

۱ مقدمه ای بر ویژوال بیسیک نت

بیسیک زبانی است که دوران تحول زیادی را پشت سر گذاشته است. اولین نسخه از زبان بیسیک طوری بود که برنامه نویسی در آن دشواری خاصی داشت. امروزه با جدیدترین نسخه بیسیک به نام ویژوال بیسیک نت سر و کار داریم. این زبان در محیط ویندوز قدرت فوق العاده ای دارد، به طوری که از سبک برنامه نویسی شیء گرا پیروی می کند. تولید برنامه ها در این زبان بسیار ساده است، چرا که بخش زیادی از برنامه را کامپایلر زبان تولید میکند.

برنامه نویسی شیء گرا

برنامه نویسی شیء گرا شیوه نوینی است که در آن می توان قطعاتی را ایجاد کرد و در برنامه های مختلف مورد استفاده قرار داد. قابلیت خوانایی برنامه هایی که در این روش نوشته می شوند بالا بوده، تست، عیب یابی و اصلاح آن ها آسان است. شیء گرای، بر اشیا تأکید دارد. تعدادی از زبان ها برای برای برنامه نویسی شیء گرا طراحی شده اند. زبان اسمالتاک یک زبان برنامه نویسی کاملاً شیء گرا است. زبان ++C که تکامل یافته زبان C است، دارای ویژگیهایی است که برنامه نویسی شیء گرا در آن امکان پذیر است. در این زبان، برنامه نویسی ساخت یافته نیز ممکن است. ویژوال بیسیک نت زبانی کاملاً شیء گرا است.

برنامه نویسی شیء گرا، اشیا دنیای واقعی را با نرم افزار مناسبی مدل سازی می کند. از امتیاز رابطه کلاس استفاده می کند که در آن، اشیا را از یک کلاس - مثل کلاسی از وسیله نقلیه -

مشخصات مشابهی دارند. از امتیاز **وراثت** و یا حتی **وراثت چندگانه** نیز استفاده می کند که در آن، کلاس های جدیدی که ایجاد می شوند، مشخصاتی از کلاسهای موجود را به ارث می برند و حتی ویژگیهای منحصر به فردی نیز دارند.

برنامه نویسی شیء گرا، داده ها (صفات)، و توابع (رفتارها) را در بسته ای به نام اشیاء، بسته بندی می کند. داده ها و توابع هر شیء شدیداً به هم گره خورده اند. اشیا دارای ویژگیهای پنهان سازی اطلاعات اند. یعنی، گرچه اشیا ممکن است طریقه ارتباط با اشیای دیگر را از طریق واسط بدانند، ولی اجازه ندارند که از چگونگی پیاده سازی اشیای دیگر آگاه باشند. به عبارت دیگر، جزئیات پیاده سازی، در داخل اشیا مخفی شده است.

در C و سایر زبان های برنامه سازی ساخت یافته، برنامه ها به فعالیت ها توجه دارند، در حالی که در زبان ویژوال بیسیک نت به اشیا توجه می شود. در C، واحد برنامه نویسی، تابع است. در ویژوال بیسیک نت، واحد برنامه نویسی، کلاسی است که اشیا سرانجام از آن نمونه سازی (ایجاد) می شوند. کلاس های ویژوال بیسیک نت حاوی رویه ها هستند.

برنامه نویسی ویژوال بیسیک نت بر روی ایجاد انواع جدیدی به نام کلاس متمرکز می شود. همان طور که می دانید، بعضی از انواع داده ها مثل String, Double, Integer در ویژوال بیسیک نت موجود است. این انواع را انواع ابتدایی می نامیم. برنامه نویس می تواند با استفاده از انواع ابتدایی، انواع دیگری را ایجاد کند که آن ها را انواع جدید می نامیم. هر کلاس علاوه بر داده ها، توابعی

دارد که داده ها، توابعی دارد که داده ها را دستکار می کنند. قطعات داده ای کلاس را اعضای داده ای گویند. قطعات تابعی کلاس را توابع عضو یا متد گویند. می دانید که نمونه ای از هر نوع

ابتدایی را متغیر گویند. به عنوان مثال، در دستور Dim x As Integer, x نمونه ای از نوع Integer

است و متغیر نام دارد. نمونه ای از انواع جدید (مثل کلاس) را شیء گویند. در ویژوال بیسیک

نت، بر کلاس ها، ولی در C بر توابع تأکید میشود. اسامی موجود در صورت مسئله (مشخصات

سیستم) به برنامه نویس کمک می کند تا کلاس هایی را تعیین می کند که با ایجاد اشایی از آن ها

که با یکدیگر کار می کنند، سیستم را پیاده سازی کند.

جدول ۱-۱ انواع عددی در ویژوال بیسیک نت.		
نوع	میزان حافظه (بایت)	بازده قابل قبول
Byte	۱	۰ تا ۲۵۵
Integer	۲	۳۲۷۶۸- تا ۳۲۷۶۷
Long	۴	۲۱۴۷۴۸۳۶۴۷ تا -۲۱۴۷۴۸۳۶۴۸
Single	۴	منفی از $-۱/۴۰۱۲۹۸E-۴۵$ تا $۳/۴۰۲۸۲۳E+۳۸$ مثبت از $۳/۴۰۲۸۲۳E+۳۸$ تا $۱/۴۰۱۲۹۸E-۴۵$
Double	۸	منفی از $-۱/۷۹۷۶۹۳۱۳۴۸۶۲۳۲+E ۳۰۸$ تا -۳۲۴ $۳/۴۰۲۸۲۳E+۳۸$ تا $۱/۴۰۱۲۹۸E-۴۵$ مثبت از -۳۲۴ $۱/۷۹۷۶۹۳۱۳۴۸۶۲۳+E ۳۰۸$ تا $۳/۴۰۲۸۲۳E+۳۸$
Currency	۸	از $۹۲۲۳۳۷۲۰۳۶۸۵۴۷۷/۵۸۰۷$ تا $-۹۲۲۳۳۷۲۰۳۶۸۵۴۷۷/۵۸۰۸$
Decimal		اگر از اعشار استفاده شود در بازه $۷/۹۲۲۸۱۶۲۵۱۴۲۶۴۳۳۷۵۹۳۵۴۳۹۵۰۳۳۵$ و اگر از اعشار استفاده نشود در بازه $±۷۹۲۲۸۱۶۲۵۱۴۲۶۴۳۳۷۵۹۳۵۴۳۹۵۰۳۳۵$

جدول ۱-۲ انواع غیر عددی.		
نوع	میزان حافظه (بایت)	بازه قابل قبول
String (طول ثابت)	طول رشته	از یک تا ۶۵۴۰۰ کاراکتر
String (طول متغیر)	طول رشته + ۱۰ بایت	از صفر تا ۲ میلیارد کاراکتر
Date	۸	تاریخ قابل قبول
Boolean	۲	True یا False
Variant (عددی)	۱۶ بایت	به اندازه نوع عددی Double
Variant (رشته ای)	طول + ۲۲ بایت	مانند رشته طول متغیر

نوع رشته ای

رشته دنباله ای از کاراکترها است، مثل Ali و Computer Science نوع رشته ای معمولاً برای ذخیره اسامی افراد و سایر متن ها به کار می رود و به صورت طول متغیر و طول ثابت استفاده می

شود.

نوع بولی

این نوع که نوع منطقی نیز نامیده می شود داده هایی را نشان می دهد که می تواند دو مقدار درستی یا نادرستی را بپذیرند که به ترتیب با True و False نمایش داده می شوند.

نوع تاریخ

تاریخ و زمان را می توان در ویژوال بیسیک نت به عنوان یک نوع مورد استفاده قرار داد، برای این

منظور باید در دو طرف آن ها علامت \neq قرار گیرد. مثال های زیر را ببینید:

≠ March 14 , 1999 ≠

≠ 19:11:22 ≠

نوع Variant

از این نوع معمولاً در مواردی استفاده می شود که از یک محل حافظه برای ذخیره چندین نوع استفاده می گردد. یعنی یک محل Variant برای ذخیره موقت هر نوع داده ها به کار می رود که این داده ها بعداً در محل دیگری با نوع مشخص ذخیره می شوند این نوع تمام انواع داده ها به جز رشته ای با طول ثابت را نگهداری می کند.

متغیرها

متغیرها نام هایی برای کلمات حافظه اند و محتویات آن ها ممکن است در حین اجرای برنامه تغییر کنند. برای نامگذاری متغیرها از ترکیبی از حروف a تا z (کوچک و بزرگ)، ارقام و خط زیر (_) استفاده می شود، به طوری که با رقم شروع نشوند، مثل SI ، SUMI و 2 _ Average طول نام متغیرها می تواند تا ۲۵۵ حرف باشد.

اعلان متغیرها

قبل از این که از این متغیر استفاده کنید، باید آن ها را در برنامه اعلان نمایید. برای این منظور، نام و نوع آن ها را تعیین می کنید و برای اعلان متغیرها از دستور Dim استفاده می شود:

نوع متغیر As نام متغیر Dim

نام متغیر نامی است که برای متغیر خود انتخاب می کنید و نوع متغیر تعیین می کند که متغیر مورد نظر از چه نوعی باشد. مثال های زیر را ببینید:

Dim X As Integer

Dim Y As Currency

Dim ST As String, P₁ As Long

Dim STR As String *5

Dim SUM As Variant

دستور اول متغیر X را از نوع صحیح، دستور دوم متغیر Y را از ارزی، دستور سوم متغیر ST را از نوع رشته ای با طول متغیر و P₁ را از نوع صحیح بزرگ و دستور چهارم متغیر STR را از نوع رشته ای با طول ثابت ۵ و دستور پنجم متغیر SUM را از نوع Variant تعریف می کند. به شکل تعریف متغیرهای رشته ای طول ثابت و طول متغیر توجه کنید. در مورد رشته های با طول ثابت دقت داشته باشید که اگر رشته انتساب داده شده به آن، بیش از طول تعیین شده باشد، فقط تا همان طول ثابت منظور خواهد شد.

اگر نوع متغیری را Variant تعریف کنید، در ادامه برنامه وقتی مقداری در آن قرار گرفت، نوع آن تعیین می شود:

Dim x As Variant

$$X = 10$$

دستور اول نوع متغیر x را Variant تعریف می کند دستور دوم مقدار ۱۰ را در x قرار می دهد و

در نتیجه نوع آن Integer تعیین خواهد شد.

مقدار دادن به متغیرها

پس از اعلان متغیرها می توان به آن ها مقدار داد و از آن ها استفاده کرد. برای مقدار دادن به سه

روش عمل می شود: (۱) مقدار دهی اولیه در هنگام معرفی متغیر، (۲) دستور انتساب و (۳)

دستورات ورودی. در این بخش دستور مقدار دهی اولیه در هنگام تعریف و انتساب را بررسی

کرده، دستورات ورودی را در جای مناسبی شرح خواهیم داد. مثال های زیر را ببینید:

STR را از نوع رشته تعریف می کند. Dim STR As string

X را از نوع Integer تعریف کرده ۱۲۵ را در آن قرار می دهد. Dim x As Integer = 125

Dim Y As Single = 1472.65 در Y قرار میگیرد.

رشته AHMAD در STR قرار می گیرد. STR = "AHMAD"

st را از نوع رشته ای تعریف کرده رشته Computer Science را در آن قرار می دهد.

Dim st As String = Computer Science

Dim Sum As Integer Sum را از نوع Integer تعریف می کند

SUM = 2545 در SUM قرار میگیرد. SUM = 2545

عملگرها

عملگرها نمادهایی هستند که برای انجام عملیات بر روی متغیرها و مقادیر ثابت مورد استفاده قرار

می گیرند، مثل نماد + که برای عمل جمع به کار می رود. عملگرها به چهار دسته تقسیم می شوند:

حسابی (جدول ۱-۳) رابطه ای (جدول ۱-۴)، منطقی (جدول ۱-۶).

در مورد جدول ۱-۵ دقت داشته باشید که X و Y عبارات منطقی اند که ممکن است ارزش

درستی یا نادرستی داشته باشند. با فرض این که T ارزش درستی و F ارزش نادرستی باشد،

عملکرد هر کدام از این عملگرها در جدول ۱-۷ آمده است. عملگر NOT بر روی یک عملوند

عمل می کند و ارزش آن را نقیص می کند عملگر AND بر روی دو عملوند ارزش T داشته

باشند، در بقیه موارد ارزش نادرستی دارد. حاصل عملگر OR وقتی ارزش نادرستی است که هر

دو عملوند آن ارزش نادرستی داشته باشند، در بقیه موارد ارزش درستی دارد. حاصل عملر

XOR وقتی ارزش درستی است که ارزش یکی از دو عملوند T و دیگری F باشد. حاصل

عملگر EQU وقتی ارزش درستی است که ارزش هر دو عملوند یکسان باشد (هر دو T یا هر دو

F باشند). حاصل عملگر IMP وقتی ارزش نادرستی است که عملوند راست F باشد در بقیه

موارد ارزش آن درست است.

تعریف عبارت

عبارت، هر ترکیبی از متغیرها، ثوابت و عملگرها است که نتیجه آن می تواند ارزیابی شود. عبارات زیر را ببینید:

$$2+3$$

$$4+X*Y$$

$$(6+P)*M$$

برای ارزیابی عبارات باید بدانیم که عملگرها به چه ترتیبی اجرا می شوند. به عبارت دیگر تقدم

عملگرها چگونه است. حاصل عبارات اول برابر با 5 است. اگر $X=2$, $Y=4$ باشد، حاصل

عبارت دوم چیست؟ دقت کنید که ابتدا عمل ضرب ($2*4$) انجام می شود که حاصل آن 8 است

و این مقدار با 4 جمع می شود که حاصل این عبارت 12 است. می گوئیم تقدم عملگر ضرب از

تقدم عملگر جمع بیشتر است. اگر $P=4$, $M=7$ باشد، حاصل عبارت سوم چیست؟ چون $6+P$

در داخل پرانتز قرار دارد، ابتدا عمل جمع انجام می شود، یعنی $6+4$ برابر با 10 است و این مقدار

در 7 ضرب شده حاصل عبارت 70 است.

اگر در یک عبارت، عملگرهای محاسباتی، رابطه ای و منطقی با هم وجود داشته باشند، تقدم آن

ها به شرح زیر است:

۱- عملگرهای محاسباتی

۲- عملگرهای رابطه ای

۳- عملگرهای منطقی

جدول ۱-۳ عملگرهای حسابی			
عملگر	نام	مثال	نتیجه
\wedge	توان	3^2	9
$*$	ضرب	$4*5$	20
$/$	تقسیم	$16/2$	8
$+$	جمع	$2+5$	7
$-$	تفریق	$15 \text{Mod } 3$	5
Mod	باقیمانده تقسیم	$15 \text{ Mod } 3$	0
\backslash	تقسیم صحیح	$14 \backslash 3$	4
$\&$ و $+$	الحاق رشته ها	$"AL" + " "$	"ALI"

جدول ۱-۴ عملگرهای رابطه ای		
عملگر	نام	مثال
$<$	کوچک تر	$x < y$
$< =$	کوچکتر مساوی	$x > y$
$>$	بزرگ تر	$> y$
$> =$	بزرگ تر مساوی	
$< >$	نا مساوی	

جدول ۱-۴ عملگرهای منطقی.

عملگر	نام	مثال
NOT	نقیض	NOT X

www.kandoo.cn.com

www.kandoo.cn.com

www.kandoo.cn.com

www.kandoo.cn.com

www.kandoo.cn.com