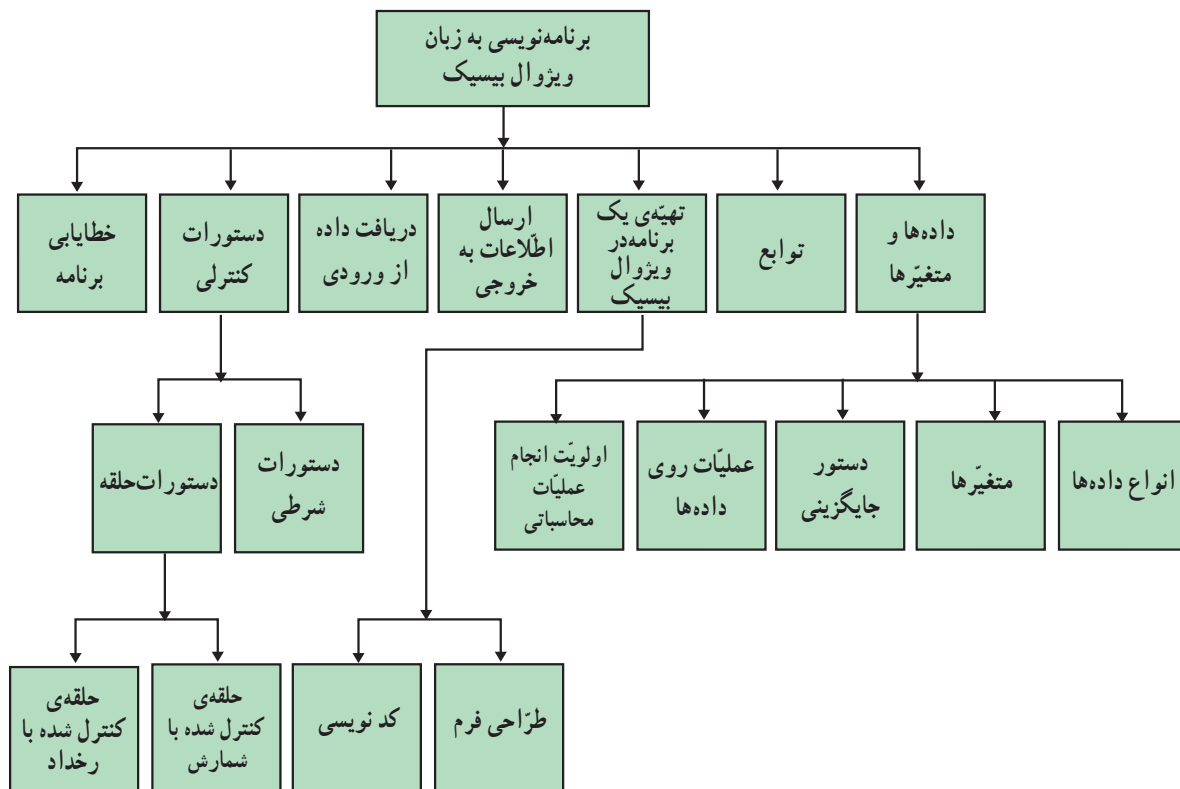


برنامه‌نویسی به زبان ویژوال بیسیک^۱

در بخش قبل با فرایند برنامه‌نویسی آشنا شدیم و دیدیم که برای حل مسئله باید روشی برای حل آن طراحی کنیم. پس از طراحی الگوریتم مناسب برای اجرای الگوریتم مورد نظر از رایانه استفاده می‌کنیم؛ یعنی باید آن الگوریتم را به یک زبان برنامه‌نویسی تبدیل کرده و با استفاده از رایانه به اجرای آن بپردازیم.

BASIC^۲ یکی از زبان‌های برنامه‌نویسی است که با استفاده از آن می‌توان به سادگی، الگوریتم را به دستورالعمل‌های قابل فهم برای ماشین تبدیل کرد و سپس به اجرا و آزمایش برنامه پرداخت؛ همان‌طور که یک زبان طبیعی، دارای لهجه‌های متفاوتی است، زبان‌های برنامه‌نویسی هم نسخه‌های متفاوتی دارند. زبان بیسیک نیز دارای نسخه‌های متعددی است که در این کتاب به معرفی نسخه‌ی ویژوال بیسیک می‌پردازیم. زبان برنامه‌نویسی ویژوال بیسیک این امکان را فراهم می‌کند که برنامه‌نویس از ابزارهای گرافیکی برای تهیه‌ی برنامه استفاده کند.

بخش دوازدهم در یک نگاه



1. Visual Basic (VB)

2. Beginner's All-Purpose Symbolic Instruction Code (به معنای زبان نمادی همه‌منظوره برای مبتدیان)

برای برنامه‌نویسی با یک زبان، باید قابلیت‌ها و امکانات آن را دانست. همان‌گونه که در زبان‌های طبیعی برای ساختن جملات معنادار، نیازی به دانستن تمام لغات زبان نیست؛ برای نوشتن یک برنامه با یک زبان برنامه‌نویسی، نیازی به دانستن تمام قواعد و دستورات آن نیست. برخی از قابلیت‌ها و دستورات کلی زبان برنامه‌نویسی عبارت‌اند از:

۱- نگهداری و آدرس‌دهی انواع داده‌ها

۲- عملیات روی داده‌ها

۳- توابع

۴- دستورات ورودی و خروجی

۵- دستورات کنترلی

علاوه بر این، ویژگی‌های بیسیک دارای رابط گرافیکی است. برنامه‌نویس در ویژگی‌های بیسیک تصمیم می‌گیرد که از عناصر گرافیکی به گونه‌ای استفاده کند تا کاربر بتواند به راحتی با برنامه ارتباط برقرار کند؛ سپس برای هر یک از این عناصر، کد مربوط به آن را می‌نویسد.

داده‌ها و متغیرها

انواع داده‌ها: یکی از قابلیت‌های مهمی که در هر زبان برنامه‌نویسی باید مورد بررسی قرار گیرد، انواع داده‌های قابل تعریف در آن است. برخی از انواع داده‌ها در ویژگی‌های بیسیک عبارت‌اند از: داده‌های عددی، رشته‌ای، تاریخ و بولین در جدول زیر، با نمونه‌هایی از انواع داده‌ها آشنا می‌شویم:

نوع داده	توضیح	مثال
عددی صحیح (Integer, Long)	داده‌ی عددی صحیح که می‌توان بر روی آن محاسبات ریاضی و منطقی انجام داد.	۲۰۵ و ۴- و ۴۰۰۶۸۲۶
عددی اعشاری (Single,...)	داده‌ی عددی اعشاری که می‌توان بر روی آن محاسبات ریاضی و منطقی انجام داد.	۳/۱۴ و ۰/۴۳۲۱
رشته‌ای (String)	ترکیبی از حروف، علائم یا ارقام است که بین دو علامت ("") قرار می‌گیرد.	"IRAN" و "۰۲۱_۸۸۳۰۱۴۸۱"
تاریخ (Date)	داده‌ای که نشان‌دهنده‌ی تاریخ (سال/روز/ماه) یا زمان (ثانیه: دقیقه: ساعت) است. این داده بین دو علامت "#" قرار می‌گیرد.	#10/23/2007 و #11:47:22
بولین (Boolean)	داده‌ای که می‌تواند دارای ارزش «درست» یا «نادرست» باشد.	False و True

متغیرها: برای استفاده از انواع داده‌ها، باید آن‌ها را به یک متغیر نسبت دهیم.

۱. داده‌ی عددی Integer، اعداد صحیح بین ۳۲۷۶۸- تا ۳۲۷۶۷ را می‌پذیرد و برای اعداد صحیح خارج از این محدوده، از نوع Long استفاده می‌شود.

متغیر، نامی برای محلی از حافظه‌ی رایانه است که داده در آن قرار می‌گیرد.

مقدار متغیر می‌تواند در طول برنامه تغییر کند. برای نام‌گذاری متغیرها، می‌توان از ترکیب حروف، اعداد و علامت «_» (underline) استفاده کرد.

مثال: mark و s - Average و class 2 و m

توجه داشته باشید که در نام‌گذاری متغیرها، موارد زیر مجاز نیستند:

– استفاده از «فاصله» (space)

– استفاده از عدد و علامت «_» در اول نام متغیر

– استفاده از کلمات رزرو شده^۱ زبان ویژوال بیسیک شامل نام دستورات، نام توابع یا نوع داده‌ها (مانند print و val و string

و...)

بهرتر است متغیرها در ابتدای برنامه تعریف شوند. برای تعریف متغیرها از دستور Dim^۲ استفاده می‌کنیم.

شکل کلی این دستور چنین است: **نوع متغیر As نام متغیر Dim**

نوع متغیر باید مناسب با نوع داده‌ای که در آن قرار می‌گیرد، تعیین شود.

اگر متغیری در برنامه تعریف نشده باشد، ویژوال بیسیک به‌طور پیش‌فرض، نوع آن را Variant در نظر می‌گیرد. به این ترتیب،

نوع متغیر، براساس مقدار داده‌ای که در آن جایگزین می‌شود، تعیین می‌شود.

دستور جایگزینی: برای مقدار دهی به یک متغیر، از دستور جایگزینی استفاده می‌کنیم.

شکل کلی این دستور چنین است: **(یا یک عبارت محاسباتی) مقدار داده = نام متغیر**

با اجرای این دستور، مقدار داده یا حاصل عبارت سمت راست تساوی، در متغیر سمت چپ جایگزین می‌شود.

به مثال‌های جدول زیر توجه کنید:

تعریف متغیر	دستور جایگزینی
Dim Country As String	Country = "Iran"
Dim Pi As Single	Pi = 3.14
Dim BD As Date	BD = # 05/23/2006#
Dim K As Boolean	K = True
Dim Num As Integer	Num = Num+ 350
Dim Count As Long	Count = 4653271

۱. Reserved word: کلماتی که در دستور زبان ویژوال بیسیک تعریف شده هستند.

عملیات روی داده‌ها: مقدار یک متغیر ممکن است نتیجه‌ی یک عبارت محاسباتی و یا سایر عملیات روی داده‌ها باشد. برای انجام عملیات مختلف بر روی داده‌ها از عملگرها استفاده می‌شود. نمونه‌هایی از عملگرهای محاسباتی و رشته‌ای را که در ویزوال بیسیک مورد استفاده قرار می‌گیرند؛ در جدول زیر مشاهده می‌کنید:

نتیجه	مثال	نام عملگر	عملگر	نوع عملگر
9	3^2	توان	^	محاسباتی
20	$4*5$	ضرب	*	
2.5	$10/4$	تقسیم	/	
7	$2+5$	جمع	+	
4	$9-5$	منها	-	
3	$23 \text{ Mod } 5$	باقی مانده‌ی تقسیم صحیح	Mod	
2	$10 \setminus 4$	تقسیم صحیح	\	
good	"**"+"good" & "**"	ترکیبی	& و +	رشته‌ای

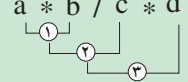
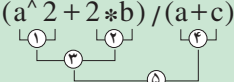
از آن‌جا که عملگر «+» به منظور جمع داده‌های عددی نیز به کار می‌رود، برای جلوگیری از خطا بهتر است از عملگر «&» برای ترکیب رشته‌ها استفاده شود.

اولویت انجام عملیات محاسباتی: در برنامه‌نویسی، اعمال ریاضی از چپ به راست با اولویت زیر انجام می‌شوند:

- عبارات داخل پرانتزها
- توان‌رسانی
- ضرب و تقسیم
- تقسیم صحیح (ا)
- باقی مانده‌ی تقسیم صحیح (Mod)
- جمع و تفریق

در جدول زیر، دو نمونه عبارت ریاضی و معادل آن در ویزوال بیسیک و هم‌چنین تعیین اولویت انجام عملیات را مشاهده

می‌کنید:

عبارت ریاضی	عبارت ویزوال بیسیک	اولویت انجام عملیات
$\frac{ab}{c} \times d$	$a*b/c*d$	$a * b / c * d$ 
$\frac{a^2 + 2b}{a + c}$	$(a^2+2*b)/(a+c)$	$(a^2 + 2*b) / (a+c)$ 

توابع

توابع^۱، برنامه‌های از پیش نوشته شده‌ای هستند که برنامه‌نویس می‌تواند از آن‌ها در برنامه‌ی خود استفاده کند. یک تابع ممکن است یک یا چند ورودی (آرگومان^۲) را دریافت کند. نتیجه‌ی تابع را می‌توان در یک متغیر جایگزین کرد. همواره علامت « () » در مقابل نام تابع آورده می‌شود، که در آن آرگومان قرار می‌گیرد. در صورتی که تابع، چند آرگومان داشته باشد، آرگومان‌ها با علامت « , » از هم جدا می‌شوند. گاهی ممکن است تابع هیچ آرگومانی نداشته باشد. در جدول زیر با برخی از توابع آشنا می‌شوید :

شکل تابع	عملکرد تابع	مثال	نتیجه‌ی اجرا
Abs(x)	قدرمطلق x را برمی‌گرداند.	Abs(-5)	5
Sqr(x)	جذر x را برمی‌گرداند.	Sqr(16)	4
Int(x)	جزء صحیح x را برمی‌گرداند.	Int(-7.2)	-8
Round(x,n)	مقدار عددی x را با n رقم اعشار گرد می‌کند.	Round(43.86,1)	43.9
Len(s)	طول رشته‌ی s را برمی‌گرداند.	Len("Ali")	3
Val(s)	رشته‌ی عددی s را به مقدار عددی تبدیل می‌کند.	Val("5.2")	5.2
Str(n)	مقدار عددی n را به رشته‌ی عددی تبدیل می‌کند.	Str(123)	"123"

تهیه‌ی یک برنامه در ویژوال بیسیک

هر برنامه در ویژوال بیسیک یک Project نامیده می‌شود که شامل مجموعه‌ای از برنامه‌های کوچک‌تر (زیر برنامه) با کدهای مستقل از هم است. هر کدام از زیربرنامه‌ها برای اجرای یک رویداد نوشته می‌شوند.

هر عملی که کاربر برای برقراری ارتباط با برنامه انجام می‌دهد، یک رویداد^۳ نامیده می‌شود.

برای مثال، کلیک کردن روی هر یک از عناصر گرافیکی صفحه یا قرار دادن نشانگر ماوس روی آن‌ها هر کدام یک رویدادند.

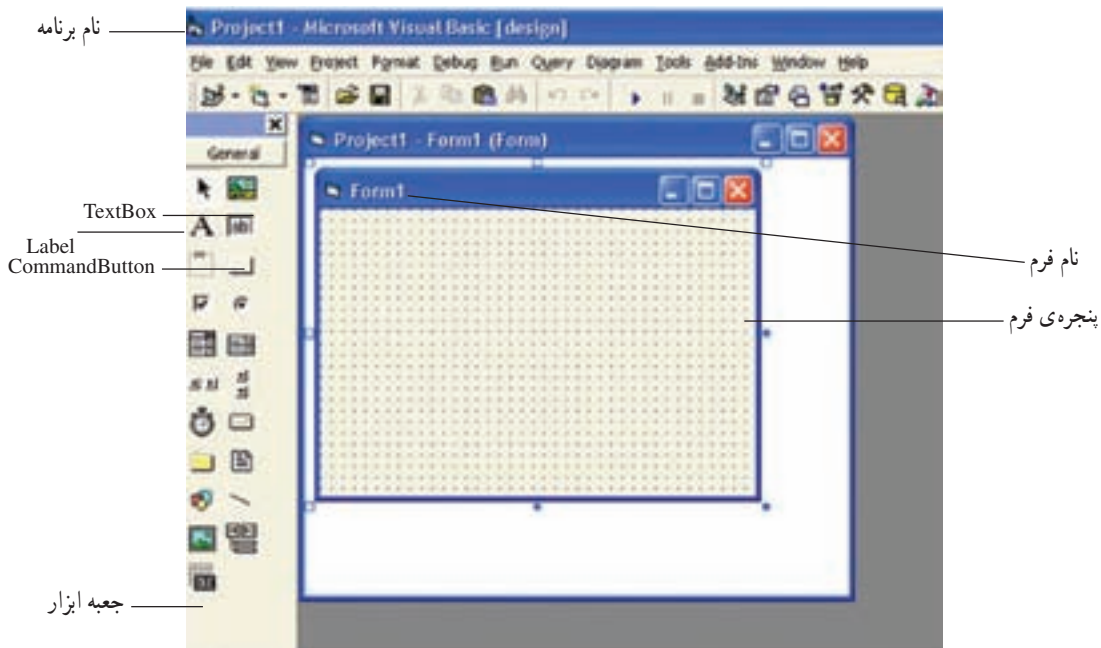
طراحی فرم: اولین قدم در تهیه‌ی برنامه در ویژوال بیسیک، طراحی فرم است. در طراحی فرم، برنامه‌نویس تعیین می‌کند که برای دریافت داده‌ها و خروج اطلاعات کدام عناصر گرافیکی مورد استفاده قرار می‌گیرند. ویژوال بیسیک دارای یک پنجره برای طراحی فرم است که می‌توان ابزارها و عناصر گرافیکی مورد نیاز را روی آن قرار داد. چند نمونه از این عناصر عبارت‌اند از : دکمه‌ی فرمان (CommandButton)، برجسب (Label) و جعبه‌ی متن (TextBox) که در جعبه ابزار (Toolbox) وجود دارند.

1. Functions

۲. آرگومان می‌تواند یک مقدار، یک متغیر یا یک عبارت باشد.

3. Event

در شکل ۱۲-۱ تصویر یک فرم و جعبه ابزار را مشاهده می‌کنید.



شکل ۱۲-۱- پنجره فرم و جعبه ابزار



شکل ۱۲-۲- کادر خصوصیات

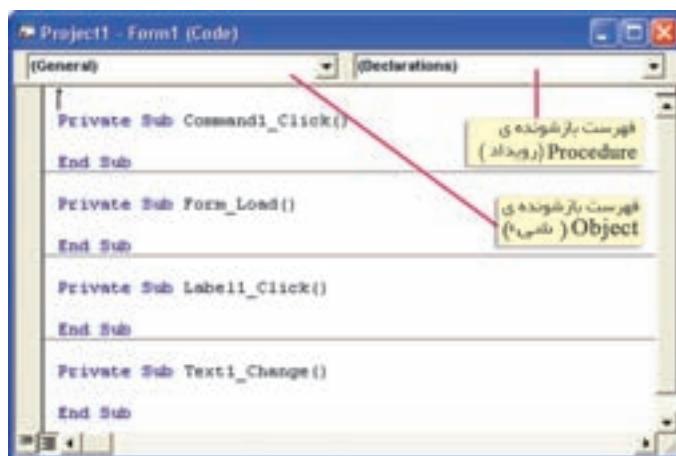
هر شیء^۱ (فرم یا عنصر گرافیکی موجود در آن) دارای مشخصه‌هایی است که در پنجره «مشخصات» (Properties) قابل تنظیم اند. (شکل ۱۲-۲)

چند نمونه از این مشخصه‌ها عبارت‌اند از:

- نام (Name)
- عنوان (Caption)
- متن (Text)
- قلم (Font)
- رنگ (Color)

برخی از این مشخصه‌ها برای عناصر مختلف مشترک‌اند و برخی دیگر مخصوص یک عنصر خاص‌اند. به‌طور کلی، هر شیء دارای مشخصه‌های پیش‌فرض است و برنامه‌نویس می‌تواند آن‌ها را به دلخواه تغییر دهد.

کدنویسی: پس از طراحی فرم، لازم است برنامه‌نویس برای ارتباط عناصر با یکدیگر، زیربرنامه‌ی مربوط به رویدادهای پیش‌بینی شده را در پنجره‌ی کدنویسی بنویسد. (شکل ۳-۱۲)



شکل ۳-۱۲ - پنجره‌ی کدنویسی

شکل کلی یک زیربرنامه در پنجره‌ی کدنویسی چنین است^۱:

```
Private Sub ( نام زیربرنامه )
    دستورالعمل‌های برنامه
End Sub
```

تغییر برخی از مشخصه‌های شیء را می‌توان در پنجره‌ی کدنویسی انجام داد^۲.

شکل کلی این دستور چنین است: مقدار = مشخصه. نام شیء

مثال: ویژگیال بیسیک، اولین دکمه‌ی فرمان را با عنوان Command1 و دکمه‌های فرمان بعدی را به ترتیب Command2، Command3 و ... در نظر می‌گیرد. برای تغییر عنوان یک دکمه‌ی فرمان به "Run" چنین می‌نویسیم:

```
Command1.Caption="Run"
```

در ویژگیال بیسیک، نام‌گذاری زیربرنامه‌ها چنین است: (رویداد - نام شیء)
برای یک فرم می‌توان رویدادهای مختلفی تعریف کرد. برای هریک از عناصر نیز رویدادهای متفاوتی وجود دارد.

مثال: (Form _ Load) به این معناست که هنگام ظاهر شدن فرم در زمان اجرا، دستورات این زیربرنامه اجرا شوند، و (Command1 _ Click) به این معناست که با کلیک کردن روی دکمه‌ی فرمان، دستورات مربوط به زیربرنامه‌ی آن اجرا شوند.

۱. زیر برنامه‌ها را به شکل‌های دیگری می‌توان نوشت که مورد بحث این کتاب نیست.
۲. برخی از مشخصه‌ها مانند Name، فقط در پنجره‌ی مشخصات، قابل تغییرند.

ارسال اطلاعات به خروجی

دستور Print

برای نمایش یک عبارت در خروجی، از دستور Print استفاده می‌کنیم.

شکل کلی این دستور چنین است^۱:

با اجرای این دستور، عبارت داخل گیومه در پنجره‌ی فرم ظاهر می‌شود. اگر بخواهیم خروجی را چاپ کنیم، از دستور زیر استفاده می‌کنیم:

"عبارت" Printer.Print

در جدول زیر، شکل‌های مختلف استفاده از دستور Print و نتیجه‌ی اجرای آن را مشاهده می‌کنید.

عبارات PRINT	نتیجه بر روی صفحه نمایش	توضیحات
Print		یک خط فاصله ایجاد می‌کند.
Print "IRAN"	IRAN	رشته‌ی داخل گیومه را نمایش می‌دهد.
Print "15+5"	15+5	محتوای داخل گیومه را عیناً نمایش می‌دهد.
Print 15+5	20	مجموع دو عدد را به دست می‌آورد و سپس نتیجه را نمایش می‌دهد.
Print "Tehran"+" IRAN"	Tehran IRAN	دو رشته را به هم وصل می‌کند و سپس نمایش می‌دهد.
Print "Hi" ; "Ali"	Hi Ali	عبارات داخل گیومه بدون فاصله نمایش داده می‌شوند.
Print "Hi" , "Ali"	Hi Ali	عبارات داخل گیومه با فاصله نمایش داده می‌شوند.
Print "name" ; Tab(20); "Family"	name Family	عبارت «name» از ستون اول و عبارت «Family» از ستون ۲۰ نمایش داده می‌شود.

تابع **Tab (n)**: با استفاده از این تابع، در دستور Print می‌توان خروجی را به ستون n^{ام} منتقل کرد.

عنصر Label

از این عنصر برای نمایش یک متن روی فرم استفاده می‌شود. این متن توسط کاربر قابل ویرایش نیست. کاربرد این عنصر برای نمایش یک پیغام یا معرفی سایر عناصر روی فرم است.

مثال: برای نمایش پیغام "Welcome to visual Basic" روی فرم، کد آن را چنین می‌نویسیم:

```
Label1.Caption="Welcome to visual Basic"
```

یا

```
Label1="Welcome to visual Basic"
```

۱. در صورتی که در برنامه، بیش از یک فرم استفاده شده باشد، شکل دستور چنین خواهد بود: "عبارت" Form1.Print

دریافت داده از ورودی

تابع () Input Box

در بعضی موارد لازم است برنامه، سؤالی را برای کاربر مطرح کند و پاسخ آن را دریافت کند. تابع () Input Box این امکان را فراهم می‌سازد. با اجرای این تابع یک کادر محاوره‌ای ظاهر می‌شود و منتظر دریافت داده از کاربر و کلیک روی یک دکمه (OK یا Cancel) می‌شود.

شکل ساده‌ی این تابع چنین است: ("عنوان کادر"، "عبارت پیغام") = InputBox نام متغیر

(در این تابع، آرگومان اول اجباری است.)

مثال: F- Name=Input Box("Enter your first name:","Get name")

نتیجه‌ی اجرای این دستور، به شکل زیر است: (شکل ۴-۱۲)



شکل ۴-۱۲

عنصر Text box

از این عنصر برای ورود داده‌های رشته‌ای استفاده می‌شود.

مثال: با اجرای دستور زیر کاربر می‌تواند نشانی مورد نظر را در کادر Text box وارد کند.

Address=Text1.Text

یا

Address=Text1

لازم به ذکر است که برای نمایش عبارات نیز می‌توان از این عنصر استفاده کرد.

مثال: با اجرای دستور زیر، کاربر نشانی مورد نظر را در کادر Text box مشاهده می‌کند.

Text2.Text="sqr.Vali Asr, No.45"

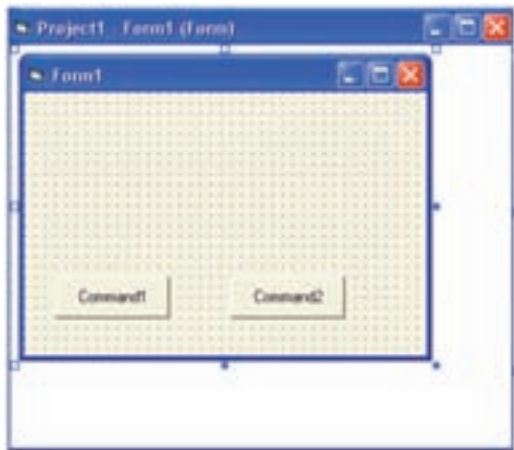
یا

Text2="Sqr.vali Asr, No.45"

این عنصر، امکان دریافت تا ۲۰۴۸ کاراکتر را به‌طور متوالی (به‌صورت پیش‌فرض) دارد. در صورتی که مشخصه‌ی Multiline

این عنصر را "True" کنیم، می‌توانیم متن را در چند سطر وارد کرده یا مشاهده کنیم.

در برنامه‌های زیر، نحوه‌ی استفاده از دستورات ورودی و خروجی را مشاهده می‌کنید :
 مثال ۱: برنامه‌ای بنویسید که سال تولد کاربر را به‌عنوان ورودی بگیرد و سن او را نمایش دهد.
 همان‌طور که قبلاً گفتیم، قدم اول در تهیه‌ی یک برنامه، طراحی فرم است؛ در این مثال، دو دکمه‌ی فرمان از جعبه ابزار را در پنجره‌ی فرم قرار می‌دهیم. (شکل ۱۲-۵)



شکل ۱۲-۵

در پنجره‌ی کدنویسی کد مربوط به این فرم و عناصر روی آن را می‌نویسیم :
 عنوان‌های "Run" (برای اجرای برنامه و مشاهده‌ی خروجی) و "Exit" (برای خروج از برنامه) را برای دکمه‌های فرمان تعیین می‌کنیم.

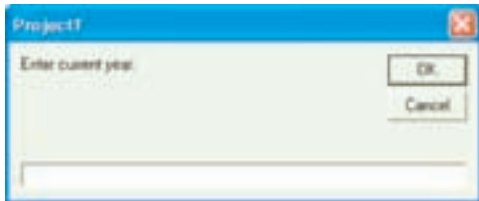
```
Private Sub Form_Load()  
Form1.Caption="Compute your age"  
Command1.Caption="RUN"  
Command2.Caption="EXIT"  
End Sub
```

کد مربوط به تعیین عملکرد دکمه‌ی Run را چنین می‌نویسیم :

```
Private Sub Command1_Click()  
Dim Year1 As Integer, Year2 As Integer, Age As Integer  
Year1=InputBox("Enter year of Birthday:")  
Year2 = InputBox("Enter Current year:")  
Age=Year2 - Year1  
Rem Insert a blank line  
Print  
Print " Your age is "; Age  
End Sub
```

کد مربوط به تعیین عملکرد دکمه‌ی Exit را نیز چنین می‌نویسیم :

```
Private Sub Command2_Click()  
End  
End Sub
```



شکل ۱۲-۶



شکل ۱۲-۷

دستور **End** برای خاتمه‌ی اجرای برنامه است.

با اجرای این برنامه، فرمی شامل دو کلید **Run** و **Exit** ظاهر می‌شود که با کلیک روی دکمه‌ی **Run**، کادرهای روبه‌رو (شکل ۱۲-۶) ظاهر می‌شوند و برنامه منتظر می‌ماند تا کاربر سال تولد و سال جاری را در کادرهای مربوطه وارد کند.

خروجی برنامه می‌تواند به شکل روبه‌رو باشد: (شکل ۱۲-۷)

دستور **Rem**^۱ برای نوشتن توضیحات (به منظور مستندسازی) در برنامه به کار می‌رود. خطی از برنامه که با عبارت **Rem** آغاز می‌شود، در اجرای برنامه تأثیر ندارد. به جای عبارت **Rem** می‌توان از علامت «'» استفاده کرد.

مثال ۲: برنامه‌ای بنویسید که نام و شماره تلفن کاربر را از ورودی بخواند و آن را با پیغام مناسبی نمایش دهد.

ابتدا در پنجره‌ی طراحی فرم، عناصر **Text Box** برای دریافت ورودی، **Label** برای نمایش عناوین و دکمه‌های فرمان برای اجرا و خروج از برنامه را قرار می‌دهیم. سپس کد برنامه را چنین می‌نویسیم:

```
Private Sub Form_Load ()
    Command1.Caption = "DISPLAY"
    Command2.Caption = "EXIT"
    Text1 = ""
    Text2 = ""
    Label1.Caption = "Your name:"
    Label2.Caption = "Your phone number:"
    Label3.Caption = ""
End Sub
```

۱. **Rem** مخفف **Remark** به معنای توضیح است.

```

Private Sub Command1_Click()
Dim x As String , y As String
x = Text1
y = Text2
Label3.Caption = "Hi" + x + "Your phone number is" + y
End Sub

```

```

Private Sub Command2_Click()
End
End Sub

```

نتیجه‌ی اجرای برنامه چنین است : (شکل ۸-۱۲)



شکل ۸-۱۲

مثال ۳: برنامه‌ای بنویسید که سه عدد را بخواند و میانگین آن‌ها را نمایش دهد.
 کد این برنامه چنین است :

```

Private Sub Form_Load()
Form1.Caption="Average of three number"
Command1.Caption="RUN"
Command2.Caption="EXIT"
Text1 = ""
Text2 = ""
Text3 = ""
Label1 = "First number:"
Label2 = "Second number:"
Label3 = "Third number:"
Label4 = "" 'for display the average
End Sub

```

عبارتی که بعد از علامت «?» آمده، یک عبارت توضیحی است و در اجرای برنامه تأثیری ندارد.

```

Private Sub Command1_Click()
Dim x As Single,
    y As Single,
    z As Single,
    m As Single
x = Val(Text1)

```

```

y = Val(Text2)
z = Val(Text3)
m = (x + y + z) / 3
Label4.Caption="Average=" & Str(m)
End Sub

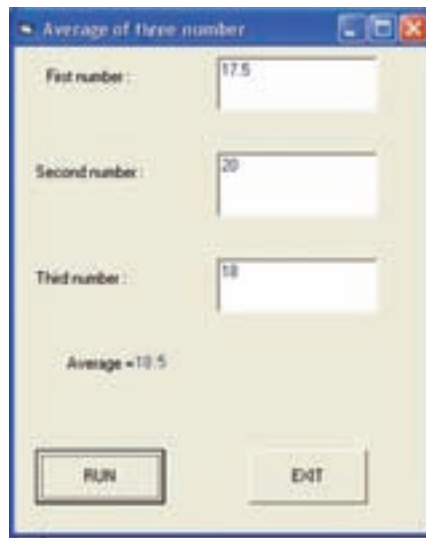
```

```

Private Sub Command2_Click()
End
End Sub

```

نتیجه‌ی اجرای برنامه چنین است: (شکل ۹-۱۲)



شکل ۹-۱۲

دستورات کنترلی

در بخش یازدهم با ساختارهای کنترلی که برای کنترل ترتیب اجرای دستورالعمل‌ها به کار می‌رود آشنا شدیم. زبان ویژوال بیسیک دارای دستوراتی است که برای پیاده‌سازی ساختارهای کنترلی در یک برنامه به کار می‌روند. قبل از معرفی این دستورات، با دو دسته‌ی دیگر از عملگرها آشنا می‌شویم:

عملگر مقایسه‌ای	توضیح
=	تساوی
<>	عدم تساوی
<	کوچک‌تر از
<=	کوچک‌تر یا مساوی
>	بزرگ‌تر از
>=	بزرگ‌تر یا مساوی

عملگرهای مقایسه‌ای برای مقایسه‌ی مقادیر یا عبارات عددی و رشته‌ای به کار می‌روند و بدین ترتیب عبارات منطقی را تشکیل می‌دهند که ارزش درست یا نادرست دارند.

مثال:

$x < \text{Sqr}(A+B)$

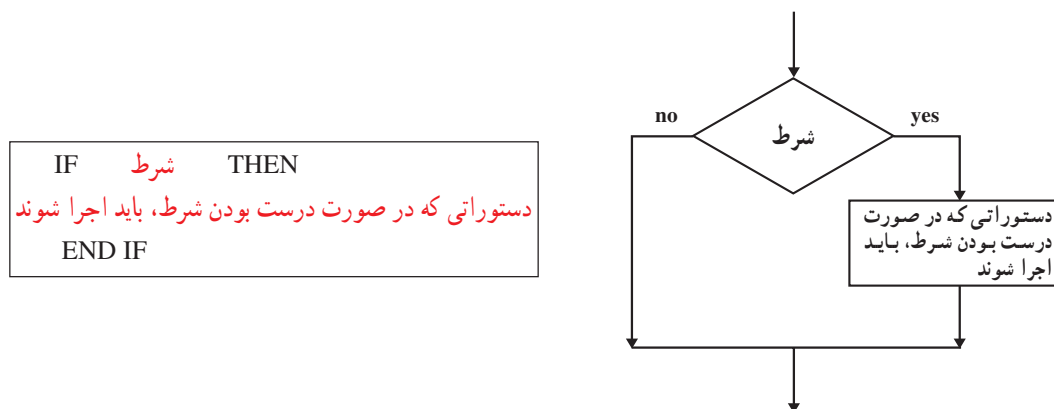
Pass <> Password

ویژوال بیسیک علاوه بر عملگرهای ریاضی و مقایسه‌ای، دارای عملگرهای منطقی AND، OR و NOT نیز است. دو عملگر اول برای ترکیب عبارات منطقی و عملگر NOT برای معکوس کردن ارزش یک عبارت منطقی به کار می‌رود. به مثال‌های جدول زیر که از عملگرهای منطقی و مقایسه‌ای استفاده شده، توجه نمایید:

عبارت منطقی	توضیح
$x=5 \text{ AND Name}="Ali"$	مقدار متغیر x برابر ۵ است و متغیر رشته‌ای Name برابر Ali.
$\text{NOT } (x>0)$	مقدار متغیر x بزرگ‌تر از عدد صفر نباشد.
$c \leq \text{Int}(x) \text{ OR Flag} <> 1$	مقدار متغیر c کوچک‌تر یا مساوی با جزء صحیح متغیر x است یا مقدار متغیر Flag با عدد ۱ برابر نیست.

دستورات شرطی

در زبان برنامه‌نویسی ویژوال بیسیک برای پیاده‌سازی ساختار انتخاب از دستورات شرطی استفاده می‌کنیم. دستور شرطی **IF... THEN**: شکل کلی این دستور چنین است:



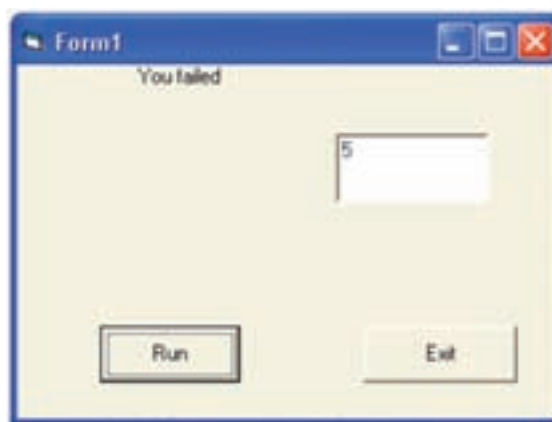
مثال ۴: برنامه‌ای بنویسید که نمره‌ی یک درس دانش‌آموز را بخواند. در صورتی که نمره کم‌تر از ۱۰ باشد، پیغام "you failed" را نمایش دهد.

کد برنامه و نتیجه‌ی اجرا چنین است:

```
Private Sub Command1_Click()  
Dim g As Single  
g = Val(Text1)  
If g < 10 Then Print Tab(15); "you failed"  
End Sub
```

```
Private Sub Command2_Click()
End
End Sub
```

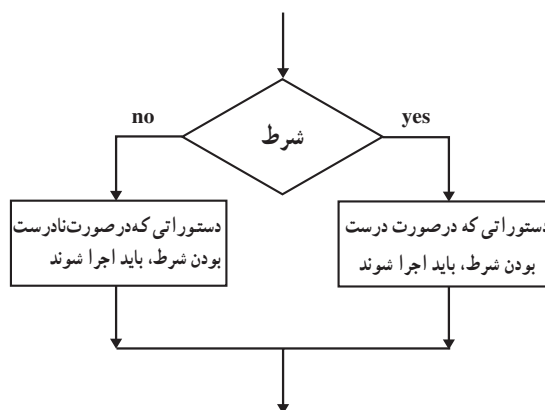
```
Private Sub Form_Load()
Command1.Caption="Run"
Command2.Caption="Exit"
End Sub
```



شکل ۱۰-۱۲

دستور IF...THEN... ELSE شکل کلی این دستور چنین است :

```
IF شرط THEN
دستوراتی که در صورت درست بودن شرط، باید اجرا شوند
ELSE
دستوراتی که در صورت نادرست بودن شرط، باید اجرا شوند
END IF
```



مثال ۵: برنامه‌ی مثال قبل را چنان تکمیل کنید که در صورتی که نمره‌ی دانش‌آموز بیش‌تر یا مساوی ۱۰ باشد، پیغام "you passed" را نمایش دهد.

کد برنامه و نتیجه‌ی اجرا چنین است :

```
Private Sub Command1_Click()
Dim g As Single
g = Val(Text1)
If g<10 Then
Print Tab(15);"you failed"
Else
Print Tab(15);"you passed"
End If
End Sub
```

```
Private Sub Command2_Click()
End
End Sub
```

```
Private Sub Form_Load()
Command1.Caption = "Run"
Command2.Caption = "Exit"
End Sub
```

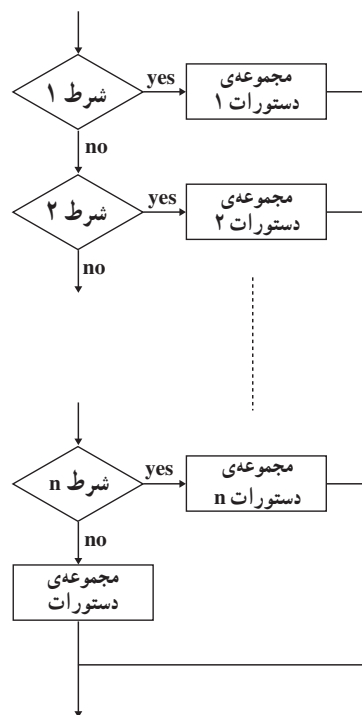
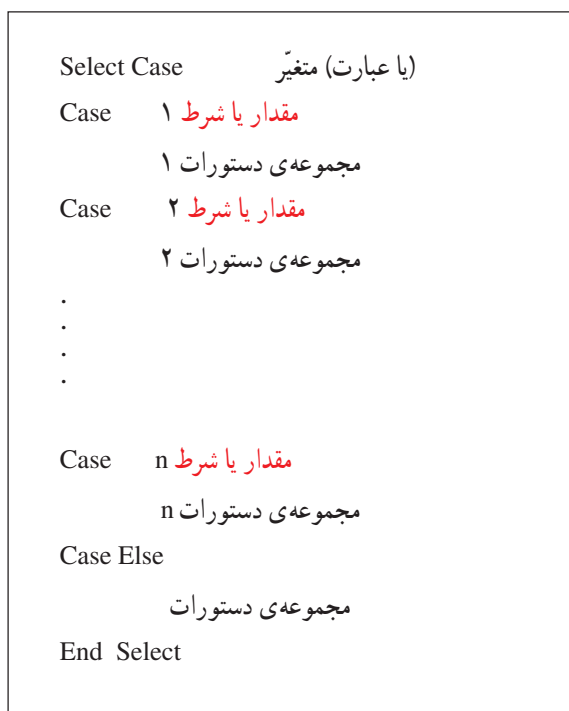


شکل ۱۱-۱۲

فعالیت ۱

برنامه‌ای بنویسید که سه عدد را از ورودی بخواند و در صورتی که این سه عدد تشکیل یک مثلث بدهند، پیغام "This is triangle" و در غیر این صورت، پیغام "This is not triangle" را نمایش دهد.

دستور **Select Case**: در صورتی که تعداد انتخاب‌ها بیش از دو مورد باشد، از این دستور استفاده می‌کنیم. کاربرد این دستور، در مواردی است که بخواهیم براساس مقادیر مختلف یک متغیر (یا یک عبارت) دستورات مختلفی را اجرا کنیم. شکل کلی این دستور چنین است:

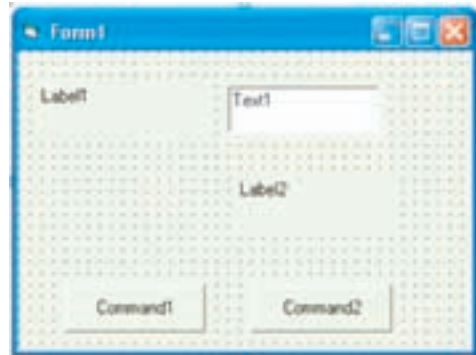


دستور Case شکل‌های مختلفی دارد که در مثال‌های زیر با آن‌ها آشنا می‌شوید.
 مثال ۶: برنامه‌ای بنویسید که شماره‌ی یک ماه را بگیرد و تعداد روزهای آن را نمایش دهد.
 فرم طراحی شده، کد برنامه و نتیجه‌ی اجرا چنین است :

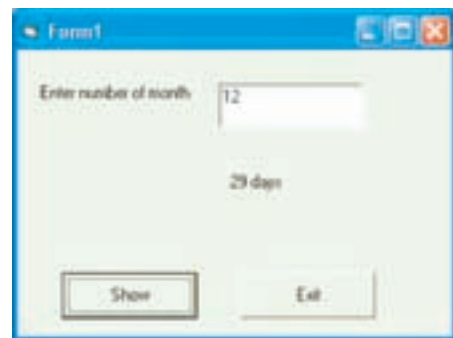
```
Private Sub Command1_Click()
Dim x As Integer
x = Val(Text1)
Select Case x
Case 1 To 6
Label2="31 days"
Case 7 To 11
Label2="30 days"
Case 12
Label2="29 days"
Case Else
Label2=" invalid number"
End Select
End Sub
```

```
Private Sub Command2_Click()
End
End Sub
```

```
Private Sub Form_Load()
Command1.Caption = "Show"
Command2.Caption = "Exit"
Text1=""
Label1 = "Enter number of month"
Label2=""
End Sub
```



الف – فرم طراحی شده



ب – نتیجه‌ی اجرا

شکل ۱۲-۱۲

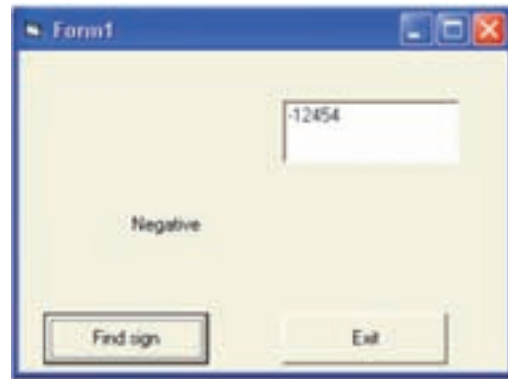
مثال ۷: برنامه‌ای بنویسید که یک عدد را بخواند و آن را تعیین علامت کند.
 کد برنامه و نتیجه‌ی اجرا چنین است :

```
Private Sub Command1_Click()
Dim Num As Integer
Num = Val(Text1)
Select Case Num
Case Is>0
Label1.Caption="Positive"
Case Is<0
Label1.Caption="Negative"
Case Else
```

```
Label1.Caption="Zero"
End Select
End Sub
```

```
Private Sub Command2_Click()
End
End Sub
```

```
Private Sub Form_Load()
Command1.Caption = "Find sign"
Command2.Caption = "Exit"
Text1 = ""
Label1 = ""
End Sub
```



شکل ۱۳-۱۲

فعالیت ۲

برنامه‌ای بنویسید که نام و نمره‌ی درس رایانه‌ی یک دانش‌آموز را بخواند و معادل حرفی آن را با پیغام مناسب مطابق جدول زیر نمایش دهد.

معادل حرفی	نمره از ۲۰
A	$18 \leq N \leq 20$
B	$15 \leq N < 18$
C	$12 \leq N < 15$
D	$10 \leq N < 12$
E	$0 \leq N < 10$

دستورات حلقه

همان‌گونه که در فصل قبل گفته شد، حلقه یک ساختار کنترلی است که برای تکرار یک یا چند عبارت به کار می‌رود. حلقه‌ها به‌طور کلی می‌توانند از دو نوع متفاوت باشند:

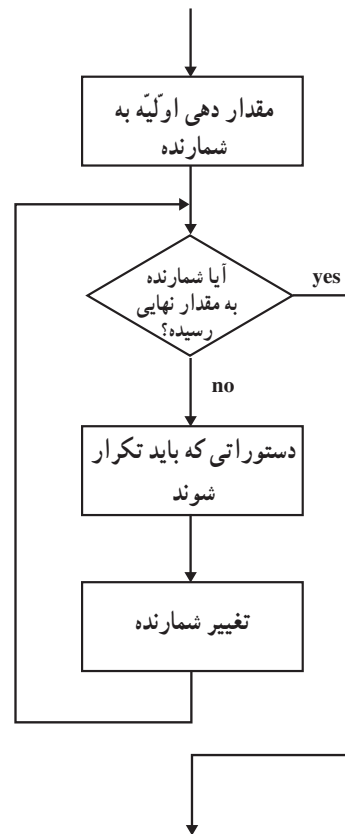
حلقه‌ی کنترل شده با شمارش: حلقه‌ای است که تعداد دفعات تکرار آن از ابتدا مشخص است و در واقع تکرار حلقه بستگی به مقدار یک شمارنده دارد. وقتی شمارنده به مقدار مشخصی رسید، تکرار حلقه متوقف می‌شود.

برای کدنویسی این نوع حلقه از دستور FOR ... NEXT استفاده می‌کنیم.

دستور FOR...NEXT: شکل کلی این دستور چنین است :

مقدار تغییر شمارنده STEP مقدار نهایی TO مقدار اولیه = متغیر شمارنده FOR
 دستوراتی که باید تکرار شوند
 متغیر شمارنده NEXT

- اگر مقدار تغییر شمارنده، ۱ باشد، می توان از نوشتن آن صرف نظر کرد.
- در صورتی که مقدار نهایی، کمتر از مقدار اولیه باشد، مقدار تغییر شمارنده باید یک عدد منفی باشد.



مثال ۸: برنامه ای بنویسید که شمارش معکوس ۱۰ تا ۱ را انجام دهد و در پایان، پیغام «Fire» را نمایش دهد.
 کد برنامه و نتیجه ی اجرا چنین است :

```

Private Sub Command1_Click()
For i=10 To 1 Step-1
Print i;
Next i
Print "Fire"
End Sub
Private Sub Command2_Click()
End
End Sub

Private Sub Form_Load()
Command1.Caption = "Start"
Command2.Caption = "Exit"
End Sub
  
```



شکل ۱۴-۱۲

مثال ۹: برنامه‌ای بنویسید که اعداد زوج بین ۰ و ۱۰۰ را تولید کرده و مجموع آن‌ها را نمایش دهد.
کد برنامه و نتیجه‌ی اجرا چنین است:

```
Private Sub Command1_Click()
Dim i As Single, sum As Single
sum=0
For i=2 To 100 Step 2
sum=sum+i
Next i
Label1.Caption="Sum is" & Str(sum)
End Sub
```

```
Private Sub Command2_Click()
End
End Sub
```

```
Private Sub Form_Load()
Form1.Caption="Sum of even 0 to 100"
Label1.Caption=""
Command1.Caption="show"
Command2.Caption="Exit"
End sub
```



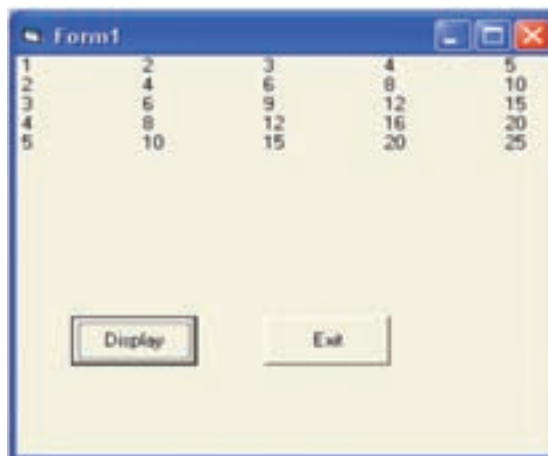
شکل ۱۵-۱۲

مثال ۱۰: برنامه‌ای بنویسید که جدول ضرب ۱ تا ۵ را نمایش دهد.
کد برنامه و نتیجه‌ی اجرا چنین است:

```
Private Sub Command1_Click()
Rem این برنامه جدول ضرب ۱ در ۵ را نمایش می‌دهد
Dim i As Integer, j As Integer
For i=1 To 5
    For j=1 To 5
        Print i*j,
    Next j
    Print
Next i
End Sub
```

```
Private Sub Command2_Click()
End
End Sub
```

```
Private Sub Form_Load()
Command1.Caption = "Display"
Command2.Caption = "Exit"
End Sub
```



شکل ۱۶-۱۲

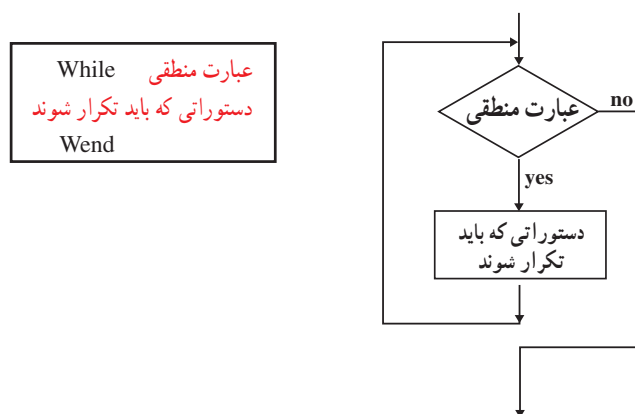
فعالیت ۳

برنامه‌ای بنویسید که تعداد دروس و نمرات مربوط به هر درس یک دانش‌آموز را بخواند و معدل دانش‌آموز را نمایش دهد.

حلقه‌ی کنترل شده با رخداد: در این نوع حلقه تعداد دفعات تکرار از ابتدا مشخص نیست و بستگی به رخداد خاصی دارد و تا زمانی که آن رخداد روی نداده است تکرار می‌شود.

برای کدنویسی این نوع حلقه دستورات مختلفی در ویژوال بیسیک وجود دارد. یکی از این دستورات While...Wend است.

دستور While...Wend: شکل کلی این دستور چنین است:



While عبارت منطقی
دستوراتی که باید تکرار شوند
Wend

مثال ۱۱: برنامه‌ای بنویسید که یک «کلمه‌ی عبور» از کاربر دریافت کند. تا زمانی که کاربر، کلمه‌ی عبور «dana» را وارد نکرده، برنامه تکرار شود و در صورتی که کلمه‌ی عبور «dana» وارد شد، پیغام «Welcome» را نمایش دهد. کد برنامه و نتیجه‌ی اجرا چنین است:

```
Private Sub Command1_Click()
    Dim x As String
    x=InputBox("Enter your password","Get password")
    While x <> "dana"
        x=InputBox("Enter your password", "Get password")
    Wend
    Label1.Caption = "Welcome"
End Sub

Private Sub Command2_Click()
    End
End Sub

Private Sub Form_Load()
    Command1.Caption = "Get password"
    Command2.Caption = "Exit"
    Label1.Caption = ""
End Sub
```



شکل ۱۲-۱۲

برنامه‌ای بنویسید که قیمت تعدادی کالا را بخواند. این عمل را تا زمانی تکرار کند که کاربر قیمت کالا را «صفر» وارد کند. در پایان، مجموع قیمت کالاها را نمایش دهد.

خطایابی برنامه

همان‌طور که در بخش یازدهم خواندیم، آزمایش برنامه و یافتن اشتباهات احتمالی آن، یکی از مراحل فرایند برنامه‌نویسی است. اشتباهات برنامه‌نویسی به دو دسته تقسیم می‌شوند. اشتباهات دستوری و اشتباهات منطقی. اشتباه دستوری^۱، یعنی عدم رعایت قواعد زبان برنامه‌نویسی؛ به‌طور مثال، در زبان برنامه‌نویسی ویژوال بیسیک، دستورالعمل "Wile...Wend" نادرست است و شکل درست آن "While...Wend" است. اشتباه منطقی^۲، زمانی رخ می‌دهد که منطق برنامه غلط باشد. به‌عبارت دیگر اشتباهات منطقی، اشتباهات موجود در الگوریتم برنامه است. حتی گاهی برنامه، بدون پیغام خطا اجرا می‌شود ولی خروجی مورد نظر تولید نمی‌شود. معمولاً یافتن این اشتباهات بسیار مشکل است.

برای مثال، اگر برنامه‌نویس عبارت $\frac{a+b}{c}$ را به صورت $a + b / c$ بنویسد، خطای منطقی رخ می‌دهد.

روش‌های مختلفی برای یافتن و رفع هر دو نوع اشتباه وجود دارد. برخی از آن‌ها عبارت‌اند از: استفاده از برنامه‌های مترجم؛ می‌دانید که برنامه‌ی مترجم، برنامه‌ی نوشته شده را از زبان برنامه‌نویسی به زبان ماشین ترجمه می‌کند. مترجم، هنگام ترجمه‌ی برنامه، اشتباهات دستوری را مشخص کرده و در هر مورد پیغام مناسبی به برنامه‌نویس می‌دهد. آزمایش برنامه با داده‌های نمونه روی رایانه؛ پس از آن که همه‌ی اشتباه‌های دستوری تصحیح شدند، برنامه برای یافتن اشتباه‌های منطقی آزمایش می‌شود. داده‌های نمونه، طوری انتخاب می‌شوند که همه‌ی حالت‌های خاص برنامه را آزمایش کند. آزمایش برنامه توسط گروهی از کاربران؛ پس از انجام مراحل فوق، برنامه در اختیار گروهی از کاربران قرار می‌گیرد تا آن را به صورت آزمایشی اجرا کنند. مزیت این کار این است که یک کاربر، ممکن است به اشکالی برخورد کند که برنامه‌نویس پیش‌بینی نکرده باشد.

– برخی از قابلیت‌ها و دستورات کلی یک زبان برنامه‌نویسی عبارت‌اند از :

۱– نگهداری و آدرس‌دهی انواع داده‌ها

۲– عملیات روی داده‌ها

۳– توابع

۴– دستورات ورودی و خروجی

۵– دستورات کنترلی

– برخی از انواع داده‌ها در ویژوال بیسیک عبارت‌اند از : عددی، رشته‌ای، تاریخ و بولین.

– متغیر، نامی برای محلی از حافظه‌ی رایانه است که داده در آن قرار می‌گیرد.

– عملگرهای ریاضی موجود در ویژوال بیسیک عبارت‌اند از : Mod ، \ و ^ و / و * و - و +

– توابع، برنامه‌های از پیش نوشته شده‌ای هستند که برنامه‌نویس می‌تواند از آن‌ها در برنامه‌ی خود استفاده کند.

– هر عملی که کاربر برای برقراری ارتباط با برنامه انجام می‌دهد، یک رویداد نامیده می‌شود.

– اولین قدم در تهیه‌ی برنامه در ویژوال بیسیک، طراحی فرم است.

– ویژوال بیسیک دارای یک پنجره برای طراحی فرم است که برنامه‌نویس می‌تواند ابزارها و عناصر گرافیکی

مورد نیاز را روی آن قرار دهد.

– چند نمونه از عناصر مورد استفاده در ویژوال بیسیک عبارت‌اند از : دکمه‌ی فرمان، برچسب و جعبه‌ی متن که

در جعبه ابزار قرار دارند.

– هر فرم یا عنصر موجود در آن دارای مشخصه‌هایی است که در پنجره‌ی «مشخصات» (properties) قابل تنظیم‌اند.

– پس از طراحی فرم، لازم است برنامه‌نویس برای ارتباط عناصر با یکدیگر، کد مربوط به رویدادهای پیش‌بینی شده

را بنویسد.

– دستور Print برای نمایش اطلاعات خروجی به کار می‌رود.

– عنصر Label برای نمایش یک متن روی فرم به کار می‌رود.

– تابع () InputBox برای دریافت داده از کاربر به کار می‌رود.

– خطی از برنامه که با REM یا آپاستروف (') شروع شود شامل توضیحات برنامه است و در اجرای برنامه اثری

ندارد.

– عنصر TextBox برای ورود داده‌های رشته‌ای استفاده می‌شود.

– دستور If...Then برای کدنویسی ساختار انتخاب به کار می‌رود.

– در صورتی که تعداد انتخاب‌ها، بیش از دو مورد باشد، از دستور Select Case استفاده می‌شود.

– حلقه‌ها برای تکرار یک یا چند دستور به کار می‌روند و بر دو نوع‌اند :

۱– حلقه‌ی کنترل شده با شمارش که تعداد دفعات تکرار آن از ابتدا مشخص است.

۲– حلقه‌ی کنترل شده با یک رخداد که تعداد دفعات اجرای آن از ابتدا مشخص نیست و بستگی به یک رخداد

خاص دارد.

– دستور For...Next برای کدنویسی حلقه‌های کنترل شده با شمارش به کار می‌رود.

– دستور While...Wend برای کدنویسی حلقه‌های کنترل شده با رخداد به کار می‌رود.



پرسش و تمرین

۱- عبارتهای زیر را با ساختار ویژگی بیسیک بنویسید.

الف) $X = uv(u + v)$

ب) $Y = (A + B)(C + D)$

ج) $Z = 8X^2 - 2X + 1$

د) $Y = \frac{\sqrt{(a+b)^3}}{a-b}$

۲- اولویت انجام عملیات را در عبارات زیر تعیین کنید:

الف) $\frac{a^2 + 2b}{a + c}$

ب) $\left(3 \times 5 \times \left(9 + \frac{9}{35} \right) \right)$

ج) $a - \frac{a^3 + b^3}{bc}$

۳- برنامه‌ای بنویسید که دو عدد را بخواند و در صورتی که عدد دوم صفر نباشد، خارج قسمت صحیح تقسیم عدد اول بر دوم را نمایش دهد و در غیر این صورت پیغام "not defined" را ظاهر کند.

۴- برنامه‌ای بنویسید که تاریخ تولد کاربر را به صورت «روز، ماه، سال» دریافت کند و سن او را به صورت «روز، ماه، سال» نمایش دهد.

۵- برنامه‌ای بنویسید که ضرایب یک معادله‌ی درجه دوم را بخواند و ریشه‌های آن را محاسبه کرده و نمایش دهد.

۶- برنامه‌ای بنویسید که شماره‌ی روز هفته را بخواند و نام آن روز را نمایش دهد.

۷- برنامه‌ای بنویسید که با دریافت مبلغ حقوق یک کارمند، مالیات آن را با استفاده از جدول زیر به دست آورد.

— تا ۲۰۰,۰۰۰ تومان معاف

— تا ۸۰۰,۰۰۰ تومان، ۱٪ به‌ازای‌مازاد

— از ۸۰۰,۰۰۰ تومان به بالا ۱۲٪ به‌ازای‌مازاد

۸- برنامه‌ای بنویسید که یک عدد چهاررقمی را بخواند و مقلوب آن را نمایش دهد. (مثال: مقلوب ۱۲۳۴

می‌شود ۴۳۲۱)

۹- برنامه‌ای بنویسید که مضارب دو رقمی عدد ۳ و تعداد آن‌ها را نمایش دهد.

۱۰- برنامه‌ای بنویسید که نام و درجه حرارت ۱۰ شهر را بخواند و درجه حرارت گرم‌ترین شهر و سردترین

شهر را همراه نام آن‌ها نمایش دهد.

فعالیت‌های تکمیلی

بخش اول

– درباره‌ی رایانه‌های نسل پنجم تحقیق کنید.

بخش دوم

– با رجوع به راهنمای استفاده از دو وسیله‌ی خاص، ویژگی‌ها و خصوصیات هر راهنما (مانند نحوه‌ی استفاده از وسیله و سادگی آن، استفاده از تصاویر، ذکر مشخصات فنی، عیب‌یابی، نحوه‌ی دریافت خدمات پس از فروش و غیره) را بیان کنید و نقاط ضعف و قدرت دو راهنما را برشمرده، با هم مقایسه کنید.

بخش سوم

- ۱– نقش ارتباطات را در شغل آینده‌ی خود بررسی کنید.
- ۲– در مورد وسایل ارتباطی بی‌سیم تحقیق کنید.
- ۳– در سیستم اطلاعاتی دانش‌آموزان کشور (سیستم دانا) ضرورت‌های ارتباطات و تبادل اطلاعات را بررسی کنید.
- ۴– در صورتی که امکان ارتباط با شبکه‌ی اینترنت مدرسه‌ی خود را دارید، پهنای باند و نوع پروتکل آن را بررسی کنید. آیا هنگامی که به اینترنت متصل هستید سرعت ارسال و دریافت داده‌ها به اندازه‌ی پهنای باند فوق است؟
- ۵– شبکه‌ی رایانه‌ای سازمان آموزش و پرورش منطقه یا استان خود را بررسی کرده و نوع شبکه و کاربردهای آن را مشخص کنید.
- ۶– درباره‌ی WiMax تحقیق کنید.

بخش چهارم

- ۱– برخی از شرکت‌ها از دفاتر خود به صورت اشتراکی استفاده می‌نمایند. نحوه‌ی کار به این شکل است که عده‌ای از کارمندان در سه روز هفته از دفتر مورد نظر استفاده می‌کنند و در سه روز دیگر جای خود را به گروه دیگری می‌دهند. در طی چند روزی که کارمندان از دفتر استفاده نمی‌کنند می‌توانند در منزل به کار خود ادامه دهند و با استفاده از امکانات ارتباطات و اینترنت و از طریق رایانه با دفتر در ارتباط باشند.
- به نظر شما این نحوه‌ی کار کردن، چه مزایا و معایبی نسبت به روش سنتی دارد؟ چه مزایا و معایبی برای کارفرمایان و کارمندان دارد؟
- ۲– درباره‌ی یکی از مراکزی که امکان دست‌یابی از راه دور (Telnet) را فراهم می‌کنند، تحقیق کنید.
- ۳– نمونه‌ای از اینترنت و اکسترانت داخلی بیابید و درباره‌ی آن توضیح دهید.
- ۴– در مورد یک نمونه از نرم‌افزارهای رایج firewall تحقیق کنید و قابلیت‌های آن را شرح دهید.
- ۵– درباره‌ی چند نمونه موتور جست‌وجو تحقیق کنید.

بخش پنجم

- ۱- با در نظر گرفتن یکی از شاخه‌های علوم درباره‌ی تأثیر رایانه در آن رشته تحقیق نمایید.
- ۲- تأثیر ویروس‌های رایانه‌ای را در اتلاف وقت و هزینه بررسی کنید.
- ۳- درباره‌ی انواع ویروس‌یاب‌های جدید و قابلیت‌های هر کدام تحقیق کنید.
- ۴- درباره‌ی موضوعات زیر تحقیق کنید :
 - الف) کتاب الکترونیکی یا e-book
 - ب) مدرسه‌ی مجازی یا Virtual School
 - ج) تجارت الکترونیکی یا e-commerce
 - د) بانکداری الکترونیکی e-banking
- ۵- درباره‌ی تأثیر نانو تکنولوژی در ساخت رایانه‌های جدید تحقیق کنید.
- ۶- درباره‌ی انواع برنامه‌های مخرب و عملکرد آن‌ها تحقیق کنید.
- ۷- چه شیوه‌هایی برای حفاظت رایانه در برابر انواع برنامه‌های مخرب می‌توان به کار برد؟
- ۸- نقش رایانه در شغل آینده شما چیست؟
- ۹- صنعت تولید نرم افزار، مدت‌هاست که با مشکل تکثیر غیرقانونی نرم افزار مواجه است. با ارائه‌ی فیلم‌های سینمایی بر روی دیسک‌های DVD و CD، صنعت سینما با مشکل مشابهی روبه‌رو شده است. آیا این مشکل قابل کنترل است؟ دلایل خود را ذکر کنید.

بخش ششم

- ۱- مشخصات رایانه‌های مدرسه‌ی خود را از نظر نوع و سرعت CPU، ظرفیت حافظه‌ی RAM و نوع و تعداد درگاه‌ها بررسی کنید.
- ۲- درباره‌ی بردهای Dual CPU و ریزپردازنده‌های Dual Core تحقیق کنید.
- ۳- فرض کنید می‌خواهید رایانه‌ی شخصی خود را ارتقاء دهید؛ با استفاده از دفترچه‌ی راهنمای برد اصلی، هریک از موارد زیر را بررسی کنید :
 - الف) برد اصلی رایانه‌ی شما چه ریزپردازنده‌هایی را پشتیبانی می‌کند؟
 - ب) چه کارت‌های توسعه‌ای را می‌توان بر روی برد اصلی نصب کرد؟
 - ج) حافظه‌ی RAM سیستم را تا چه اندازه می‌توانید افزایش دهید؟
 - د) نوع و ظرفیت حافظه‌ی RAM را مشخص کنید.
- ۴- درباره‌ی انواع حافظه‌ی RAM و ROM، سرعت، نحوه‌ی عملکرد و کاربرد آن‌ها تحقیق کنید.
- ۵- درباره‌ی سرعت انواع درگاه‌ها تحقیق کنید.

بخش هفتم

- ۱- وسایل چندکاره‌ای وجود دارند که به تنهایی به جای چند وسیله به کار می‌روند. برای مثال، برخی از این دستگاه‌ها قابلیت‌های متفاوتی چون پویس کردن، ارسال و دریافت نامبر، کیبورد و چاپ کردن دارند. محاسن و معایب این وسایل چند کاره را در مقایسه با وسایلی که هر کدام برای یک کار خاص ساخته شده‌اند، بیان کنید.

۲- درباره‌ی سیستم‌های تشخیص صدا^۱، انواع آن‌ها، نحوه‌ی عملکرد و محدودیت‌های آن‌ها مطالبی را گردآوری نمایید.

۳- درباره‌ی وسایل ورودی و خروجی بی‌سیم (wireless) تحقیق کنید.

۴- درباره‌ی وسایل ورودی حسگر (Sensors) تحقیق کنید.

۵- درباره‌ی صفحه کلید مجازی تحقیق کنید.

۶- درباره‌ی صفحه نمایش لمسی تحقیق کنید.

بخش هشتم

۱- درباره‌ی دیسک‌های Blu-Ray تحقیق کنید.

۲- درباره‌ی انواع دیسک‌های سخت و ویژگی‌های آن‌ها تحقیق کنید.

۳- مشخصات حافظه‌های جانبی مورد استفاده در رایانه‌های مدرسه‌ی خودتان را بررسی کنید.

۴- درباره‌ی نحوه‌ی ذخیره‌سازی و حذف اطلاعات در حافظه‌های Flash تحقیق کنید.

بخش نهم

۱- درباره‌ی کاربرد و نحوه‌ی عملکرد برنامه‌های کمکی Movie Maker, Disk Defragmenter و Multimedia Player تحقیق کنید.

۲- درباره‌ی نسل‌های زبان‌های برنامه‌نویسی تحقیق کنید.

۳- درباره‌ی برنامه‌نویسی شیء گرا و برنامه‌نویسی ویژوال تحقیق کنید.

بخش دهم

۱- با در نظر گرفتن شغل مورد علاقه‌ی خود، درباره‌ی انواع برنامه‌های کاربردی مربوط به آن تحقیق کنید.

۲- یکی دیگر از برنامه‌های کاربردی عمومی، مدیریت اطلاعات شخصی^۲ یا (PIM) است. درباره‌ی این برنامه‌ها و ویژگی‌های آن‌ها اطلاعاتی جمع‌آوری کرده و در کلاس ارائه دهید.

۳- یک برنامه‌ی گرافیکی را از نظر نوع کاربرد و امکانات آن بررسی کنید.

۴- یکی از برنامه‌های کاربردی را که در زمینه‌ی تولید فیلم یا انیمیشن به کار می‌روند، بررسی کنید.

۵- یکی از برنامه‌های کاربردی مدیریت بانک‌های اطلاعاتی را بررسی کنید.

۶- یکی از برنامه‌های کاربردی از مجموعه‌ی نرم‌افزارهای Office، نرم‌افزار OneNote است. درباره‌ی امکانات آن تحقیق کنید.

بخش دوازدهم

۱- عملکرد توابع و عناصر زیر را بررسی کنید.

الف) تابع () MsgBox

ب) عنصر CheckBox

ج) عنصر OptionButton

د) توابع رشته‌ای () Replace, () String, () InStr, () Right, () Left, () Mid

ه) تابع () LoadPicture

و) عناصر Image , PictureBox

۲- نحوه‌ی عملکرد برنامه‌ی Crystal Report در تهیه‌ی گزارش چایی از اجرای برنامه‌ی ویژوال بیسیک را

بررسی کنید.

۳- عملکرد متغیرهای اندیس‌دار (آرایه‌ها) را در V.B. بررسی کنید.