

توانایی تعریف انواع متغیرها، ثابت‌ها و استفاده از عملگرهای ریاضی و رشته‌ای

هدف‌های رفتاری

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

- ۱- انواع متغیرها و ثابت‌ها را بیان کند و نحوه تعریف و استفاده از آن‌ها را توضیح دهد.
- ۲- عملگرهای ریاضی و رشته‌ای را بیان کرده، نحوه استفاده و اولویت اجرای آن‌ها را نسبت به یکدیگر شرح دهد.
- ۳- متغیرهای عمومی و محلی را توضیح دهد و نحوه تعریف و کار با آن‌ها را بیان کند.

کلیات

در طراحی پروژه‌های واقعی لازم است تا داده‌ها و اطلاعات جمع‌آوری شده به کامپیوتر و نرم‌افزار داده شود و پس از انجام پردازش‌هایی روی آن‌ها، نتیجه یا نتایج حاصل به کاربر ارائه شود. بنابراین در تمام مراحل ورود، محاسبه و ارائه نتایج نیاز به ذخیره‌سازی داده و اطلاعات در حافظه کامپیوتر احساس می‌شود که در این رابطه می‌توانید از حافظه اصلی و حافظه جانبی استفاده کنید. در این واحد کار نحوه استفاده از حافظه اصلی یا RAM را برای ذخیره‌سازی داده‌ها و اطلاعات فرا می‌گیرید، به علاوه برای انجام عملیات ریاضی، نحوه استفاده از انواع عملگرها را در ویژوال بیسیک می‌آموزید.

۴-۱ نحوه تعریف و استفاده از انواع متغیرها در ویژوال بیسیک

یکی از عملیاتی که در برنامه‌ها انجام می‌گیرد، ذخیره‌سازی داده و اطلاعات در حافظه کامپیوتر است. ابتدا نحوه معرفی انواع متغیرها را در ویژوال بیسیک توضیح می‌دهیم، سپس نحوه انجام این کار را همراه با یک مثال آموزش می‌دهیم.

۴-۱-۱ نحوه معرفی متغیرها در ویژوال بیسیک

متغیر، مکانی از حافظه است که برای نگهداری داده‌ها و اطلاعات حاصل از پردازش داده‌ها استفاده می‌شود، به عبارت دیگر متغیر ظرف نگهداری داده‌ها و اطلاعات است و نوع آن با توجه به داده‌ای که نگهداری می‌کنند، تعیین می‌شود. به علاوه هر متغیر یک نام ویژه دارد که به وسیله آن می‌توان داده‌ها را در حافظه ذخیره کرد یا مقدار آن را مورد دستیابی قرار داد. برای تعریف متغیرها می‌توانید از دستور Dim به این صورت استفاده کنید:

نوع داده As نام متغیر Dim

نام متغیر دلخواه بوده و می‌توانید از کاراکترهای حرفی، رقمی یا ترکیبی از آن‌ها استفاده کنید، اما بهتر است نام متغیر متناسب با مقداری که در آن ذخیره می‌شود، تعیین شده و از اسامی طولانی و نامأنوس استفاده نشود. برای نام‌گذاری یک متغیر این موارد را رعایت کنید:

- الف- نام متغیر با یک کاراکتر حرفی آغاز شود و حداکثر ۲۵۵ کاراکتر باشد.
- ب- استفاده از فاصله خالی، نقطه، کاما و نظایر آن‌ها مجاز نیست، اما استفاده از کاراکتر زیرخط (_) امکان‌پذیر است.

ج- در نام‌گذاری متغیرها از علایم و اسامی رزرو شده در ویژوال بیسیک استفاده نکنید. علایم و اسامی رزرو شده در ویژوال بیسیک علایم و عباراتی هستند که از آن‌ها برای انجام اعمال و دستورالعمل‌های خاصی استفاده می‌شود مانند علایم + و - و عباراتی مانند Form، Control، و غیره.

د- ویژوال بیسیک تفاوتی بین کاراکترهای حرفی کوچک و بزرگ قائل نمی‌شود. نوع داده یک متغیر، تعیین‌کننده نوع مقداری است که در آن ذخیره می‌شود و میزان حافظه‌ای را که برای آن اختصاص داده می‌شود و نحوه ذخیره‌شدن آن را در حافظه معین می‌کند. میزان حافظه اشغالی برای ذخیره‌سازی یک اسم با میزان حافظه اشغالی توسط یک عدد صحیح یا اعشاری متفاوت است.

به عنوان مثال یک متغیر از نوع عدد صحیح (Integer) ۲ بایت فضا اشغال می‌کند، در صورتی که یک متغیر از نوع عدد اعشاری (Single) ۴ بایت فضا نیاز دارد. بنابراین در زمان تعریف یک متغیر لازم است تا نوع داده‌ای که در آن ذخیره خواهد شد، معین شود تا ویژوال بیسیک با توجه به آن، مقدار حافظه لازم را در اختیار متغیر قرار دهد.

انواع داده‌هایی که در ویژوال بیسیک ۶ قابل استفاده هستند به همراه میزان فضای مورد نیاز و محدوده مقادیر در جدول ۱-۴ ارایه شده‌اند.

جدول ۱-۴ انواع داده‌ها در ویژوال بیسیک

نوع داده	نوع مقدار	میزان حافظه اشغالی	محدوده مقادیر	پیشوند مناسب
Byte	اعداد صحیح	۱ بایت	صفر تا ۲۵۵	byt
Integer	اعداد صحیح	۲ بایت	۳۲۷۶۸- تا ۳۲۷۶۷+	int
Long	اعداد صحیح	۴ بایت	$\pm 2 \times 10^9$	lng
Single	اعداد اعشاری	۴ بایت	$\pm 3 \times 10^{38}$ تا $\pm 1 \times 10^{-45}$	sng
Double	اعداد اعشاری	۸ بایت	$\pm 5 \times 10^{-324}$ تا $\pm 1 / 8 \times 10^{308}$	dbl
Currency	اعداد اعشاری	۸ بایت	$\pm 9 \times 10^{14}$	cur
(با طول ثابت) String	متن (رشته)	یک بایت برای هر کاراکتر	کاراکتر ۶۵۴۰۰	str
(با طول متغیر) String	متن (رشته)	یک بایت برای هر کاراکتر	2×10^9 کاراکتر	str
Boolean	منطقی	۲ بایت	True یا False	bln
Date	تاریخ و ساعت	۸ بایت	۱/۱۰۰ / ۱ / ۱۱ الی ۱۲/۳۱/۹۹۹۹	dem
Variant	هر یک از انواع فوق	با توجه به نوع داده متغیر است.	_____	vnt


ویژوال بیسیک از انواع داده‌های مختلفی پشتیبانی می‌کند که می‌توانند نیازهای مختلف شما را در زمان برنامه‌نویسی پوشش دهند و با استفاده از انواع داده‌های ارایه شده در جدول ۱-۵ انواع متفاوتی از متغیرهای مورد نیاز را ایجاد کنند. در این جا به توضیح انواع آن‌ها می‌پردازیم.

متغیرهای عددی

ویژوال بیسیک از چند نوع داده عددی مختلف پشتیبانی می‌کند؛ داده صحیح، اعشاری و پولی. داده صحیح شامل انواع Integer، Long و Byte می‌باشد، با انواع Integer و Long می‌توانید متغیرهایی ایجاد کنید که در محدوده اعداد صحیح باشند، مقادیری مانند ۲۲۰، ۲۰۰۵، ۷۸- و غیره. اما نوع Byte در مواردی به کار می‌رود که فقط از اعداد صحیح مثبت در محدوده صفر تا ۲۵۵ استفاده می‌کنیم. انواع داده اعشاری نیز شامل انواع Single، Double و Currency هستند و توانایی ایجاد متغیرهایی را دارند که می‌توانند هرگونه عدد اعم از صحیح، اعشاری، مثبت و منفی مانند ۱۴۲-، ۱۷/۷۵- یا ۱ / ۰۰۲۵ / ۴۸۹۰ را در خود جای دهند. در صورتی که نیاز به نگهداری داده‌هایی از نوع پولی داشته باشید می‌توانید از نوع داده Currency استفاده کنید. به عنوان مثال به این تعاریف توجه کنید:

Dim	intcounter	As	Integer
Dim	dbltotal	As	Double
Dim	sngdistance	As	Single, intgrade As Integer
Dim	bytage	As	Byte, cursalary As Currency

همان‌طور که مشاهده می‌کنید متغیر intcounter از نوع اعداد صحیح و dbltotal از نوع اعشاری تعریف شده‌اند. به علاوه متغیرهای sngdistance و intgrade به‌طور هم‌زمان با استفاده از یک دستور Dim تعریف شده‌اند. همین‌طور متغیرهای bytage و cursalary نیز به ترتیب از نوع Byte و پولی معرفی شده‌اند.

 **نکته** استفاده از پیشوندهای مناسب با نوع داده در نام متغیر، شناسایی آسان متغیرها و در نتیجه خوانایی کدها را در پی خواهد داشت.

متغیرهای منطقی

این نوع متغیرها توانایی دریافت مقادیر درست (True) یا نادرست (False)، بله (Yes) یا خیر (No) را دارا می‌باشند، از این نوع متغیرها معمولاً زمانی استفاده می‌شود که داده‌های دو حالتی مانند زن یا مرد، مردود یا قبول، بیمار یا سالم و نظایر آن‌ها مورد نظر باشند به عنوان مثال برای ذخیره‌سازی وضعیت قبولی یک دانش‌آموز در مدرسه می‌توان متغیر `blnresult` را به این شکل تعریف کرد:

```
Dim blnresult As Boolean
```

متغیرهای تاریخ و زمان (Date & Time)

با استفاده از این نوع متغیرها می‌توانید مقادیر ساعت و تاریخ یا ترکیبی از این دو را ذخیره کنید. به عنوان مثال برای ذخیره‌سازی تاریخ تولد افراد می‌توانید متغیر `dtmb_Date` را به این صورت تعریف کنید:

```
Dim dtmb_Date As Date
```

متغیرهای رشته‌ای

اگر بخواهید مقادیر غیر عددی مانند متون و کاراکترها را ذخیره کنید، می‌توانید متغیرهای خود را از نوع داده رشته‌ای تعریف کنید. به عنوان مثال این نوع داده را می‌توان برای ذخیره‌سازی نام و نام خانوادگی اشخاص، نام شهر، آدرس یک محل و غیره به کار برد. متغیر رشته‌ای با طول ثابت، توانایی ذخیره‌سازی داده‌های رشته‌ای را با توجه به تعداد کاراکترهای تعیین شده دارد و برای تعریف آن از این الگو استفاده می‌شود:

```
Dim نام متغیر رشته‌ای As String*n
```

`n` عدد صحیح و مثبتی است که تعداد کاراکترها را در متغیر رشته‌ای تعیین می‌کند. به عنوان مثال برای ذخیره‌سازی نام افراد با حداکثر طول نام ۲۵ کاراکتر، از این دستور استفاده می‌شود:

```
Dim strname As String*25
```

متغیر رشته‌ای با طول متغیر، توانایی نگهداری یک تا ۲ میلیارد کاراکتر را دارد و در هنگام ذخیره‌سازی اطلاعات، فضای لازم را با توجه به طول رشته در اختیار متغیر قرار می‌دهد. برای تعریف این گونه متغیرها از این الگو استفاده می‌شود:

```
Dim نام متغیر رشته‌ای As String
```

به عنوان مثال برای ذخیره‌سازی اسامی افراد که حداکثر طول نام آن‌ها مشخص نیست، از این دستور استفاده می‌شود:

Dim strname As String

متغیرهای ترکیبی (Variant)

اگر به تغییری احتیاج دارید که نوع داده‌هایی که در آن ذخیره می‌شوند قابل پیش‌بینی نبوده و انواع متفاوتی را دربرمی‌گیرد، از این نوع متغیر استفاده کنید. این نوع متغیر تمام انواع داده‌های قبلی را دربرمی‌گیرد و می‌تواند مقادیر عددی، رشته‌ای، منطقی و تاریخی را بدون آن‌که نیازی به تبدیل آن‌ها باشد در خود جای دهد.

نکته در صورتی که نوع یک متغیر تعیین نشود، نوع متغیر به طور خودکار از نوع Variant در نظر گرفته خواهد شد.

تمرین:



کدام یک از این تعاریف اشتباه است:

Dim 1Age As Integer

Dim snggrade 1 As Single,n-no As Double

Dim sngaverage As Single,intno As Integer

جدول ۲-۴

کاراکتر	نوع داده
%	Integer
&	Long
!	Single
#	Double
@	Currency
\$	String

علاوه بر روش‌هایی که تاکنون برای تعریف انواع متغیرها گفته شد، می‌توانید با استفاده از بعضی کاراکترها نوع متغیر را به طور ضمنی تعریف کنید. به عبارت دیگر با اضافه کردن کاراکترهای رایج شده در جدول ۲-۴ به انتهای نام متغیر، نوع آن را معین کنید. از این روش فقط می‌توانید برای داده‌های عددی و رشته‌ای استفاده کنید.

به عنوان مثال به موارد ارایه شده در جدول ۳-۴ توجه کنید:

جدول ۳-۴

تعریف متغیرها به صورت ضمنی	تعریف متغیرها با دستور Dim
Intage %	Dim intage As Integer
Lngspeed&	Dim lngspeed As Long
sngaverage!	Dim sngaverage As Single
dblgrade#	Dim dblgrade As Double
cursalary@	Dim cursalary As Currency
strname\$	Dim strname As string

۲-۱-۴ نحوه مقداردهی به متغیرها

پس از تعریف یک متغیر بهتر است مقداری را در آن ذخیره کنیم. به این کار انتساب مقادیر یا مقداردهی به متغیر گفته می‌شود. برای انجام این کار می‌توانید از این الگو استفاده کنید:

Let نام متغیر = expression

البته ذکر کلمه کلیدی Let اختیاری است و معمولاً از آن استفاده نمی‌شود. Expression عبارتی است که می‌تواند یکی از انواع داده‌های گفته شده، فرمول و عبارات محاسباتی، خصوصیت یک شیء مانند فرم یا کنترل و غیره باشد. مشاهده می‌کنید که دستور انتساب شبیه به مقداردهی به خصوصیات اشیا مانند فرم‌ها و کنترل است. به عنوان مثال به مواردی که در جدول ۴-۴ ارایه شده است، توجه کنید.

جدول ۴-۴

دستور انتساب	نوع عبارت
Strname="Ali"	رشته‌ای
int count=-4	عددی
Sngave=snglength/3	عبارت ریاضی
dtmb_Date=#Dec-4-90#	تاریخ
blntest=True	منطقی
dblsum=inta+intb+intc	عبارت ریاضی
dtmtime= # 18:05:00 PM#	ساعت
dtmzone= #8/10/2004 5:12:17 AM#	تاریخ و ساعت

نکته در مقداردهی به متغیرهای منطقی می‌توانید از مقادیر یک و صفر به جای True و False استفاده کنید.

متغیرهای منطقی در هنگام تعریف به طور خودکار مقدار False، متغیرهای عددی مقدار صفر و متغیرهای رشته‌ای به صورت یک رشته خالی ("") مقداردهی اولیه می‌شوند. مقداردهی متغیرهای تاریخ و ساعت در هنگام تعریف با توجه به تنظیمات تاریخ و ساعت که به وسیله برنامه Regional and Language Options در بخش Control Panel ویندوز انجام می‌شود به دو روش صورت می‌گیرد. اگر ساعت روی حالت ۲۴ ساعته تنظیم شود مقدار اولیه به صورت ۰۰:۰۰:۰۰ یا ۰:۰۰:۰۰ خواهد بود اما وقتی ساعت روی حالت ۱۲ ساعته تنظیم شود مقدار اولیه به صورت ۱۲:۰۰:۰۰ AM می‌باشد.

مثال ۱: یک پروژه مطابق شکل ۴-۱ طراحی کنید که اطلاعات مربوط به یک کارمند شامل نام، نام خانوادگی و میزان حقوق وی را دریافت کرده، سپس با کلیک روی دکمه Calculate میزان حقوق خالص وی را پس از کسر مالیات و بیمه بر اساس فرمول‌های زیر محاسبه کرده و نمایش دهد؛ هم‌چنین با کلیک روی دکمه New، برنامه برای دریافت اطلاعات کارمند جدید آماده شود.

۳ درصد حقوق = بیمه

۵ درصد حقوق = مالیات



The image shows a window titled "Salary" with a standard Windows interface. It contains four input fields: "Name", "Family", "Salary", and "Pension". The "Pension" field has the value "0" displayed. At the bottom of the window, there are three buttons: "Calculate", "Item", and "Exit".

شکل ۴-۱

۱- برنامه ویژوال بیسیک را اجرا کنید و یک پروژه از نوع Standard EXE شامل یک فرم و کنترل‌های کادرمتن و برچسب را مطابق شکل ۴-۱ و جدول‌های ۴-۵، ۴-۶ و ۴-۷ طراحی کنید.

جدول ۴-۵ خصوصیات فرم


مقدار	خصوصیت
frmsal	Name
Salary	Caption

جدول ۴-۶ خصوصیات کنترل‌ها

کنترل خصوصیت	Label	Label	Label	Label	Label
Name	lblname	lblfam	lblsal	lblp	lblpay
Caption	Name:	Family:	Salary:	Payment:	o

جدول ۴-۷ خصوصیات کنترل‌ها

کنترل خصوصیت	TextBox	TextBox	TextBox	Command Button	Command Button	Command Button
Name	txtname	txtfam	txtsal	cmdcalculate	cmdnew	cmdexit
Caption	--	--	--	&Calculate	&New	&Exit

۲- در این مرحله لازم است کنترل‌های دکمه فرمان را روی فرم قرار دهید. به این منظور در جعبه ابزار روی آیکن  (Command Button) دابل کلیک کنید پس از ایجاد دکمه‌های فرمان خصوصیت‌های آن‌ها را مطابق جدول ۴-۷ تنظیم نمایید.

۳- برای تعیین کلید ایجاد دسترسی سریع (Hot Key) در کنترل‌ها می‌توان در خصوصیت Caption و قبل از حرف موردنظر از کاراکتر & استفاده کرد بنابراین لازم است قبل از حروف C, N و E در سه دکمه فرمان ایجاد شده کاراکتر & را تایپ کنید. پس از تعیین کلید دسترسی سریع، بلافاصله در پنجره طراحی فرم زیر حروف مربوط یک خط مشاهده می‌شود.

۴- به پنجره ماژول فرم بروید و رویداد Click دکمه فرمان Calculate, New و Exit را به این صورت تنظیم کنید:

```
Private Sub cmdcalculate_Click()
```

```
Dim sngsal As Single
```

```
Dim sngins As Single
```

```
Dim sngtax As Single
```

```
Dim sngpay As Single
```

```
sngsal = Val(txtsal.Text)
```

```
sngins=sngsal *0.03
```

```
sngtax=sngsal * 0.05
```

```
sngpay= sngsal-sngins-sngtax
```

```
lblpay.Caption=sngpay
```

```
End Sub
```

در این رویداد ابتدا با استفاده از دستور Dim متغیرهای sngsal برای حقوق، sngins برای بیمه، sngtax برای مالیات و sngpay برای حقوق خالص از نوع اعشاری تعریف شده‌اند سپس مقدار حقوق که در خصوصیت Text کادر متن txtsal ذخیره شده است با استفاده از تابع Val به عدد تبدیل و در متغیر اعشاری sngsal ذخیره می‌شود تا بعداً برای محاسبه بیمه، مالیات و حقوق پرداختی از آن استفاده شود. تابع Val می‌تواند یک عبارت رشته‌ای را که شامل کاراکترهای رقمی باشد دریافت کرده و به نوع داده عددی تبدیل کند. در مرحله آخر نیز حقوق خالص وی در خصوصیت Caption کنترل برچسب lblpay ذخیره می‌شود تا روی فرم نمایش داده شود.

۵ - رویداد Click دکمه فرمان New را به این صورت تنظیم کنید:

```
Private Sub cmdnew_Click()
```

```
txtname.Text=""
```

```
txtfam.Text=""
```

```
txtsal.Text=""
```

```
lblpay.Caption = 0
```

```
txtname.setFocus
```

```
End Sub
```

در این رویداد برای آن‌که بتوان داده‌های قبلی را در کادرهای متن پاک کرده و امکان ورود داده‌های جدید به وجود آید ابتدا محتویات کادرهای متن پاک شده و سپس مقدار

صفر در کنترل برچسب نمایش داده می‌شود سپس با استفاده از متد SetFocus، فوکوس به اولین کادر متن منتقل می‌شود تا وقتی کاربر روی دکمه New کلیک کند بلافاصله بتواند نام کارمند را وارد نماید.

متد، یک یا مجموعه عملیاتی است که روی یک فرم یا کنترل انجام می‌شود و روی آن تأثیر می‌گذارد. برای استفاده از متدها می‌توانید از شکل کلی زیر استفاده کنید:

نام متد . نام کنترل

نکته: برای دسترسی به متدهای یک فرم در ماژول فرم مربوط به آن می‌توان از نام متد به تنهایی یا همراه با عبارت me. استفاده کرد.

فوکوس امکانی است که با استفاده از ماوس یا صفحه کلید به کاربر اجازه می‌دهد به فرم یا کنترل موردنظر خود دسترسی پیدا کند. در ویندوز چندین برنامه می‌توانند به طور هم‌زمان اجرا شوند، اما کاربر فقط می‌تواند با یک برنامه کار کند یعنی برنامه‌ای که نوار عنوان پنجره آن فعال است یا به عبارت دیگر فوکوس را در اختیار دارد. در یک پنجره و فرم با چند کنترل نیز کاربر می‌تواند در هر لحظه با یک کنترل کار کند در این حالت کنترلی که فوکوس را در اختیار دارد، قابل استفاده است. برای انتقال فوکوس بین فرم‌ها می‌توانید از کلیک روی فرم مورد نظر یا کلید ترکیبی Alt+Tab و برای انتقال فوکوس از یک کنترل به کنترل دیگر از عمل کلیک یا کلید Tab استفاده کنید.

متد SetFocus یکی از متدهای مشترک بین کنترل‌ها و فرم‌هاست و اجرای آن فوکوس را به فرم یا کنترل مربوطه منتقل می‌کند.

رویداد Click دکمه فرمان Exit را به این صورت تنظیم کنید:

```
Private Sub cmdexit_Click()
```

```
Unload Me
```

```
End Sub
```

در این رویداد با استفاده از دستور Unload Me فرم برنامه از حافظه خارج می‌شود در نتیجه برنامه خاتمه می‌یابد.

۶ - پروژه و فرم را با نام salary ذخیره کنید، سپس پروژه را اجرا کرده و اطلاعات

دلخواهی را در کادرهای متن فرم تایپ کنید.

۷ - روی دکمه Calculate کلیک کنید تا نتیجه محاسبات یا به عبارت دیگر میزان حقوق خالص کارمند مربوطه نمایش داده شود.

۸ - روی دکمه New کلیک کنید و این بار مشخصات فرد دیگری را تایپ کرده و میزان حقوق وی را محاسبه نمایید.

۹ - با کلیک روی دکمه Exit به محیط طراحی بازگردید.



تمرین:

پروژه salary را به شکلی تغییر دهید تا هزینه بیمه، مالیات و حقوق دریافتی هر کارمند به صورت جداگانه، روی فرم قابل مشاهده باشد.

۲-۴ نحوه انجام عملیات ریاضی در ویژوال بیسیک

در تمام زبان‌های برنامه‌نویسی برای تبدیل انواع فرمول‌ها و انجام محاسبات، امکاناتی در نظر گرفته می‌شود. در زبان برنامه‌نویسی ویژوال بیسیک نیز برای این کار، از عملگرهای ریاضی برای انجام اعمالی چون جمع، تفریق، به توان‌رسانی و غیره استفاده می‌شود. با استفاده از این عملگرها انجام بخش عمده‌ای از عملیات محاسباتی امکان‌پذیر می‌شود.



مثال ۲: یک ماشین حساب ساده مطابق شکل ۲-۴ و جداول ۸-۴ الی ۱۰-۴ طراحی کنید که توانایی انجام عملیات ریاضی ساده را داشته باشد. به این منظور مراحل بعد را به ترتیب انجام دهید:

جدول ۸-۴ خصوصیات فرم

مقدار	خصوصیت
frmcalc	Name
۳۷۲۰	Height
۳۰۰	Left
۳۰۰	Top
۴۹۰۵	Width
Calculator	Caption



شکل ۲-۴

جدول ۹-۴ خصوصیات کنترل‌ها

کنترل خصوصیت	Label	Label	Label	Label	TextBox
Name	lblnum۱	lblnum۲	lblr	lblresult	txtnum۱
Caption	Number ۱ :	Number ۲ :	Result :	°	--

جدول ۱۰-۴ خصوصیات کنترل‌ها

کنترل خصوصیت	TextBox	Command Button	Command Button	Command Button	Command Button
Name	txtnum۲	cmdadd	cmddiv	cmdminus	Cmdmulti
Caption	--	+	/	-	×

۱ - برنامه ویژوال بیسیک را اجرا کرده و کنترل‌های لازم را مطابق شکل ۵-۴ روی فرم ایجاد کنید.

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

```
Private Sub cmdadd_Click()
```

```
    lblresult.Caption=Val(txtnum1.Text) + Val(txtnum2.Text)
```

```
End Sub
```

۳ - فرم و پروژه را با نام Calculator ذخیره کنید، سپس پروژه را اجرا کرده و دو عدد در کادرهای متن بنویسید؛ سپس روی دکمه جمع کلیک کنید و نتیجه را بررسی کنید، همان‌طور که می‌بینید حاصل جمع اعداد تایپ شده در بخش Result نمایش داده می‌شود.

همان‌طور که در رویداد Click دکمه جمع مشاهده کردید برای جمع دو عدد که در کادرهای متن تایپ می‌شوند از کاراکتر + استفاده می‌شود. در زبان ویژوال بیسیک برای انجام عملیات جمع از این کاراکتر استفاده می‌شود. به علاوه مقادیری که در کادرهای متن تایپ می‌شوند ماهیت غیر عددی دارند و جمع آن‌ها به صورت عادی امکان‌پذیر نیست؛

بنابراین با استفاده از تابع Val رشته عددی تایپ شده در کادرهای متن را به نوع عددی تبدیل کرده و سپس آن‌ها را با هم جمع و در خصوصیت Caption کنترل برچسب ذخیره می‌کنید تا نتیجه محاسبه نمایش داده شود.

۴ - در این مرحله رویدادهای Click دکمه‌های تفریق، ضرب و تقسیم را به این صورت و مشابه دکمه جمع تنظیم کنید:

```
Private Sub cmdminus_Click()
```

```
    lblresult.Caption=Val(txtnum1.Text)-Val(txtnum2.Text)
```

```
End Sub
```

```
Private Sub cmdmulti_Click()
```

```
    lblresult.Caption=Val(txtnum1.Text)*Val(txtnum2.Text)
```

```
End Sub
```

```
Private Sub cmddiv_Click()
```

```
    lblresult.Caption=Val(txtnum1.Text)/Val(txtnum2.Text)
```

```
End Sub
```

۵ - پروژه را اجرا کرده و عملکرد برنامه را در رابطه با چهار عمل اصلی بررسی کنید.

۶ - پنجره برنامه را ببندید و سپس تغییرات اعمال شده را ذخیره کنید.

۳-۴ نحوه تعریف و استفاده از ثابت‌ها در ویژوال بیسیک

قبلاً روش‌های مختلفی را برای ذخیره‌سازی اطلاعات فراگرفته‌اید. روش‌های دیگری نیز وجود دارد که یکی از آن‌ها استفاده از ثابت‌هاست (Constant). علاوه بر ثابت‌هایی که به طور آماده در ویژوال بیسیک وجود دارند (برای مثال، می‌توان به ثابت‌هایی که برای تعیین رنگ‌ها استفاده می‌شوند، اشاره کرد). می‌توانید ثابت‌های مورد نیاز خود را نیز در برنامه تعریف کنید. تفاوتی که بین ثابت‌ها و متغیرها وجود دارد این است که مقدار یک ثابت پس از تعریف قابل تغییر نیست، در صورتی که مقدار متغیرها را می‌توانید بارها در هنگام اجرای برنامه تغییر دهید.

برای استفاده از ثابت‌ها دلایلی نیز وجود دارد به عنوان نمونه گاهی اوقات مقادیر عددی و

رشته‌ای خاصی به طور مکرر در برنامه استفاده می‌شوند که ذکر آن‌ها توأم با صرف زمان و ایجاد اشتباه در سطح برنامه خواهد شد، اما با استفاده از یک ثابت می‌توان مقدار مورد نظر را در آن ذخیره کرد و در هر جایی از برنامه نام ثابت را به جای مقدار مربوطه به کار برد. در این صورت برنامه خواناتر شده و ایجاد تغییرات در برنامه در آینده سریع‌تر و با اشتباه کمتری همراه خواهد بود.

برای تعریف یک ثابت به این صورت عمل کنید:

مقدار = نوع ثابت As نام ثابت Const

تعریف ثابت‌ها و قوانین نام‌گذاری آن‌ها شبیه به متغیرهاست. نوع ثابت می‌تواند از انواع داده‌ها در ویژوال بیسیک باشد که در جدول ۱-۴ ارایه شده‌اند. پس از تعیین نوع داده برای ثابت و قراردادن علامت مساوی، مقدار موردنظر برای ذخیره‌سازی در ثابت قرار می‌گیرد. مقادیر رشته‌ای را داخل علامت (") و مقادیر تاریخ و ساعت را داخل علامت (#) قرار دهید. به عنوان مثال به این نمونه‌ها توجه کنید:

Const Conlength As Integer = 20

Const Conname As String = "Ali"

Const ConcreteDate As Date = #12/5/98#



مثال ۳: یک پروژه طراحی کنید که کاربر توانایی محاسبه حجم یک استوانه دلخواه را داشته باشد.

۱ - یک پروژه از نوع Standard EXE با یک فرم و کنترل‌های مربوطه مطابق شکل ۳-۴ و جداول ۱۱-۴ الی ۱۳-۴ ایجاد کنید.

جدول ۱۱-۴ خصوصیات فرم

خصوصیت	مقدار
Name	frm cyl
Caption	Cylinder



شکل ۳-۴

جدول ۱۲-۴ خصوصیات کنترل‌ها

کنترل خصوصیت	Label	Label	Label	Label
Name	lblr	lblh	lblc	lblresult
Caption	Radius :	Height :	Capacity :	o

جدول ۱۳-۴ خصوصیات کنترل‌ها

کنترل خصوصیت	TextBox	TextBox	Command Button
Name	txtr	txth	cmdcalc
Caption	--	--	Calculate&

۲ - خصوصیت Border Style فرم را روی مقدار Fixed Dialog 3- تنظیم کنید. به وسیله این خصوصیت می‌توان کادر دور فرم، دکمه‌های کنترلی منوی کنترل و قابلیت تغییر اندازه را در فرم تعیین کرد. این خصوصیت می‌تواند یکی از موارد ارائه شده در جدول ۱۴-۴ را کسب نماید.

جدول ۱۴-۴ خصوصیت Border Style

مقدار خصوصیت	توضیح
0-None	فرم بدون کادر، نوارعنوان، منوی کنترل، دکمه‌های کنترلی و با اندازه ثابت
1- Fixed Single	فرم با کادر، نوارعنوان، دکمه‌های کنترلی و قابلیت تغییر اندازه با دکمه‌های Maximize و Minimize
2- Sizable	فرم با کادر، نوارعنوان، دکمه‌های کنترلی و قابلیت تغییر اندازه
3- Fixed Dialog	فرم با کادر، منوی کنترل، نوارعنوان، بدون دکمه‌های کنترلی Minimize و Maximize و با اندازه ثابت. در ضمن در هنگام اجرای برنامه آیکن فرم در Taskbar نمایش داده نمی‌شود.
4-Fixed Tool Window	فرم با کادر، نوارعنوان و دکمه Close (با اندازه کوچک) و بدون قابلیت تغییر اندازه، به علاوه در هنگام اجرای برنامه آیکن فرم در Taskbar نمایش داده نمی‌شود.
5- Sizable Tool Window	فرم با کادر، نوارعنوان و دکمه Close (با اندازه کوچک) و قابلیت تغییر اندازه، به علاوه در هنگام اجرای برنامه آیکن فرم در Taskbar نمایش داده نمی‌شود.



نکته منوی کنترل در نوارعنوان فرم‌ها در سمت چپ عنوان فرم به صورت یک آیکن کوچک نمایش داده می‌شوند و شامل گزینه‌هایی مانند Move، Size، Maximize، Restore و Close است که برای مدیریت فرم‌ها استفاده می‌شود.

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

```
Private Sub cmdcalc_Click()
```

```
Const pi As Single=3.14159265358979
```

```
lblresult.Caption=pi*Val(txtx.Text)^2*Val(txth.Text)
```

```
End Sub
```

۴- پروژه و فرم را با نام cylinder ذخیره کنید، سپس پروژه را اجرا کرده و برنامه را به ازای چند مقدار آزمایش کنید.

۵- به اجرای پروژه خاتمه دهید و به پنجره ویژوال بیسیک بازگردید.
برای استفاده از مقدار عدد π از ثابت pi استفاده شده است که موجب دقیق‌تر شدن محاسبات و تایپ راحت‌تر دستورات خواهد شد.

۴-۴ متغیرهای ایستا، محلی و عمومی در ویژوال بیسیک

در این جا لازم است انواع متغیرها را از نظر میدان دید (Scope) و شناسایی آن‌ها در بخش‌های مختلف برنامه مورد بررسی قرار دهید. میدان دید یک متغیر معین می‌کند که بعد از تعریف یک متغیر چه بخش‌هایی از برنامه می‌توانند به آن دسترسی داشته باشند و از آن استفاده کنند.

وقتی متغیری در یک رویه رویداد تعریف می‌شود فقط دستورات داخل آن رویه رویداد می‌توانند به آن دسترسی داشته باشند چنین متغیرهایی را متغیرهای محلی (Local) در سطح رویه می‌گویند. در این صورت با خروج از رویه رویداد، متغیر در حافظه پاک شده و مقدار ذخیره شده در آن از بین می‌رود. برای تعریف این نوع از متغیرها می‌توانید از دستور Dim استفاده کنید.

نوع دیگری از متغیرهای محلی نیز وجود دارند که با استفاده از کلمه کلیدی Static به جای Dim تعریف می‌شوند، این نوع متغیرها تمام ویژگی‌های متغیرهای محلی را دارند

با این تفاوت که پس از خاتمه اجرای رویه‌ای که در آن تعریف شده‌اند، مقدار آن‌ها در حافظه از بین نمی‌رود و در اجرای بعدی رویه مربوطه می‌توانید از مقادیر قبلی متغیر استفاده کنید.

به عنوان مثال به این نمونه‌ها توجه کنید:

Dim intno As Integer

Static sngk As Single

اما گاهی اوقات لازم است تا یک متغیر در تمام رویه‌های یک ماژول فرم قابل استفاده باشد، برای این کار می‌توانید متغیر مورد نظر را در بخش تعاریف ماژول فرم و با استفاده از کلمه کلیدی Private تعریف کنید. در صورت تعریف یک متغیر به صورت فوق تمام رویه‌های موجود در همان ماژول می‌توانند از متغیر مربوطه استفاده کنند و تا زمانی که فرم از حافظه خارج نشده است، مقدار خود را حفظ نمایند. شکل کلی نحوه استفاده از این نوع متغیرها به این صورت است:

نوع داده As نام متغیر Private

به عنوان مثال در صورتی که متغیرهای زیر در بخش تعاریف ماژول فرم تعریف شوند در همه سطح ماژول فرم قابل شناسایی و استفاده می‌باشد.

Private strname As String

Dim snggrasde As Single


نکته به جای استفاده از کلمه کلیدی Private می‌توانید از کلمه کلیدی Dim نیز استفاده کنید.




۴-۵ عملگرهای رشته‌ای در ویژوال بیسیک

علاوه بر عملگرهای ریاضی، مقایسه‌ای و منطقی، عملگرهای دیگری نیز وجود دارند که توانایی انجام عملیات را روی رشته‌ها فراهم می‌کنند. یکی از این عملگرها، عملگر جمع (+) است این عملگر می‌تواند دو یا چند مقدار رشته‌ای را به یکدیگر اضافه کند به عنوان مثال عبارت "Basic" + "Visual"، رشته‌ای به صورت "VisualBasic" را به وجود می‌آورد. البته این عملگر را می‌توانید برای متغیرها و

خصوصیات از نوع رشته‌ای به کار ببرید.

 نکته به جای عملگر + می‌توانید از عملگر & نیز استفاده کنید.

 **مثال ۴:** پروژه‌ای طراحی کنید که هزینه رنگ‌آمیزی یک سالن یا اتاق را با دریافت ابعاد آن‌ها و با فرض این‌که هزینه هر مترمربع ۴۰۰۰۰ ریال باشد، محاسبه نماید. برای این کار عملیات زیر را به ترتیب انجام دهید:
یک پروژه از نوع Standard EXE ایجاد کنید که شامل یک فرم و کنترل‌های مربوطه مطابق شکل ۴-۴ و جداول ۴-۱۵ الی ۴-۱۸ باشد؛ سپس خصوصیت BorderStyle فرم را روی مقدار 3-Fixed Dialog تنظیم کنید.
سپس رویداد Click دکمه فرمان Calculate را به این صورت تنظیم کنید:

```
Private Sub cmdcalc_Click()  
    Dim dblarea As Double  
    Dim curpay As Currency  
    dblarea=2*Val(txtl.Text)*Val(txtrh.Text)+ 2*Val(txtrw.Text)*Val(txtrh.Text)+ _  
    Val(txtl.Text)*Val(txtrw.Text)  
    curpay = dblarea * 40000  
    lblar.Caption=Str(dblarea)+ "M2"  
    lblpay.Caption=Str(curpay)+"Rials"  
End Sub
```

در این رویداد پس از محاسبه مساحت سطح و هزینه رنگ‌آمیزی شده، مقادیر آن‌ها به ترتیب در متغیرهای dblarea و curpay ذخیره می‌شوند، سپس با استفاده از کنترل‌های برچسب مساحت سطح رنگ شده و کل هزینه برای پرداخت نمایش داده می‌شود. در دستوراتی که برای نمایش نتیجه محاسبات استفاده شده‌اند، ابتدا با استفاده از تابع Str مقادیر عددی به مقادیر رشته‌ای تبدیل می‌شوند، سپس با استفاده از عملگر + آن‌ها را به هم متصل کرده و در خصوصیت Caption کنترل‌های برچسب ذخیره می‌شوند تا در روی فرم نمایش داده شوند.



شکل ۴-۴

جدول ۴-۱۵ خصوصیات فرم

مقدار	خصوصیت
Name	frmpaint
Caption	Paint

جدول ۴-۱۶ خصوصیات کنترل‌ها

کنترل خصوصیت	Label	Label	Label
Name	lblrl	lblrw	lblrh
Caption	Room Length :	Room Width :	Room Height :

جدول ۴-۱۷ خصوصیات کنترل‌ها

کنترل خصوصیت	Label	Label	Label	Label	Command Button
Name	lbla	lblar	lblpay	lblpayr	cmdcalc
Caption	Area:	•	Payment :	•	&Calculate

جدول ۴-۱۸ خصوصیات کنترل‌ها

کنترل خصوصیت	TextBox	TextBox	TextBox
Name	txtrl	txtrw	txtrh

نکته در صورتی که خطوط مربوط به دستورات بیش از حد طولانی باشند می‌توانید آن‌ها را در چند خط تایپ کنید و آن‌ها را با کاراکتر زیرخط به هم مرتبط کنید.

۳ - پروژه و فرم را با نام Paint ذخیره و اجرا کنید، سپس اعداد ۴، ۲/۵ و ۳ متر را به ترتیب برای طول، عرض و ارتفاع وارد کنید و روی دکمه Calculate کلیک کرده و نتیجه محاسبات را بررسی کنید.

۴ - برنامه را برای مقادیر دیگری نیز آزمایش کنید، سپس از برنامه خارج شوید.

تمرین:



پروژه‌ای طراحی کنید که نام و نام خانوادگی هر فرد دلخواهی را دریافت کند سپس ترتیب آن دو را در کنار هم با یک کنترل برچسب نمایش دهد.

علاوه بر عملگر جمع، عملگر دیگری به نام Like نیز وجود دارد که توانایی انجام عملیات روی رشته‌ها را دارد. این عملگر یک رشته را با رشته دیگری مقایسه کرده و نتیجه را مشخص می‌کند. می‌توانید از این عملگر به صورت زیر استفاده کنید:

عبارت رشته‌ای دوم Like عبارت رشته‌ای اول

اگر عبارت رشته‌ای اول با عبارت رشته‌ای دوم مشابه باشد، مقدار True و اگر عبارت رشته‌ای اول با عبارت رشته‌ای دوم مشابه نباشد، مقدار False به دست می‌آید، به عنوان مثال مقایسه "ali" Like "alireza" نتیجه True و مقایسه "ali" Like "ali" نتیجه False در پی خواهد داشت. به علاوه با استفاده از کاراکترهای *، ؟، # و [] می‌توانید ترکیب‌های متنوعی از مقایسه‌ها را ایجاد کنید.

جدول ۱۹-۴

عبارت	
"aBBBa" Like "a#a"	True
"F" Like "[A-Z]"	True
"F" Like "[!A-Z]"	False
"a۲a" Like "a#a"	True
"*BAT۱۲۳khg" Like "B?T"	True
"*CAT۱۲۳khg" Like "B?T"	False
"aM۵b" Like "a[L - P]#[!c-e]"	True

کاراکتر * بیانگر گروهی از کاراکترها، کاراکتر ? بیانگر هر کاراکتر دلخواه، کاراکتر # بیانگر هر کاراکتر رقمی و کاراکترهای [] برای ایجاد بازه‌ای از کاراکترها در مقایسه به کار می‌روند. برای مشخص شدن بهتر مطلب به مثال‌هایی که در جدول ۱۹-۴ ارائه شده‌اند، توجه کنید.

همان‌طور که ملاحظه می‌کنید اولین مقایسه نتیجه True در بردارد، زیرا * می‌تواند جانشین گروهی از کاراکترها شود. در عبارت دوم حرف F در بازه حروف A تا Z قرار دارد، پس نتیجه True به دست می‌آید؛ اما در عبارت سوم از کاراکتر ! استفاده شده است و به این معنی است که F جزء حروف A تا Z نیست و نادرست است. در عبارت بعدی وجود کاراکتر # در بین حروف، پذیرش کاراکتر رقمی ۲ را در رشته "a۲a" توجیه می‌کند. عبارت‌های بعدی نیز به همین صورت قابل بررسی هستند و ترکیبی از حالت‌های گفته شده در آن‌ها به کار رفته است.

تمرین:



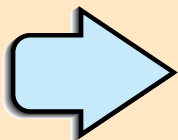
نتیجه این مقایسه‌ها را به دست آورید:

"Vb6" Like "?b#"

"Microsoft" Like "M*"

"Windows XP" Like "?i[a-z]*"

"Visual C++" Like "Visual ?[!A-Z]++"



Learn in English

Variables

In Visual Basic, you use variables to temporarily store values during the execution of an application. Variables have a name (the word you use to refer to the value the variable contains) and a data type (which determines the kind of data the variable can store).

Declaring Variables

To declare a variable is to tell the program about it in advance. You declare a variable with the Dim statement, supplying a name for the variable:

Dim variablename [**As** type]

واژه‌نامه

Constant	ثابت
Currency	مقادیر پولی
Declare	تعریف کردن
Determine	تعیین کردن
Execution	اجرا
Precedence	اولویت
Private	خصوصی
Public	عمومی
Store	ذخیره کردن
Supply	تأمین کردن، عرضه کردن
Temporary	موقت
Variable	متغیر
Variant	ترکیبی

خلاصه مطالب

- برای انجام عملیات ریاضی از عملگرهای ریاضی استفاده می‌شود.
- انواع داده‌ها در ویژوال بیسیک عبارتند از: داده‌های عددی، رشته‌ای، منطقی، پولی، تاریخ و ساعت و داده‌های ترکیبی.
- برای تعریف متغیر از دستور Dim استفاده می‌شود.
- در صورت استفاده از دستور Option Explicit، باید متغیر را قبل از استفاده از آن تعریف کرد.
- متغیرهایی که در یک رویداد تعریف می‌شوند فقط در همان رویداد قابل شناسایی و استفاده هستند به این متغیرها، متغیرهای محلی می‌گویند.
- متغیرهای خصوصی در تمام رویدادهای ماژول فرم قابل شناسایی و استفاده هستند و برای تعریف آن‌ها از بخش تعاریف استفاده می‌شود.
- برای محاسبه زمان و اجرای دستورات مورد نظر به‌طور خودکار و مکرر در فاصله‌های زمانی معین از کنترل زمان‌سنج استفاده می‌شود.
- عملگرهای رشته‌ای در ویژوال بیسیک عبارتند از: +، & و Like.
- شکل کلی نحوه استفاده از متدها به این صورت است:

نام متد. نام شیء

آزمون نظری

- ۱ - کدام نوع از انواع داده برای تعریف یک متغیر از نوع پولی مناسب‌تر است؟
 الف - Single ب - Double ج - Currency د - Boolean
- ۲ - کدام عملگر ریاضی نسبت به سایرین از اولویت پایین‌تری برخوردار است؟
 الف - جمع ب - Mod ج - توان د - ضرب
- ۳ - کدام گزینه برای تعریف یک متغیر از نوع اعداد صحیح بلند درست است؟
 الف - % ب - & ج - ! د - #
- ۴ - کدام عملگر رشته‌ای مشابه عملگر جمع در رشته‌هاست؟
 الف - % ب - & ج - Like د - #
- ۵ - اگر متغیری در بخش تعاریف مازول فرم با دستور Dim معرفی شود متغیر
 نامیده می‌شود.
- الف - عمومی ب - محلی ج - خصوصی د - ایستا
- ۶ - کدام نوع داده برای تعریف اعداد مثبت کوچک‌تر از ۲۵۵ مناسب است؟
 الف - String ب - Integer ج - Byte د - Long
- ۷ - حاصل عبارت "Computer2004" Like "C?m?u*####" چیست؟
 الف - True ب - False
- ج - عبارت اشتباه است. د - در هنگام اجرا پیام خطا نمایش داده می‌شود.
- ۸ - متغیرهای خصوصی یا Private در چه محدوده‌ای قابل شناسایی هستند؟
 الف - فرم‌ها
 ب - مازول فرمی که در آن تعریف شده‌اند.
 ج - تمام پروژه
 د - رویدادی که در آن تعریف شده‌اند.
- ۹ - The variable can be declared with statement in within a procedure.
 a- private b- public c- dim d- as
- ۱۰ - مفهوم متغیر و نوع داده را توضیح دهید و نحوه تعریف متغیرها را بیان کنید.
- ۱۱ - تفاوت بین متغیرهای عمومی و محلی را شرح دهید.
- ۱۲ - انواع عملگرهای ریاضی و عملگرهای رشته‌ای را توضیح دهید.

آزمون عملی

۱ - پروژه‌ای طراحی کنید که با توجه به نوع سپرده‌گذاری در یک بانک، میزان بهره سالیانه را مطابق این جدول محاسبه کرده و نمایش دهد.

نوع سپرده	میزان بهره
قرض الحسنه	—
کوتاه مدت	۸ درصد
بلند مدت ۳ ساله	۱۲ درصد
بلند مدت ۶ ساله	۱۵ درصد

۲ - یک پروژه از نوع Standard EXE طراحی کنید که شامل فرم به صورتی که در شکل نشان داده شده و با این شرایط باشد.

الف- نام کاربر با اندازه ۲۴ و با رنگ سیاه و ضخیم در روی فرم نمایش داده شود.

ب- کاربر بتواند با کلیک روی دکمه‌های Red، Green، و Blue رنگ قلم را در نام کاربر روی رنگ قرمز، سبز یا آبی تنظیم کند.

ج- در صورت کلیک روی دکمه Hide نام کاربر مخفی شود؛ به علاوه دکمه Hide غیرفعال و دکمه Show فعال شود.

د- در صورت کلیک روی دکمه Show نام کاربر نمایش داده شود؛ به علاوه دکمه

Show غیرفعال و دکمه Hide فعال شود.

هـ - فرم و پروژه را با نام Show ذخیره کنید.



۳- عبارات زیر را به زبان ویژوال بیسیک تبدیل کنید.

الف- $\frac{\frac{a-c}{b-d}}{5-\frac{e}{f}}$

ب- $(3x + y^2) \times (6z - 12)$

ج- $4x^2 - 5x + 3$

د- $\frac{\left(\frac{15}{7y}\right)}{9z - 18x}$

۴- عبارات زیر را به صورت ریاضی بنویسید.

الف- $1/2 * x + 14 * z^3$

ب- $x \setminus y * z / w^u$

ج- $243 - a / b^2 + 3 * z$

۵- پس از انجام دستورات زیر حاصل متغیر x چیست؟

Dim str1 As String , str2 As String

Dim x As Integer

str 1 ="34"

str2="Ali"

x=Val (str1)+Val (str2)

۶- با توجه به تمرین ۵، حاصل عبارات $str1=str1+str2$ و $str1=Str(x)+str2$ چیست؟

