

بسمه تعالی



ت

دانشگاه آزاد اسلامی واحد سقز
گروه کامپیوتر و فناوری اطلاعات

برنامه نویسی با VB.NET

امجد عثمانی

Amjad.osmani@yahoo.com

این جزوه شامل موارد زیر است:

- (۱) تعریف متغیر
- (۲) انواع داده
- (۳) انواع ثابت ها (شامل ثابت شمارشی)
- (۴) تبدیل انواع (پرخطر و کم خطر - CType)
- (۵) عملگر ها (شامل عملگر Like)
- (۶) دستورات شرطی (if then else - select case)
- (۷) کار با انواع حلقه ها (while - for - for Each - do - Exit - Continue)
- (۸) تابع IIF
- (۹) تابع Microsoft.VisualBasic.Choose
- (۱۰) سابروتین
- (۱۱) تابع
- (۱۲) انواع ارسال آرگومانها (Byval - Byref)
- (۱۳) انواع پارامتر های ورودی (اختیاری و اجباری)
- (۱۴) ارسال آرگومانها به تعداد متغیر (ParamArray)
- (۱۵) حفظ مقدار داده ها در فاصله ی میان دو فراخوانی (Static)

[illegible]

عبارت Lambda	(۱۶)
Func Type	(۱۷)
Extension Method/Attribute	(۱۸)
انواع الگوریتم ها (بازگشتی و غیر بازگشتی)	(۱۹)
آرایه	(۲۰)
Jagged Arrays	(۲۱)
Array List	(۲۲)
کار بر روی رشته ها	(۲۳)
کار با کلاس ها (سازنده ها - Shared Members - وراثت - Overloadin	(۲۴)
& Overriding - پنهان کردن متدها - متدها و کلاس های انتزاعی -	
ایندکسر ها)	
Operator Overloading	(۲۵)
Structure	(۲۶)
LINQ	(۲۷)
مقابله با خطا	(۲۸)
Goto دستور	(۲۹)
کار با Collection ها (Stack - Queue - Hashtable -)	(۳۰)
کلاس های از نوع Generic	(۳۱)
مثال های گوناگون	(۳۲)
کار با Timer در محیط لاینول	(۳۳)
ترتیب اعداد تعادلی	(۳۴)
الگوریتم های جستجو (ترتیب د)	(۳۵)
	(۳۶)

☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐ 6 ☐ 7 ☐ 8 ☐ 9 ☐ 10 ☐ 11 ☐ 12 ☐ 13 ☐ 14 ☐ 15 ☐ 16 ☐ 17 ☐ 18 ☐ 19 ☐ 20 ☐ 21 ☐ 22 ☐ 23 ☐ 24 ☐ 25 ☐ 26 ☐ 27 ☐ 28 ☐ 29 ☐ 30 ☐ 31 ☐ 32 ☐ 33 ☐ 34 ☐ 35 ☐ 36 ☐ 37 ☐ 38 ☐ 39 ☐ 40 ☐ 41 ☐ 42 ☐ 43 ☐ 44 ☐ 45 ☐ 46 ☐ 47 ☐ 48 ☐ 49 ☐ 50 ☐ 51 ☐ 52 ☐ 53 ☐ 54 ☐ 55 ☐ 56 ☐ 57 ☐ 58 ☐ 59 ☐ 60 ☐ 61 ☐ 62 ☐ 63 ☐ 64 ☐ 65 ☐ 66 ☐ 67 ☐ 68 ☐ 69 ☐ 70 ☐ 71 ☐ 72 ☐ 73 ☐ 74 ☐ 75 ☐ 76 ☐ 77 ☐ 78 ☐ 79 ☐ 80 ☐ 81 ☐ 82 ☐ 83 ☐ 84 ☐ 85 ☐ 86 ☐ 87 ☐ 88 ☐ 89 ☐ 90 ☐ 91 ☐ 92 ☐ 93 ☐ 94 ☐ 95 ☐ 96 ☐ 97 ☐ 98 ☐ 99 ☐ 100

برخی از فضاها هم عبارتند از:

- 1- System : حاوی کلاس های پایه ای است.
- 2- System.Collection : حاوی کلاسهای کار با انواع ^{کلیکون} ها مثل صف
آرایه و جدول ردیف سازی می باشد.
- 3- System.Windows.Forms : حاوی کلاسهای برای ساخت برنامه های کاربردی
ویندوزی است. با استفاده از این کلاسها می توان با به کار گیری امکانات
سیستم عامل و ویندوز، واسطه های کاربردی غنی ایجاد کرد.
- 4- System.Data : حاوی کلاسهای لازم برای کار بر روی پایگاه داده ها
مثل SQL Server می باشد.

-5-

↑

لایه ۱	Applications (برنامه‌های کاربردی) C#, VB, ...
لایه ۲	FCL: Framework Class Library
لایه ۳	CLR: Common Language Runtime
لایه ۴	OS: Operating System

↓
سنت افزار کامپیوتر

باتوجه به شکل بالا یک معماری لایه‌ای را می‌توان در نظر گرفت.

۱- در لایه ۱ لایه لایه برنامه‌های کاربردی قرار دارند. باز اینها به شکل C# و VB نوشته می‌شوند.

۲- در لایه ۲ هم کلاس‌های کاربردی Net یا FCL قرار دارند. این کلاس‌ها در فایل‌هایی قرار دارند که به آنها Assembly گفته می‌شود. به منظور لایه‌بندی و مدیریت اجرای برنامه‌های مبتنی بر Net، این اسمبلی‌ها به نام Load می‌شوند. باتوجه به تعداد زیاد کلاس‌های موجود در چارچوب Net، مایکروسافت آنها را در Namespace ها دسته‌بندی می‌کند. کلاس‌ها بر حسب نوع کاری که انجام می‌دهند در Namespace ها دسته‌بندی می‌شوند.

۳- در لایه سوم CLR قرار دارد. CLR قلب NetFramework است. CLR مدیریت اجرای دستورات را بر عهده دارد و سرویس‌های لازم مانند امنیت، مدیریت حافظه، و ... را نیز مدیریت می‌کند. در اختیار برنامه‌های کاربردی قرار می‌دهد.

Data Type

→ B.Net انواع داده ها به تو گروه زیر تقسیم می شوند:

1- ذاتی: که به صورت پیش ساخته در درون زبان قرار دارند مثل

Integer - char - string - short , ...

2- کاربری یا UDT: که توسط برنامه نویس تعریف می کنند مثل

class - structure - enum , ...

انواع داده ها در B.Net عبارتند از:

	Size (Byte)	Type
True / false	N/A	Boolean
	1	Byte
اعداد صحیح	2	Short
~	4	Integer
~	8	Long
اعداد اعشاری	4	Single
~	8	Double
کاراکتر	2	Char
سری	N/A	String
مقادیر تاریخ و زمان	8	Date
هو مقارن	4	Object

→ VB.Net انواع داده‌ها بر حسب نحوه ذخیره‌سازی و نحوه دستیابی دو گروه زیر تقسیم می‌گردد:

1- مقدار ی: این دسته مقدار مربوط به خود را در درون خود نگه‌داری می‌کنند و نیازی به تعیین مقدار آن‌ها را مستقیماً مقدار دستیابی قرار دادن مثل: Integer, Single, ...

2- ارجاعی: این دسته مقدار مربوط به خود را نگه‌نمی‌دارند و به جای آن به یک مکانی در حافظه اشاره می‌دهند. نیازی به تعیین مقدار آن‌ها در هنگام تعریف در این دسته است اما باید آنگاه که می‌خواهیم آن‌ها را به دست آوریم در دسترس باشند. استفاده از آن به سبب ذخیره‌سازی واقعی نیست. مثل:

String, Object, ...

متغیر: قسمتی از حافظه برای ذخیره اطلاعات است.

ذاتی → نوع متغیر $A\&$ نام متغیر Dim

1- نام متغیر باید با کاراکتری از میان کاراکترهای $a-z$ یا $A-Z$ شروع

2- نام متغیر می تواند حاوی اعداد 0 تا 9 باشد

3- نام متغیر می تواند با underline شروع شود و با اعدادی آن باشد.

4- اگر نخواهیم کامپایلر برنامه نویسی را مجبور به تعریف متغیر به صورت زیر بنماید با سی از Option Explicit on استفاده کنیم.

نام متغیر Dim

5- اگر نخواهیم کامپایلر برنامه نویسی را مجبور به تعریف متغیر به صورت زیر بنماید با سی از Option Strict on استفاده کنیم.

نوع متغیر $A\&$ نام متغیر Dim

6- هرگاه است همواره نوع داده های همه متغیرها را مشخص کنید تا کامپایلر بتواند موارد استفاده نامناسب آنها را تشخیص دهد و تبدیل آن برنامه را به یک خطا یا اخطار تبدیل کند. در غیر این صورت بسیاری از مشکلات سازمان اجرای برنامه معلوم نخواهند شد و بنابراین احتمال آلودگی برنامه در زمان اجرا، به دلیل وقوع چنین خطاهای متوقف شود. افزایش می یابد. ضمناً سرعت تبدیل انواع داده های مختلف به یکدیگر نیز کاهش می یابد که ممکن است باعث کاهش سرعت اجرای برنامه شود.

خبر سال از قریب به صفر

Dim a, b As Short →

تقریب دو متغیر از نوع short

Dim a As short, b, c As Integer →

Dim a As Integer = 100 →

مقدار دهی اولیه به متغیر a

گات ها:

گات هم مثل متغیر یک مقدار را در خود ذخیره می کند اما برخلاف متغیر مقدار معهود در یک گات را نمی توان در جریان اجرای برنامه تغییر داد

ذاتی → نوع گات A & نام گات Const

۱- نام گات از قوانین مربوط به نام گذاری متغیرها پیروی می کند:

مثال: برنامه ای بنویسیم ساعت دایره را به دست بیاورد؟

Const Pi AS Double = 3.14

Dim r AS Double

Dim a AS Double = Pi * r ^ 2

↓
ساعت دایره

انواع گات: ۱- نمادین (Symbolic) ۲- Literal ۳- شمارشی (enum)

۱- نمادین: با کلمه کلیدی Const شروع می شود

Const b AS Integer = 101

۲- Literal: مثل عدد 32 در دستور مقابل

a = a + 32

۳- شمارشی (enum): گاهی کار کردن با نام های نامدار (شمارشی) آسان تر از کار کردن با مقدار گات عددی است. با استفاده از گات شمارشی می توان گات ها را نامدار مرتبط ایجاد کرد. چنین مجموعه ای می تواند حاوی نام رنگها یا حای نام روزهای هفته باشد.

```

enum نوع رنگ Color {
    1 سبز Green,
    2 قرمز Red,
    ...
}
end enum
  
```

```

enum Color {
    Red = 11
    Green = 1
    Blue
}
end enum
  
```

Dim C As Color

C = Color.Red

C = 11 → Option Strict Off

C = CType(N, Color)

Option Strict On

Console.WriteLine(C) → 11 چاپ می‌کند

Console.WriteLine(C.ToString) →

دروغی Red چاپ می‌کند

تبدیل انواع داده ها
 معروف است لازم باشد که داده های از یک نوع، در مقیاس های از نوع دیگر قرار گیرند و عمل
 تبدیل نوع (Type conversion) صورت می گیرد.
 دستورات زیر را در نظر بگیرید:

Dim a As Short = 11

Dim b As Long

$b = a \rightarrow$ این نوع تبدیل را تبدیل پستی می گویند که بی خطا است

$a = b \rightarrow$ این نوع تبدیل را تبدیل سببی می گویند که داده های با اندازه بزرگتر را
 به مقادیر با اندازه کوچکتر تبدیل می کند و خطا است

۱- برای جلوگیری برنامه نویس به استفاده از این دستورات می تواند تبدیل نوع از
 Option Strict On استفاده کند.

۲- توابع تبدیل نوع به شرح زیر:
 CBool - CByte - CChar - CDate - CDBL - CInt - CLng
 CObj - CShort - CSng - CStr - CType - Val - Str

۳- تابع CType: AS (نوع مقصد، مقدار یا درگاه تبدیل) CType
 مقدار در قالب نوع مقصد

$$a = a + b$$

1- توان * ضد + جمع - تفریق / تسمیہ اعلیٰ درجی
2- تسمیہ صریح MOD با قیامندہ

$$8_{\text{m}} - 2 = 1_{\text{m}} / 5 \times 5 = 1_{\text{m}} = 1_{\text{m}} \text{ (15) } - 2$$

$\rangle_s \langle_s \rangle \langle \rangle = \text{ایک ایسا } \langle \rangle \text{ ہے جس کا } s=3$
 Like

4- [✓] مملکتان اتصال

orelse Andalso or And NOT : 5

$$a = a + 1 \iff a + 1$$

امکان توپا: یکی از فنون بهین‌تری است که توسط کامپایلر به کار می‌رود. در این سوره
بخشهای زاید شرط که نتیجه‌ی آنها تأثیری بر نتیجه‌ی نهایی ندارد از زبانی منی کمند
و سرعت اجرای دستورات بالایی رود. نمونگیهای And و OR هر دو عبارت
منطقه‌ی خود را حدیقل از مقدار 1 ن بررسی می‌کنند اما اگر به جای And از
AndAlso استفاده شود در صورتی که نتیجه عبارت اول False باشد عبارت دوم از زبانی
منی شود و اگر به جای OR از OrElse استفاده کنند در صورتی که نتیجه عبارت اول
True باشد عبارت دوم محاسبه نمی‌گردد. (از زبانی منی شود).

نمک like
نتیجی آن True یا False

Dim Str1 As String	} like
Dim Str2 As String	
Dim res As Boolean	
res = Str1 like Str2	

- 1- [] بازهای از کار، آترها را لای نیاورد
- 2- # نه نهی یک کار، آتر عدداست
- 3- ? نه نهی یک کار، آتر است
- 4- * نه نهی عدادی از کار، آترها است

Str1	like	Str2
aBBBa		a#a
F		[A-Z]
F		[!A-Z]
a2a		a#a
BAT123		B?T*
aM5b		a[L-P]#[!c-e]
.Net		?N*
FR AMEWork		[A-Z]*
CLR 123		?*#2#
1classobject123		1*obj*#23

[illegible]

→ VB.Net برنامه ها از دستورات تشکیل می شوند. هر دستوری Statement کاری را که باید توسط کامپیوتر انجام گیرد مشخص می کند.

→ VB.Net دستورات مربوط به تصمیم گیری به انواع زیر تقسیم می شوند:

1- if-then-else :

به کمک دستورات شرطی می توان اجرای برنامه را بر اساس مقدار متغیرها و یا مقایسه آنها به مسیرهای مختلف هدایت کرد.

اختیار این دسته به سه زیراست :

if شرط Then
↓
-endif دستورات if

if شرط Then
↓
A دستورات
else
B دستورات
↓
-endif if-else

if A شرط Then
↓
A دستورات
else
if B شرط Then
↓
B دستورات
else
C دستورات
↓
-endif
-endif

⇒

if A شرط Then
↓
A دستورات
else if B شرط Then
↓
B دستورات
else
C دستورات
↓
-endif

مثال: بررسی و شمار if بررسی کند که آیا عدد معطوری متغیر n زوج است یا فرد؟

```
Dim n As Integer
```

```
n = CInt(Console.ReadLine())
```

```
if (n Mod 2 = 0) Then  
    Console.WriteLine("{0} is a even number", n)  
else  
    Console.WriteLine("{0} is a odd number", n)  
endif
```

مثال: ضمن بررسی و شمار if زیر، خروجی یابی را در صورتی که مقدار متغیر day برابر با 3 باشد، بنویسید؟

```
Dim day As Integer = 3
```

```
if day = 1 Then  
    Console.WriteLine("شنبه")  
else  
    if day = 2 Then  
        Console.WriteLine("یکشنبه")  
    else  
        if day = 3 Then  
            Console.WriteLine("دوشنبه")  
        else  
            Console.WriteLine("سه شنبه")  
        endif  
    endif  
endif
```