

ثابتها متغیرهایی هستند که برای یکبار در برنامه تعریف می شوند و مقادیری که به آنها نسبت داده می شود مجدداً قابل تغییر نمی باشند و هر گاه نیاز به تغییر مقادیر آنها بود می بایست اصل (Source) برنامه تغییر یابد.

بعنوان مثال می خواهید از عدد π (۳/۱۴) در برنامه استفاده کنید و این مقدار نبایست در هیچ جا قابل تغییر باشد به صورت زیر عمل کنید:

Const Pi= 3.14

همانطور که ملاحظه می کنید برای تعریف ثابتها از کلمه کلیدی Const استفاده می شود و بعد از آن نام متغیر به همراه مقداری که باید به آن تعلق گیرد آورده می شود.

حال اگر در هر جای برنامه نیازی به مقدار مذکور باشد کافی است از متغیر Pi استفاده گردد.

مزایای استفاده از ثابتها یکی استفاده از حافظه کامپیوتر به دلیل تعریف متغیر در یک حافظه بوده و دیگر اینکه در صورت نیاز به تغییر مقدار آن فقط باید به یک نقطه از برنامه که ثابت در آن تعریف شده است دست زد.

۳-۴- عملگرها Operators

عملگرها دستوراتی هستند که بر روی مقادیر ثابت یا متغیرهای موجود در یک عبارت عمل می کنند. به عنوان مثال فرض کنید می خواهید دو رشته حرفی را به هم متصل و حاصل آن را در یک متغیر قرار دهید. دستورات مربوط به این مثال عبارتند از:

Dim S As String

S = "Visual" & "Basic"

عملگر & باعث می شود تا دو واژه Visual و Basic به یکدیگر متصل شده و کلمه

VisualBasic در متغیر S قرار گیرد.

زبان V.B عملگرهای زیادی را پشتیبانی می کند که در جدول زیر این عملگرها و کاری را که انجام می دهند آورده شده است.

شرح	عملگر
-----	-------

&	اتصال دو رشته به یکدیگر
*	ضرب دو عدد
+	جمع دو عدد
-	تفاضل دو عدد
/	تقسیم دو عدد بر یکدیگر و برگرداندن حاصل به صورت ممیز شناور (floating point)
	تقسیم دو عدد بر یکدیگر و برگرداندن خارج قسمت صحیح تقسیم
	توان رساندن عدد اول به اندازه عدد دوم
Address of	بدست آوردن آدرس یک روال (Procedure)
AND	ارزیابی True در صورتی که دو عبارت صحیح باشند
Comparison	جهت مقایسه دو عبارت با یکدیگر استفاده می شود (<,<=,>,>=,=,<>,IS,Like)
Eqv	در صورتی که هر دو عبارت یکسان باشند مقدار True و در غیر این صورت False برمی گرداند
Imp	مفهوم منطقی دو عبارت را اجرا می کند
Is	جهت مقایسه دو اشاره گر به Object
Link	به منظور یافتن الگویی تعیین شده در یک رشته
Mod	باقیمانده تقسیم دو عدد را برمی گرداند
Not	منفی مقدار یک عبارت را می دهد
Or	اگر حداقل یکی از چندین عبارت موجود صحیح باشد مقدار True را برمی گرداند
Xor	اگر فقط یکی از عبارات صحیح باشد مقدار True را برمی گرداند و در غیر این صورت مقدار False را بر می گرداند

۳-۵- زیرروالها Subroutines

زیر روال عبارتست از گروهی از دستورات برنامه که تحت یک نام خاص نوشته شده‌اند و در زمان مناسب با استفاده از نامشان فراخوانده و اجرا می‌شوند. مثال:

Public Sub Test-Message()

MsgBox "This is for test"

End sub

در مثال فوق یک زیرروال به نام Test-Message تعریف شده که با فراخوانی آن پیغام This is for test بر روی صفحه نمایان می‌گردد. اکنون به تشریح چگونگی نوشتن یک زیرروال می‌پردازیم.

اگر بخواهید زیرروال نوشته شده در همه قسمت‌های مختلف برنامه شناخته شده و قابل فراخوانی باشد واژه کلیدی Public را در ابتدای آن قرار دهید. واژه کلیدی بعدی Sub است که مشخص کننده تعریف زیرروال می‌باشد و بعد از آن نام زیرروال آورده می‌شود. در صورتی که زیرروال هیچ پارامتری نداشته باشد مانند مثال فوق یک جفت پرانتز به صورت () در مقابل نام آن می‌گذارید و سپس دستورات مربوط به آن را می‌نویسید.

در مثال فوق یکی از Object های موجود در V.B به نام MsgBox استفاده شده است که توسط آن پیام نوشته بر روی صفحه ظاهر شده و سیستم منتظر فشردن کلید OK برای ادامه سایر دستورات عملها در زیر روال خواهد بود.

سرانجام کلمه کلیدی End Sub نشان دهنده خاتمه زیرروال می‌باشد

کلیه دستوراتی که بین خطوط Public Sub و End Sub قرار دارند بدنه (body) زیرروال می‌باشند.

حال در هر جای برنامه که بخواهید زیرروال فوق را اجرا کنید فقط کافی است نام آن را قرار دهید.

Test-Message

در صورتی که زیرروالی که تعریف می‌شود دارای پارامتر باشد باید داخل پرانتز به تعداد پارامترهایی که دریافت می‌کند. متغیرهایی از نوع پارامترها تعریف کرد.

مثال:

Public Sub Area (L As Integer, W As Integer)

$$S = L * W$$

End Sub

و در برنامه اصلی زیرروال فوق را به صورت زیر صدا می‌زنیم.

Area (2,3)

که در این صورت مقدار S برابر عدد ۶ خواهد شد.

۶-۳ Functions

توابع بسیار شبیه زیرروالها می‌باشند با این تفاوت که در اثر اجرای آنها یک مقدار برگردانده می‌شود. توابع اکثراً اطلاعات ورودی خود را به عنوان پارامتر دریافت و نتیجه عملیات بر روی آنها را برمی‌گردانند. اجازه دهید به منظور درک بهتر موضوع ادامه مطلب را با ذکر یک مثال ادامه دهیم. در این مثال تابعی را تعریف می‌کنیم که دو عدد را به عنوان پارامتر دریافت کرده و حاصل جمع آنها را برمی‌گرداند.

Public Function Test-Add(a As Integer,b As Integer) As Integer

$$\text{Test-Add} = a+b$$

End Function

واژه کلیدی Public بیانگر این است که تابع فوق توسط تمامی قسمت‌های گوناگون برنامه شناخته شده بوده و می‌تواند فراخوانده شود. کلمه کلیدی Function نشانه تعریف تابع و به دنبال آن نام تابع که در این مثال Test-Add است آورده شده است. داخل پرانتز مقابل نام، دو متغیر از نوع Integer به عنوان پارامترهای ورودی تعریف شده‌اند.

As Integer آخر بیان کننده این مطلب است که خروجی این تابع از نوع Integer می‌باشد.

عبارت End Function نیز نشانه خاتمه دستورات تابع است.

اگر به بدنه تابع دقت کنید ملاحظه خواهید کرد که حاصل جمع دو پارامتر آن به نام تابع منتسب شده‌اند. بدین ترتیب مقداری را که می‌خواهیم تابع برگرداند مشخص شده یعنی هر گاه مقداری را که باید تابع برگرداند می‌بایست به نامش نسبت داده شود.

$$\text{Test-Add} = a+b$$

اگر دستورات زیر را در برنامه داشته باشیم:

Dim Value As Integer

Value = Test-Add (1,2)

مقدار سه به متغیر Value تخصیص می یابد.

Modules -۳-۷

Modules ها مکانهایی هستند که شما می توانید زیرروالها، توابع، datatype و ثابتهایی که عمومی می باشند، در آنجا تعریف کرده و با اضافه کردن آنها به پروژه های جدید از آن ویژگیها استفاده کنید.