشته شده و حذف می‌شود.

۵ - پروژه را اجرا کرده و نام و مسیر یک فایل گرافیکی را در کادر متن بنویسید و روی دکمه OK کلیک کنید.

۶ - اجرای پروژه را خاتمه دهید و به پنجره ویژوال بیسیک بازگردید.

۵-۵ اولویت اجرای عملگرها نسبت به یکدیگر

تاکنون اولویت اجرای عملگرها در گروه خود را فرا گرفتید. عملگرهای موجود در گروه‌های مختلف نیز در صورتی که در یک عبارت قرار بگیرند، دارای اولویت اجرایی نسبت به هم هستند، اولویت عملگرها در تمام گروه‌ها در جدول ۵-۱۴-۱۴ ارایه شده‌اند. بالاترین اولویت مربوط به عملگرهای ریاضی و پایین‌ترین اولویت مربوط به عملگرهای منطقی می‌شود.

جدول ۵-۱۴

عملگرهای ریاضی براساس اولویت‌های ذکر شده
عملگر اتصال رشته‌ها (&)
عملگرهای مقایسه‌ای براساس اولویت‌های ذکر شده
عملگرهای منطقی براساس اولویت‌های ذکر شده

۵-۶ نحوه استفاده از کادرهای پیام در ویژوال بیسیک

احتمالاً تاکنون با کادرهای پیام در ویندوز برخورد کرده‌اید، کادرهای پیام معمولاً برای تأیید یا لغو عملیات یا مطلع کردن کاربران از یک اتفاق یا خطا استفاده می‌شوند و معمولاً پیامی را به همراه یک یا چند دکمه به کاربر نمایش می‌دهند تا با توجه به نیاز خود تصمیم‌گیری مناسب را انجام دهد. به عنوان مثال می‌توان به کادرهای پیامی که در شکل‌های ۵-۱۱ و ۵-۱۲ نمایش داده شده‌اند، اشاره کرد.



شکل ۵-۱۲



شکل ۵-۱۱

ویژوال بیسیک توانایی ایجاد انواع مختلفی از کادرهای پیام را برای شما فراهم می‌کند. برای نمایش کادرهای پیام در ویژوال بیسیک از تابع MsgBox استفاده می‌شود. نحوه استفاده از این تابع به یکی از این روش‌ها امکان‌پذیر است:

(عنوان, نوع و تعداد دکمه‌ها و آیکن کادر پیام, پیام‌نمایشی) MsgBox = نام متغیر از نوع صحیح

عنوان, نوع و تعداد دکمه‌ها و آیکن کادر پیغام, پیام‌نمایشی MsgBox

پیام‌نمایشی و عنوان می‌توانند یک عبارت یا متغیر رشته‌ای باشند که به ترتیب، پیام‌نمایشی داخل کادر پیام و عنوان کادر پیام را تعیین می‌کنند.
 بخش دوم در تابع MsgBox می‌تواند نوع و تعداد دکمه‌ها، نوع آیکن کادر پیام و دکمه پیش فرض را در کادر پیام تعیین کند. برای این کار می‌توانید مقادیر عددی یا رشته‌ای ارائه شده در جداول ۱۵-۵ تا ۱۷-۵ را با استفاده از کاراکتر + با هم ترکیب کنید.

جدول ۱۵-۵ مقادیر مربوط به نوع و تعداد دکمه‌ها در کادر پیام

ثابت رشته‌ای	ثابت عددی	توضیح
vbOKOnly	۰	کادر پیام با دکمه OK
vbOKCancel	۱	کادر پیام با دکمه OK و Cancel
vbAbortRetryIgnore	۲	کادر پیام با دکمه Abort، Retry، Ignore
vbYesNoCancel	۳	کادر پیام با دکمه Yes، No، Cancel
vbYesNo	۴	کادر پیام با دکمه Yes و No
vbRetryCancel	۵	کادر پیام با دکمه Retry و Cancel

جدول ۱۶-۵ مقادیر مربوط به نوع آیکن در کادر پیام

ثابت رشته‌ای	ثابت عددی	توضیح
vbCritical	۱۶	آیکن پیام خطای بحرانی 
vbQuestion	۳۲	آیکن پرسش 
vbExclamation	۴۸	آیکن پیام هشدار 
vbInformation	۶۴	آیکن پیام اطلاعات 

جدول ۱۷-۵ مقادیر مربوط به دکمه پیش فرض

ثابت رشته‌ای	ثابت عددی	توضیح
vbDefaultButton1	صفر	اولین دکمه، دکمه پیش فرض است.
vbDefaultButton2	۲۵۶	دومین دکمه، دکمه پیش فرض است.
vbDefaultButton3	۵۱۲	سومین دکمه، دکمه پیش فرض است.

نکته دکمه پیش فرض دکمه‌ای است که در زمان نمایش کادر پیام، فوکوس را در اختیار می‌گیرد.



وقتی که کاربر روی دکمه‌ای در کادر پیام کلیک کند، یکی از مقادیر ارائه شده در جدول ۱۸-۵ بازگشت داده می‌شود. با ذخیره‌سازی این مقدار در یک متغیر از نوع Integer می‌توان با توجه به پاسخ کاربر اجرای دستورالعمل‌ها را کنترل و مدیریت کرد.

جدول ۱۸-۵ مقادیر مربوط به دکمه‌ای که کلیک می‌شود.

ثابت رشته‌ای	ثابت عددی	توضیح
vbOK	۱	اگر روی دکمه OK کلیک شود.
vbCancel	۲	اگر روی دکمه Cancel کلیک شود.
vbAbort	۳	اگر روی دکمه Abort کلیک شود.
vbRetry	۴	اگر روی دکمه Retry کلیک شود.
vbIgnore	۵	اگر روی دکمه Ignore کلیک شود.
vbYes	۶	اگر روی دکمه Yes کلیک شود.
vbNo	۷	اگر روی دکمه No کلیک شود.

به عنوان مثال به دستور زیر توجه کنید:

Intanswer=MsgBox("Data Entry Is Invalid!" ,

vbRetryCancel+vbCritical+vbDefaultButton2 , "Error")



شکل ۱۳-۵

در صورت اجرای این دستور کادر پیامی مطابق شکل ۱۳-۵ نمایش داده می‌شود. همان‌طور که ملاحظه می‌کنید در بخش دوم با استفاده از کاراکتر + مقادیر مربوط به نوع آیکن، تعداد و نوع دکمه‌ها و دکمه پیش فرض تعیین شده‌اند.

اگر کاربر روی دکمه Retry کلیک کند مقدار VbRetry (معادل عدد ۴) در متغیر Intanswer ذخیره می‌شود و اگر روی دکمه Cancel کلیک کند مقدار VbCancel (معادل عدد ۲) در متغیر Intanswer ذخیره می‌شود.

نکته اگر نوع و تعداد دکمه‌ها تعیین نشده باشند کادر پیام با دکمه OK نمایش داده می‌شود. در صورت عدم انتخاب آیکن، هیچ آیکنی نمایش داده نخواهد شد. به علاوه اگر دکمه پیش فرض انتخاب نشده باشد، اولین دکمه در کادر پیام به عنوان دکمه پیش فرض انتخاب خواهد شد.



به عنوان مثال این دستور یک کادر محاوره مطابق شکل ۱۴-۵ ایجاد می‌کند.

Intanswer=MsgBox("Password Is Not Correct!?", vbOKCancel)



شکل ۱۴-۵

مثال ۶: می‌خواهیم پروژه Image را به شکلی تنظیم کنیم تا هنگام خروج از برنامه پیام تأییدی به کاربر نمایش داده شود.



- ۱ - پروژه Image را باز کنید و به پنجره طراحی فرم بروید.
- ۲ - روی دکمه فرمان Exit دوبار کلیک کنید تا به رویداد Click دکمه فرمان Exit بروید.
- ۳ - دستورات رویداد Click دکمه فرمان Exit را به این صورت تغییر دهید.

```
Private Sub cmdexit_Click()
```

```
Dim intanswer As Integerintanswer=MsgBox("Do You Want To Exit?",
```

```
VbYesNo + vbQuestion,"Exit")
```

```
If intanswer =vbYes Then UnLoad me
```

```
End Sub
```

در این رویداد با استفاده از تابع MsgBox کادر پیامی طراحی شده است که شامل عبارت Do You Want To Exit? و عنوان Exit به همراه دو دکمه Yes و No می‌باشد. در صورت اجرای برنامه و کلیک روی دکمه Exit این رویداد اجرا می‌شود و با نمایش کادر پیام سبب

توقف اجرای برنامه می‌شود و تا زمانی که کاربر پاسخی به کادر پیام ندهد اجرای برنامه متوقف خواهد ماند؛ اما در صورتی که کاربر روی یکی از دکمه‌های کادر پیام کلیک کند تابع MsgBox مقداری مناسب با دکمه مربوطه در متغیر intanswer ذخیره می‌کند. اگر کاربر روی دکمه Yes کلیک کند، با بسته شدن فرم مقدار vbYes در متغیر intanswer ذخیره می‌شود و با استفاده از یک دستور If و بررسی مقدار intanswer اجرای برنامه خاتمه می‌یابد، اما در صورتی که کاربر روی دکمه No یا دکمه Close کلیک کند، مقدار vbNo در متغیر intanswer ذخیره می‌شود و برنامه بدون هیچ گونه تغییری در مقدار زمان ادامه می‌یابد.

۴ - تغییرات را ذخیره کرده و پروژه را اجرا نمایید. روی دکمه Exit کلیک کنید همان‌طور که ملاحظه می‌کنید کادر پیامی مطابق شکل ۱۵-۵ نمایش داده می‌شود.



شکل ۱۵-۵

۵ - روی دکمه No کلیک کنید تا به برنامه بازگردید.

۶ - روی دکمه Exit کلیک کنید. این بار روی

دکمه Yes در کادر پیام کلیک کنید و تفاوت را با حالت قبل بررسی کنید.

۷ - اجرای برنامه را متوقف کنید و به محیط

ویژوال بیسیک بازگردید.

۵-۷ نحوه استفاده از کادرهای ورود داده در ویژوال بیسیک

تاکنون برای ورود داده‌های خود به برنامه از کنترل کادر متن استفاده کرده‌اید. در ویژوال بیسیک روش راحت‌تری برای انجام این کار وجود دارد و آن استفاده از کادر ورود داده یا InputBox است. کادر ورود داده در واقع شبیه به یک کادر پیام است، با این تفاوت که دارای یک کادر متن برای ورود داده و دو دکمه OK و Cancel است (شکل ۱۶-۵). برای ایجاد کادرهای ورود داده در ویژوال بیسیک از تابع InputBox استفاده می‌شود.



شکل ۱۶-۵

این تابع می‌تواند یک کادر ورود داده ایجاد کند و در صورتی که کاربر روی OK کلیک کند داده‌های وارد شده در کادر متن را در یک متغیر رشته‌ای ذخیره می‌کند. اما اگر کاربر روی دکمه Cancel کلیک کند، یک رشته خالی ("") را در متغیر رشته‌ای ذخیره می‌کند. شکل کلی نحوه استفاده از این تابع به یکی از روش‌های زیر است:

(x,y, مقدارپیش‌فرض در کادرمتن, عنوان کادر ورود داده
, پیام نمایشی در کادر ورود داده) = InputBox = نام متغیر رشته‌ای

x, y, عنوان کادر ورود داده, پیام نمایشی در کادر ورود داده InputBox

مقادیر مربوط به پیام نمایشی و عنوان کادر ورود داده از نوع رشته‌ای است و مقدار پیش‌فرض مقداری است که در زمان نمایش کادر ورود داده در کادر متن نمایش داده می‌شود. x و y مقادیر عددی هستند که به ترتیب فاصله کادر ورود داده را از سمت چپ و بالای دسک‌تاپ تعیین می‌کنند.

نکته در صورتی که عنوان کادر ورود داده و کادر پیام تعیین نشود نام فایل پروژه در نوار عنوان آن‌ها نمایش داده می‌شود

مثال ۷: مطابق شکل ۱۷-۵ برنامه‌ای طراحی کنید که کاربر توانایی محاسبه محیط و مساحت هر دایره به شعاع دلخواه را داشته باشد و برای دریافت داده‌ها از کادر ورود داده که به وسیله دکمه Enter Radius فعال می‌شود، استفاده کند.
۱ - یک پروژه با یک فرم، یک کنترل دکمه فرمان و کنترل‌های برچسب مطابق شکل ۱۷-۵ ایجاد کنید.



شکل ۱۷-۵

۲ - رویداد Click کنترل دکمه فرمان را به این صورت تنظیم کنید:

```
Private Sub cmdr_Click()
```

```
Dim str As String
```

```
Dim sngarea As Single, sngper As Single
```

```
str = InputBox("Enter Radius:", "InputData:", 0)
```

```
sngper = 2 * 3.14 * Val(str)
```

```
sngarea = 3.14 * Val(str)^2
```

```
lblper.Caption = sngper
```

```
lblarea.Caption = sngarea
```

```
End Sub
```

۳ - پروژه و فرم را با نام Circle و Circles ذخیره کنید، سپس پروژه را اجرا کنید.

۴ - روی دکمه Enter Radius کلیک کنید تا کادر ورود داده نمایش داده شود.

۵ - عدد ۳ را در کادر متن، کادر ورود داده تایپ کنید و روی دکمه OK در کادر

ورود داده کلیک کنید. در این صورت مقدار تایپ شده در کادر متن در متغیر رشته‌ای str ذخیره می‌شود، سپس با استفاده از تابع Val مقدار متغیر str به عدد تبدیل می‌شود و با استفاده از فرمول‌های مربوطه محیط و مساحت دایره محاسبه می‌شود.

۶ - مجدداً روی دکمه Enter Radius کلیک کنید و عدد ۴ را در کادر متن، کادر ورود

داده تایپ کنید و سپس روی دکمه Cancel کلیک کنید. در این صورت یک رشته خالی در متغیر str ذخیره می‌شود و در تبدیل متغیر str با تابع Val مقدار صفر تولید می‌شود که نتایج محاسبات نیز خواهد بود.

۷ - به اجرای پروژه خاتمه دهید و به پنجره ویژوال بیسیک بازگردید.

۸-۵ کنترل کادر علامت CheckBox

این کنترل امکان ایجاد انتخاب‌های دو حالتی را به وجود می‌آورد به شکلی که در صورت انتخاب کنترل (قرار دادن علامت چک مارک در داخل مربع کنترل) تصمیمات خاصی در برنامه اتخاذ شود و براساس آن پردازش و محاسبات انجام شود یا با برداشتن علامت

چک‌مارک از داخل مربع کنترل کادر علامت از تصمیمات مورد نظر صرف نظر شود. بدین ترتیب کاربر می‌تواند با استفاده از این ویژگی‌ها شرایط خود را به برنامه اطلاع دهد.

جدول ۱۹-۵

خصوصیت	توضیح
Alignment	در صورتی که مقدار این خصوصیت روی 0-Left Justify تنظیم شود، کادر علامت در سمت چپ عنوان کنترل و در صورتی که روی 1-Right Justify تنظیم شود کادر علامت در سمت راست عنوان آن نمایش داده می‌شود.
Style	در صورتی که مقدار این خصوصیت روی 0-Standard تنظیم شود شکل کنترل به صورت یک کادر علامت معمولی و وقتی روی مقدار 1-Graphical تنظیم شود به شکل یک دکمه فرمان دیده می‌شود.
Value	در صورتی که مقدار این خصوصیت برابر 0-Unchecked باشد نشان دهنده این است که کنترل انتخاب نشده است و اگر برابر 1-Checked باشد نشان دهنده این است که کنترل انتخاب شده است و اگر مقدار آن برابر 2-Grayed باشد به معنی این است که کنترل دارای علامت چک مارک بوده و زمینه مربع کادر علامت نیز خاکستری رنگ است.

۹-۵ معرفی مهم‌ترین ویژگی‌های فرم‌ها و کنترل‌های کادر متن، دکمه فرمان و غیره

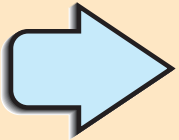
در این واحد کار با اشیایی نظیر فرم و کنترل‌های متعدد آشنا شدید، در پایان لازم است تا شما را با بعضی از خصوصیات و رویدادهای دیگر فرم و کنترل ذکر شده بیشتر آشنا کنیم.

جدول ۲۰-۵ خصوصیات فرم‌ها و کنترل‌ها

خصوصیت	توضیح
Appearance	اگر مقدار این خصوصیت روی Flat- تنظیم شود، کنترل یا فرم به صورت مسطح و اگر روی مقدار ۱-۳D تنظیم شود، کنترل یا فرم به صورت سه بعدی نمایش داده می‌شود.
BackColor	رنگ زمینه را در فرم و بعضی از کنترل‌ها مانند کادر متن تعیین می‌کند.
Enabled	اگر مقدار این خصوصیت روی True تنظیم شود، کنترل یا فرم فعال و قابل استفاده بوده و اگر روی False تنظیم شود، کنترل و فرم غیرفعال و غیر قابل استفاده می‌شود.
ForeColor	رنگ قلم را در فرم و بعضی از کنترل‌ها تعیین می‌کند.
Font	تنظیمات قلم را در فرم و بعضی از کنترل‌ها تعیین می‌کند.
Picture	یک تصویر را روی فرم و بعضی از کنترل‌ها تعیین می‌کند.
MousePointer	اشاره‌گر ماوس را برای فرم و بعضی از کنترل‌ها تعیین می‌کند.
MouseIcon	یک آیکن دلخواه را به عنوان اشاره‌گر برای فرم و بعضی کنترل‌ها تعیین می‌کند.
Visible	اگر مقدار این خصوصیت روی True تنظیم شود کنترل یا فرم نمایش داده شده و اگر مقدار این خصوصیت روی False تنظیم شود کنترل یا فرم نمایش داده نخواهد شد.
TabIndex	این خصوصیت مربوط به کنترل‌ها می‌شود و ترتیب حرکت روی کنترل‌ها را با استفاده از کلید Tab تعیین می‌کند و مقادیر مساوی یا بزرگ‌تر از صفر را کسب می‌کند.
TabStop	این خصوصیت از نوع منطقی بوده و مربوط به کنترل‌ها می‌شود و اگر روی مقدار False تنظیم شود در هنگام استفاده از کلید Tab، کنترل مربوط، فوکوس را دریافت نمی‌کند.
BorderStyle	اگر این خصوصیت در کنترل‌های برجسته و کادر متن روی مقدار None- ۰ تنظیم شود، کنترل بدون حاشیه و کادر دور و اگر روی مقدار 1-Fixed Single تنظیم شود، کنترل با حاشیه و کادر دور مشاهده می‌شود.
MaxButton	اگر مقدار این خصوصیت روی True تنظیم شود، دکمه Maximize در فرم فعال و در صورتی که مقدار آن روی False تنظیم شود، دکمه Maximize غیرفعال می‌شود.
MinButton	اگر مقدار این خصوصیت روی True تنظیم شود، دکمه Minimize در فرم فعال و در صورتی که مقدار آن روی False تنظیم شود دکمه Minimize غیرفعال می‌شود.
Icon	یک آیکن دلخواه را به عنوان آیکن منوی کنترل در فرم تعیین می‌کند.

ادامه جدول ۲۰-۵

خصوصیت	توضیح
Alignment	این خصوصیت در کنترل‌هایی مانند برچسب و کادر متن وجود دارد و تراز عبارت را در کنترل تعیین می‌کند، مقدار 0-Left Justify تراز از چپ، 1-Right Justify تراز از راست و 2-Center تراز از وسط را به وجود می‌آورند.
ToolTipText	این خصوصیت در بیشتر کنترل‌ها وجود دارد و زمانی که اشاره‌گر ماوس روی کنترل نگه داشته شود عبارت موجود در این خصوصیت نمایش داده می‌شود. از این خصوصیت برای راهنمایی کاربران استفاده می‌شود.
WordWrap	این خصوصیت مربوط به کنترل برچسب است. اگر محتویات خصوصیت Caption بیش از اندازه کنترل باشد و بخواهید عبارت موجود در خصوصیت Caption در چند خط نمایش داده شود، مقدار این خصوصیت و سپس مقدار خصوصیت AutoSize را روی True تنظیم کنید.
MultiLine	این خصوصیت مربوط به کنترل کادر متن است و در صورتی که مقدار آن روی True تنظیم شود، می‌توانید اطلاعات خود را در خطوط جداگانه وارد کنید. البته بهتر است که ارتفاع کنترل را نیز طوری تنظیم نمایید تا بتوان چند خط را به طور هم‌زمان مشاهده کرد.
Moveable	اگر مقدار این خصوصیت روی True تنظیم شود امکان جابه‌جا کردن فرم روی دسک‌تاپ وجود دارد، اما اگر مقدار آن روی False تنظیم شود این امکان وجود ندارد.
WindowState	این خصوصیت مربوط به فرم‌ها بوده و سه مقدار مختلف را کسب می‌کند. در صورتی که مقدار آن روی 0-Normal تنظیم شود موقعیت فرم روی دسک‌تاپ به وسیله خصوصیات Top و Left تنظیم می‌شود و اگر مقدار آن روی 1-Minimized تنظیم شود پنجره برنامه در زمان اجرا با اندازه حداقل و در صورتی که مقدار آن روی 2-Maximized تنظیم شود، پنجره برنامه با اندازه حداکثر نمایش داده خواهد شد.



Learn in English

If...Then...Else

Use an If...Then...Else block to define several blocks of statements, one of which will execute

If *condition1* Then

[*statementblock*]

[Else

[*statementblock*]]

End If

Visual Basic first tests *condition1*. If it's False, Visual Basic executes the corresponding statement Else statement block

For example, your application could perform different actions with depending on the value of grade variable:

```
If grade < 10 Then
```

```
    result ="failed"
```

```
Else
```

```
    result ="passed"
```

```
End If
```


واژه‌نامه

Border	کادر، حاشیه
Compare	مقایسه کردن
Condition	شرط
CorresPonding	متناظر، متناسب
Decision	تصمیم
Execute	اجرا کردن
Image	تصویر
Input Box	کادر ورود داده
Logical	منطقی
Message Box	کادر پیام
Operation	عملگر
Perform	انجام دادن
Structure	ساختار
Style	حالت

خلاصه مطالب

• برای کنترل روند اجرای برنامه در شرایط مختلف از دستورات شرطی مانند If استفاده می‌شود.

شکل کلی دستور If به این صورت است:

```
If (شرط(ها)) Then
    .....
    دستورات
Else
    .....
    دستورات
End If
```

• برای اضافه کردن یک فرم جدید به پروژه می‌توانید از گزینه Add Form در منوی Project، کلیک راست در پنجره پروژه و انتخاب گزینه Add Form (یا از دکمه Add Form در نوار ابزار استاندارد) استفاده کنید.

- برای مقایسه مقادیر رشته‌ای، عددی و سایر انواع داده‌ها از عملگرهای مقایسه‌ای استفاده می‌شود.
- متد Show می‌تواند یک فرم را در حافظه بارگذاری کرده و نمایش دهد.
- متد Hide می‌تواند یک فرم را بدون آن که از حافظه خارج نماید، مخفی کند.
- برای ایجاد ترکیب چند شرط با یکدیگر از عملگرهای منطقی And، Or و Not استفاده می‌شود.
- با استفاده از دستور Unload می‌توان یک فرم را از حافظه خارج کرد.
- ترکیب شرطی که با استفاده از عملگر And ایجاد می‌شود، وقتی نتیجه درست (True) خواهد داشت که تمام مقایسه‌ها به طور هم‌زمان درست باشند و در سایر حالت‌ها نتیجه نادرست خواهد داشت.
- ترکیب شرطی که با استفاده از عملگر Or ایجاد می‌شود، وقتی نتیجه نادرست (False) خواهد داشت که تمام مقایسه‌ها نادرست باشند و در سایر حالت‌ها نتیجه درست خواهد داشت.
- از کنترل کادر تصویر و کنترل تصویر می‌توان برای نمایش فایل‌های گرافیکی یا تصویر استفاده کرد.
- عملگر Not نتیجه هر عبارت مقایسه‌ای را که درست باشد به نادرست و اگر نادرست باشد به درست تبدیل می‌کند.
- از کادرهای پیام، برای نمایش پیام‌های مناسب در مواقع نیاز و ایجاد حالت‌های مختلف انتخاب برای کاربر استفاده می‌شود.
- از کادرهای ورود داده برای دریافت داده‌ها از کاربر استفاده می‌شود.
- کادرهای پیام با استفاده از تابع MsgBox و کادرهای ورود داده با استفاده از تابع InputBox ایجاد می‌شوند.
- تابع RTrim کاراکترهای فاصله در انتهای یک عبارت متنی (رشته‌ای) را حذف می‌کند.
- تابع LTrim کاراکترهای فاصله در ابتدای یک عبارت متنی (رشته‌ای) را حذف می‌کند.
- تابع Trim کاراکترهای فاصله در ابتدا و انتهای یک عبارت متنی (رشته‌ای) را حذف می‌کند.
- با استفاده از تابع Load Picture می‌توان یک فایل گرافیکی یا تصویر را در کنترل کادر تصویر یا کنترل تصویر نمایش داد.

آزمون نظری

۱ - کدام خصوصیت در کادر متن حداکثر تعداد کاراکترهای ورودی را تعیین می‌کند؟

الف - BorderStyle ب - PasswordChar ج - MaxLength د - TabIndex

۲ - کدام عملگر منطقی از اولویت پایین‌تری نسبت به سایر عملگرهای منطقی

برخوردار است؟

الف - And ب - Or ج - Not د - And و Or

۳ - کدام تابع برای ایجاد یک کادر ورود داده مناسب است؟

الف - MsgBox ب - InputBox ج - InBox د - msg

۴ - در صورتی که نوع و تعداد دکمه‌ها در کادر پیام تعیین نشود، به‌طور پیش‌فرض از

چه دکمه‌هایی استفاده می‌شود؟

الف - OK ب - OK و Cancel ج - Yes و No د - Cancel

۵ - کدام گزینه در رابطه با خصوصیت TabIndex درست است؟

الف - ترتیب نمایش کنترل‌ها را روی فرم تعیین می‌کند.

ب - ترتیب حرکت بین کنترل‌ها را روی یک فرم معین می‌کند.

ج - ترتیب بارگذاری کنترل‌ها را در حافظه تعیین می‌کند.

د - وضعیت کلید Tab را در صفحه کلید بررسی می‌کند.

۶ - حاصل عبارت $3 < 5 * 3 > 12$ چیست؟

الف - True ب - False ج - ۲ د - ۲

۷ - کدام خصوصیت مشترک بین فرم و کنترل‌ها وجود دارد که توانایی مخفی کردن

یا نمایش آن‌ها را دارد؟

الف - Caption ب - Enabled ج - Visible د - Appearance

۸ - حاصل ترکیب شرطی زیر چیست؟

(با توجه به عدم استفاده از دستور Option Compare Text)

"Cpu" = "Cpu" Or 50 <= -70 And Not True

الف - True

ب - False

ج - استفاده از عملگر Not و And در کنار هم اشتباه است.

د - استفاده از عملگر مقادیر منطقی، عددی و رشته‌ای در یک ترکیب اشتباه است.

۹- در صورت عدم انتخاب عنوان یک کادر پیام به طور پیش فرض از استفاده می شود.
۱۰- الف- نام فرم ب- نام پروژه ج- نام فایل فرم د- نام فایل پروژه
با استفاده از کدام خصوصیت در کادر متن می توان از ویرایش داده ها توسط کاربر جلوگیری کرد؟

الف- Wordwrap ب- Lock ج- Locked د- MultiLine

۱۱- In the If statement while condition is false, Visual Basic executes the statements that exist within the part.

a- then b- else c- if else d- end if

۱۲- عملگرهای مقایسه ای را بیان کنید سپس کاربرد هر یک را توضیح دهید.
۱۳- دستور شرطی If را توضیح دهید و حالت های مختلف نحوه استفاده از آن ها را بیان کنید.

۱۴- اصطلاحات زیر را توضیح دهید:

الف- متد Show

ب- رویداد Unload

ج- متد Hide

د- رویداد Change

۱۵- تفاوت بین کادر پیغام و کادر ورود داده را بیان کنید.
۱۶- عملگرهای منطقی را نام ببرید و کاربرد هر یک را بیان کنید.

آزمون عملی

- ۱ - حاصل عبارت‌های زیر به ازای مقادیر $a = 15$ ، $b = 20$ و $c = 100$ چه خواهد بود؟
- الف - $a > 40$ AND $b >= 10$ OR $c < 0$
- ب - $b <= 10$ OR $c <> 100$
- ج - $a <= 35$ OR $c > 100$ AND $b >= 20$

۲ - پروژه‌ای طراحی کنید که مطابق شکل ۱ نام، نام خانوادگی و نمره ۴ درس یک دانش‌آموز را دریافت کند، سپس معدل وی را محاسبه نماید. در صورتی که معدل وی کوچک‌تر از ۱۲ باشد یک کادر پیام مطابق شکل ۲ و در صورتی که معدل نمره وی بزرگ‌تر یا مساوی ۱۲ باشد کادر پیامی مطابق شکل ۳ نمایش داده شود، به علاوه برای دریافت داده از کادر ورود داده استفاده شود. لازم به ذکر است که کلیک روی دکمه‌های فرمان، کادر ورود داده‌ای را در اختیار کاربر قرار می‌دهد.

The image shows a window titled 'Student' with a close button in the top right corner. Inside the window, there are six input fields arranged in two columns. The left column contains 'Student Name', 'Student Family', and 'Mathematics'. The right column contains 'Physics', 'Chemistry', and 'English'. Below these fields is a single input field labeled 'Average'.

شکل ۱



شکل ۳



شکل ۲

