

# VB.NET - P1

由微软开发，将 .NET Framework 和公共语言运行库的强大功能与作为 Visual Basic 标志的生产力优势相结合

VB.NET (Visual Basic .NET) 是在 .NET Framework 上实现的面向对象的编程语言  
不兼容 VB6，任何旧版本编写的代码均不能在 VB.NET 下编译

## 编程功能

布尔条件 (boolean condition)

自动垃圾回收 (automatic garbage collection)

标准库 (standard library)

程序集版本控制 / 组件版本 (assembly versioning)

属性 (property) 和事件 (event)

委托 (delegate) 和事件管理 (event management)

易于使用的泛型 (generic)

索引器 (indexer)

条件编译 (conditional compilation)

简单的多线程 (multithreading)

.NET 框架 (.NET Framework)，用于编写 Windows 应用 / web 应用 / 网页服务的平台

.NET 框架应用是多平台应用程序

设计方式可以使用不同语言：Visual Basic / C# / C++ / Jscript / COBOL

## 组件

公共语言运行时 (common language runtime, CLR)

.NET 框架类库 (.NET Framework class library)

公共语言规范 (common language specification)

通用类型系统 (common type system)

元数据和组件 (metadata and assembly)

Windows 窗体 (Windows form)

ASP.NET 和 ASP.NET AJAX

ADO.NET (ActiveX data object)

Windows 工作流基础 (Windows workflow foundation, WF)

Windows 演示基础 (Windows presentation foundation)

Windows 通讯基础 (Windows communication foundation, WCF)

语言集成查询 (LINQ, language integrated query)

# VB.NET - P2

8.9 - Namespaces

命名空间声明 (namespace declaration), 基本语法为 Imports <namespace>  
[Namespace] used to include namespace in the program

类 / 模块 (class / module), 每个程序必须包含一个类的模块 (module of a class)  
[Module] to include class  
VB.NET 是完全面向对象的 (completely object oriented)  
[Object to include]  
包含程序使用的数据和过程 (procedure)

过程 (procedure), 类 / 模块 包含一个或多个过程  
[Procedure] to include  
过程包含可执行代码, 定义了类 / 模块 的行为  
[Behavior]  
过程有不同种类: Function / Sub / Operator  
Get / Set

Add Handler / Remove Handler / Raise Event

主过程 (Main procedure) : VB.NET 程序的入口点, (entry point)

注释 (comment), 在 VB.NET 中, 单引号 (' ) 后的程序视为注释, 被编译器忽略

class 定义为描述其类型的对象所支持的行为 / 状态 的模板 / 蓝图

define template / blueprint describe behavior / state that object support

object 具有状态 和 行为 (state and behavior)

method (方法), 类可以包含多个方法, 方法基本上是一种行为

where logics are written, data is manipulated, actions are executed

instance variable (实例变量), 每个对象有唯一的实例变量集

对象的状态由其实例变量 的值 组成

object's state created by values assigned to instance variables

标识符 (Identifier), 在 VB.NET 中用于类, 变量, 函数或其他用户定义项的名称

由字母, 数字, 下划线组成, 且 第一个字符不能是数字

不能包含嵌入的空格 (embedded space) 或其他特殊符号

不能是保留关键字 (reserved keyword)

# VB.NET - P3

139 - Second

## 数据类型

| type         | storage allocation                | value range  | in-line type conversion |
|--------------|-----------------------------------|--|-------------------------|
| Byte         | 1 byte (8-bit)                    | 0 ~ 255 (无符号)  | CByte()                 |
| SByte        | 1 byte (8-bit)                    | -128 ~ 127 (有符号)   | CSByte()                |
| Char         | 2 byte (16-bit)                   | 0 ~ 65535 (无符号)  | CChar()                 |
| Short        | 2 byte (16-bit)                   | -32768 ~ 32767 (有符号)   | CShort()                |
| UShort       | 2 byte (16-bit)                   | 0 ~ 65535 (无符号)  | CUShort()               |
| Integer      | 4 byte (32-bit)                   | - $2^{31}$ ~ $2^{31}-1$ (有符号)  | CInt()                  |
| UInteger     | 4 byte (32-bit)                   | 0 ~ $2^{32}-1$ (无符号)   | CUInt()                 |
| Long         | 8 byte (64-bit)                   | - $2^{63}$ ~ $2^{63}-1$ (有符号)  | CLng()                  |
| ULong        | 8 byte (64-bit)                   | 0 ~ $2^{64}-1$ (无符号)   | CULng()                 |
| Single       | 4 byte (32-bit)                   | 单精度浮点数 (float)   | CSng()                  |
| Double       | 8 byte (64-bit)                   | 双精度浮点数   | CDbl()                  |
| Decimal      | 16 byte (128-bit)                 | 0 ~ $\pm 2^{96}-1$ , 没有十数点<br>$0 \sim \pm \frac{2^{96}-1}{10^{28}}$ , 有28位十数 | CDec()                  |
| Boolean      | 取决于 implementing platform         | True / False   | CBool()                 |
| String       | implementing platform             | 0 ~ 约20亿个 Unicode 字符   | CStr()                  |
| Date         | 8 byte (64-bit)                   | 0001/01/01 00:00:00 ~ 9999/12/31 23:59:59                                    | CDate()                 |
| Object       | 4 byte (32-bit) / 8 byte (64-bit) | 任何类型都可存储于 Object 类型  | CObj()                  |
| User-Defined | implementing platform             | 每个成员由其数据类型确定并独立于其他成员   |                         |

## 变量声明

在VB.NET中, Dim语句用于一个或多个变量的声明和内存分配

用于模块,类,结构(Structure),过程或块(block)

基本语法为 [ <attributelist> ] [ access modifier ] [ [ Shared ] [ Shadows ] [ Static ] ]

[ ReadOnly ] Dim [ With Events ] <variablelist>

attributelist : 适用于变量的属性列表

access modifier : 定义变量的访问级别 (access level)

Public / Protected / Friend / Protected Friend / Private

Shared : 声明共享变量,不与类或结构的任何实例关联(绑定),而是可用于所有实例

Shadows : 表示变量在基类中重新声明和隐藏一个同名元素或一组重载元素

Static : 表示变量将保留其值,即使在声明的过程终止之后

ReadOnly : 表示变量可以读取,但不能写入

WithEvents : 指定变量用于响应分配给变量实例引发的事件

specify variable used to respond to event

raised by instance assigned to the variable

# VB.NET - P4

## 变量声明

~~variablelist~~ 提供声明的变量列表

基本语法结构为 <variable> [, <variable>]

variable 的基本语法结构为

<variablename> [(<boundslist>)] [As <New>] <datatype> [= <initializer>]

variablename : 变量声明的名字, 应为合法标识符

boundslist : 在声明数组变量时, 表示数组变量的每个维度的边界列表

New : 表示在 Dim 语句运行时, 创建一个类的新实例

datatype : 表示声明变量的数据类型

如果 Option Strict 设置为 On, 则 <datatype> 是必需的

initializer : 表示在声明变量时, 计算表达式的值并赋值给创建的类的新实例

如果已指定了 New, 则 = <initializer> 是必需的

## 初始化

(initialization), 在 VB.NET 中, 变量可以在声明时初始化

基本语法结构为 [= constant | expression (常量表达式)]

## 输入

在 VB.NET 中, 命名空间 System 中的控制台类 (Console) 提供 ReadLine 函数用于用户输入

如 message = Console.ReadLine 将用户输入的字符串赋值给 message

另外 Console.ReadLine() 也用于在 Visual Studio .NET 中

阻止屏幕在程序启动后快速运行完成后关闭

左值 (lvalue) : 可出现在赋值语句左侧或右侧的表达式且可修改

右值 (rvalue) : 只能出现在赋值语句右侧的表达式

字面量与常量表达式都属于右值

widening conversion (扩大转换), 数据类型可以扩大转换为另一个类型

Byte ≤ Short ≤ Integer ≤ Long ≤ Decimal ≤ Single ≤ Double

注意在转换时可能损失精度 (precision), 但是不会损失信息 (information) 和指数

widening conversion 总是成功的, 所以可以隐式地使用

enumerated type (枚举) → underlying integer 以及 integer 可扩展的类型

Char → Char, String

任意 type → Object 以及其实现 (implement) 的接口 (interface)

任意 derived type → 任意基类 (base type)

Nothing → 任意 data type 或 object type

# VB.NET - P5

2019-September

## 常量

(constant), 在VB.NET程序执行过程中不会改变的固定值, 也称为文字(literal)

可以是任何基本数据类型: 整数/浮点数/字符/字符串/枚举

视为常规变量( regular variable), 除了在定义之后不能修改常量的值

在VB.NET中, Const语句用于一个或多个常量的声明和定义

用于模块, 类, 结构, 过程, 块中, 替换文字(in place of literal value)

基本语法为 [<attributelist>] [<accessmodifier>] [Shadows]

Const <constantlist>

attributelist: 指定应用于常量的属性列表

可以提供多个属性, 并用逗号分隔

accessmodifier: 指定常量的访问控制

Public / Protected / Friend / Protected Friend / Private

Shadows: 表示常量在基类中隐藏一个同名的编程元素(programming element of identical name)

constantlist: 提供声明的常量列表

基本语法结构为 <constant> [, <constant>]

constant 的基本语法结构为 <constantname> [ As <datatype> ] = <initializer>

constantname: 常量声明的名称, 应为合法的标识符

datatype: 表示声明变量的类型

initializer: 表示在常量声明和定义时, 分配给常量的初始值

## 枚举类

在VB.NET中, 使用Enum语句声明并定义一个枚举类及其成员的值

用于模块, 类, 结构, 过程, 块中

基本语法为 [<attributelist>] [<accessmodifier>] [Shadows]

Enum <enumerationname> [ AS <datatype> ]

<memberlist>

End Enum

attributelist: 指定应用于变量的属性列表

accessmodifier: 指定枚举的访问控制 Public / Protected / Friend / Private

Shadows: 表示枚举在基类中隐藏一个同名的编程元素

enumerationname: 枚举类声明的名称, 应为合法的标识符

datatype: 表示声明枚举类及其成员的数据类型

memberlist: 指明声明枚举类的成员常量

成员声明的基本语法为 [<attributelist>] <membername> [= <initializer>]

membername: 成员声明的名称, 应为合法的标识符

initializer: 在成员声明和定义时, 分配给成员的值

# VB.NET - P6

修饰符 (modifier), 在 VB.NET 中修饰符为通过任何编程元素添加的关键字  
keyword added with any programming element

特别强调编程元素将如何在程序中执行 或将被访问  
to give especial emphasis on how programming element  
will behave or will be accessed in the program

| modifier | description   |
|----------|---|
| Ansi     | 指定 Visual Basic 应该将所有字符串编组到美国国家标准协会(ANSI)值<br>不考虑正在声明的外部过程的名称   |
| Assembly | 指定源文件开头的属性适用于整个程序集  |
| Async    | 表示修改的方法或 lambda 表达式是异步的，称为异步方法(async method)<br>异步方法调用者可以恢复其工作，而不必等待异步方法完成  |
| Auto     | charset modifier part in declare statement<br>supply character set information for marshaling string<br>during call to the external procedure<br>also affect how Visual Basic search the external file<br>for the external procedure name |
| ByRef    | 指定 Visual Basic 应根据 .NET Framework 规则编组 (marshal) 字符串<br>指定参数通过引用传递，即调用过程可以更改调用代码中参数变量的值<br>can change value of variable underlying argument in calling code<br>声明 (Declare) statement<br>函数 (Function) statement<br>子过程 (Sub) statement  |
| ByVal    | 指定传递参数时，调用过程和属性不能更改调用代码中参数变量的值<br>Declare / Function / Sub Statement<br>运算符 (Operator Statement)<br>属性 (Property Statement)   |
| Default  | 表示属性作为类、结构、接口的默认属性  |
| Friend   | 指定声明的编程元素可以从包含其声明的程序集中访问，不仅是声明其的组件<br>declared programming element accessible from<br>within assembly that contain declarations<br>Friend access 通常是编程元素的首选级别 (preferred level)<br>Friend 是接口、模块、类、结构的默认调用级别 (default access level)       |

# VB.NET - P7

修饰符

## 修饰符

| modifier       | description   |
|----------------|---|
| In             | 用于通用接口和代理 (generic interface and delegate)  |
| Iterator       | 指定函数或 Get 访问器 (accessor) 是迭代器<br>iterator 对集合 (collection) 执行自定义迭代 (custom iteration)   |
| Key            | 允许为匿名类型 (anonymous type) 的属性指定行为  |
| Module         | 指定源文件开头的属性 (attribute at beginning of source file)<br>适用于当前装配模块 (current assembly module)                                     |
| MustInherit    | 不同于 Module 语句 (Module statement)  |
| MustOverride   | 指定一个类只能作为基类，而不能直接创建对象<br>指定属性或过程 (procedure) 未在当前类中实现<br>必须在派生类 (derived class) 中重写 (override)<br>类似于 C++ 中的 virtual 关键字      |
| Narrowing      | 表示转换运算符 (conversion operator) (Type<br>将类或结构转换为的类或结构<br>可能不能保存原始类或结构的某些可能值<br>(possible value of original class or structure) |
| NotInheritable | 指定一个类不可作为基类 (base class)<br>类似于 Java 中 final 修饰符作用于类  |
| NotOverridable | 指定属性或过程不可在派生类中被重写<br>类似于 Java 中 final 修饰符作用于方法  |
| Optional       | 指定过程的参数在调用过程时是可省略的  |
| Out            | 对于通用类型参数 (generic type parameter)<br>指定类型是协变的 (covariant)   |
| Overloads      | 指定属性或过程重新声明 (redclare)<br>已存在的一个或多个同名的属性或过程<br>(one or more existing property/procedure with same name)                       |
| Overridable    | 指定属性或过程可由派生类中的同名属性或过程重写   |
| Overrides      | 指定属性或过程重写从基类中继承的同名属性或过程   |
| ParamArray     | 允许将任意数量的参数传入指定过程<br>Param Array 传入的参数始终使用 By Val 声明   |
| Partial        | 表示类或结构的声明是类或结构的部份定义 (partial definition)  |
| ReadOnly       | 指定变量或属性可以读取但不可写入  |

# VB.NET - P8

ASP.NET - 基础知识

## 修饰符

|            |  |
|------------|--|
| modifier   | description  |
| Private    | 指定一个或多个编程元素只能在其声明上下文中访问<br>包括上下文包含的任何类型                |
| Protected  | 指定一个或多个编程元素只能从其所属的类或派生类中访问                             |
| Public     | 指定一个或多个编程元素没有访问限制                                      |
| Shadows    | 指定声明的编程元素在基类中<br>重新声明和隐藏相同命名的元素或一组重载元素                 |
| Shared     | 指定一个或多个声明的编程元素与类或结构关联<br>注意不是类或结构的特定实例                 |
| Static     | 指定一个或多个已声明的局部变量将继续存在<br>并在声明的过程终止后保留其最新值               |
| Unicode    | 指定 Visual Basic 将所有字符串编组为 Unicode 值<br>而不考虑正在声明的外部过程名称 |
| Widening   | 对于转换运算符 CType<br>将类或结构转换为可以保存原始类或结构的所有可能值的类型           |
| WithEvents | 指定一个或多个声明的成员变量引用可以引发事件的类的实例                            |
| WriteOnly  | 指定属性为可以写入但不可读取   |

语句 (statement) 在 Visual Basic 程序中的完整指令 (complete instruction)

可以包含关键字, 运算符, 变量, 字面值, 常量, 表达式

语句由关键词, operator, variable, literal value, constant, expression 构成

声明语句 (declaration statement), 命名变量 / 常量 / 过程的语句, 也可以指定数据类型

用于命名和定义过程, 变量, 属性, 数组, 常量

当声明编程元素时, 还可以定义数据类型, 访问级别, 范围

data type, access level, scope

可以声明的编程元素包括

变量 (variable), 常数 (constant), 枚举 (enumeration), 类 (class), 结构 (structure)

模块 (module), 接口 (interface), 过程 (procedure), 过程参数 (procedure parameter)

函数返回值 (function return), 外部过程引用 (external procedure reference)

运算符 (operator), 属性 (property), 事件 (event), 委托 (delegate)

以及参数 (parameters)

# VB.NET - P9

Section 9

语句 = statements

声明语句

Dim 语句：声明和分配一个或多个变量的存储空间  
如 Dim message As String = "Hello World!"

Const 语句：声明和定义一个或多个常量  
如 Const MAXIMUM As Long = 1024

枚举

Enum 语句：声明一个枚举并定义其成员的值

如 Enum Result  
Lose = 0  
Tie = 1  
Win = 2

End Enum

Class 语句：声明类的名称，并引入类包含的变量，属性，事件，过程的定义

如 Class Rectangle  
Private length As Double

Private width As Double

End Class

Structure 语句：声明结构的名称，并引入结构所包含的变量，属性，事件，过程的定义

如 Structure Box

Public length As Double

Public breadth As Double

Public height As Double

End Structure

Module 语句：声明模块的名称，并引入模块所包含的变量，属性，事件，过程的定义

如 Public Module myModule

Sub Main()

Dim user As String = InputBox("name")

MsgBox("Hello " & user & "!")

End Sub

End Module

# VB.NET - P10

声明语句

Interface 语句：声明接口的名称，并引入接口包含的成员定义

```
如 Public Interface MyInterface  
Sub doSomething()  
End Sub  
End Interface
```

Function 语句：声明函数过程的名称，参数，代码，返回值

```
如 Function circleArea ( ByVal radius As Double ) As Double  
Const PI As Double = 3.14  
Return PI * radius * radius  
End Function
```

Sub i语句：声明子程序过程的名称，参数，代码

```
如 Sub mySub ( ByVal s As String )  
MsgBox ( s )  
Return  
End Sub
```

Declare 语句：声明对外部文件中实现的过程的引用

```
如 Declare Function getUserName  
Lib "advapi32.dll"  
Alias "GetUserNameA"  
( ByVal lpBuffer As String,  
ByRef nSize As Integer ) As Integer
```

Operator 语句：对于在类或结构上定义的运算符过程 (operator procedure)

如 Public Shared Operator + ( ByVal x As Object, ByVal y As Object ) As Object

```
Dim r As New Object  
' implement code for r = x + y  
Return r  
End Operator
```

# VB.NET - P11

Unit 07 SQ

VB.NET - step 2

## 声明语句

Property 语句：声明属性的名称，以及用于存储和检索属性值的属性过程  
property procedure used to store and retrieve value of property

如 ReadOnly Property quote() As String

Get

Return quoteString

End Get

End Property

Event 语句：声明用户定义的事件 (user-defined event)

如 Public Event Finished()

Delegate 语句：用于声明一个委托

如 Delegate Function MathOperator(

ByVal x As Double,

ByVal y As Double

) As Double

## 语句

可执行语句 (executable statement)，启动动作的语句 (initiate action)

可以调用方法或函数 (call method or function)

循环或穿过代码块 (loop or branch through block of code)

将数值或表达式赋值给变量或常量

assign value / expression to variable / constant

也称为赋值语句 (Assignment statement)

## 指令

(directive) 在 VB.NET 中，提供了编译器指令 (compiler directive)

给编译器的指令，用于在实际编译开始前预处理信息

指令以井号 "#" 开头，且在指令所在的一行之前只能出现空格

特别注意，指令不是语句

在 VB.NET 中，并没有独立的预处理器 (separate preprocessor)

指令的作用类似于预处理器

在 VB.NET 中，编译器指令也用于帮助条件编译 (conditional compilation)

与 C/C++ 的指令相比，不用于创建宏 (create macro)

# VB.NET - P12

8/9 - 7/30 8/1



## 指令

#Const 指令：# 定义条件编译常量 (conditional compiler constant)

基本语法为：#Const <constname> = <expression>

<constname>：指定常量的名称

<expression>：字面值 (literal) 或其他条件编译常量

或包含任何或所有算术或逻辑运算符的组合

除了 Is 运算符

#ExternalSource 指令：用于指定特定行的源代码与源向外部文本之间的映射

indicate mapping between specific lines of source code

映射某行的源代码到外部文本

允许包括从外部代码文件的外部代码到源代码文件

include external code from external code file into source code file

基本语法为：#ExternalSource <String Literal>, <Int Literal>

[<Logical Line>]  
#End ExternalSource

<String Literal>：外部文件的路径

<Int Literal>：第一行的行号

<Logical Line>：发生错误的行 (line where error occurred)

#If...Then...#Else 指令：有条件地编译所选的 Visual Basic 代码块

基本语法为：#If <expression> Then

<statements>

[#ElseIf <expression> Then]

<statements>]\*

[#Else

<statements>]

#End If

#Region 指令：用于在 Visual Basic 文件中折叠和隐藏代码段

基本语法为：#Region <identifier\_string>

#End Region

# VB.NET - P13

算术运算符 在VB.NET中提供的算术运算符：加法(+)，减法(-)，乘法(\*)，取模(MOD)，幂(^)

除法(/)：返回一个浮点型的结果，如  $7/2 \rightarrow 3.5$

取商(\)：返回一个整型值的结果，如  $7\backslash 2 \rightarrow 3$

比较运算符 在VB.NET中提供的比较运算符：

LT(<)，LE(<=)，EQ(=)，NE(<>)，GE(>=)，GT(>)

特别注意，在VB.NET中，比较相等的运算符与赋值运算符同为一个等号(=)

Is：比较对象引用变量，判断是否引用完全相同的对象实例

IsNot：比较对象引用变量，判断是否引用不同的对象实例

Like：比较字符串(String)与模式(pattern)

位运算符 在VB.NET中提供的位运算符：

按位与(And)，按位或(Or)，按位非(Not)，按位异或(Xor)

左移(<<)，右移(>>)

逻辑运算符 在VB.NET中提供的逻辑运算符：

逻辑与(And)，逻辑或(Or)，逻辑非(Not)，逻辑异或(Xor)

注意在VB.NET中，And/Or/Not/Xor 既是位运算符，也是逻辑运算符

于是 And / Or 没有短路设计(short-circuiting)

即总是计算两个运算数的表达式

逻辑与(AndAlso)，逻辑或(OrElse)

对应于 And / Or，但只接受布尔类型，且有短路设计

确定为真(IsTrue)，确定为假(IsFalse)

赋值运算符 在VB.NET中提供了赋值运算符(=)，并且可与其他运算符组合使用

与算术运算符：+=，-=，\*=，/=，\=，^=

与位运算符：<<=，>>=

&=：将 String 表达式连接到 String 变量或属性  
concatenate String expression to String variable / property

AddressOf 返回过程的地址 (return address of procedure)

且传入的参数只有过程名 (name of method without parentheses)

注意不能用于变量或其他类型

# VB.NET - P14

Await

在 VB.NET 中，应用于异步方法或 lambda 表达式中的操作数  
applied to operand in asynchronous method or lambda expression

以暂停该方法的执行，直到等待的任务完成

to suspend execution of the method until the awaited task complete

) GetType 用于返回指定类型的 Type 对象 (return Type object for specified type)

Type 对象提供有关该类型的信息：属性 / 方法 / 事件

Type object provide information about the specified type  
property / method / event

如 GetType(Integer).ToString() → System.Int32

lambda 表达式。在 VB.NET 中，Function 表达式用于声明和定义 lambda 表达式的参数和代码

Function expression declare parameter and code define function lambda expression

如 Dim func = Function(ByVal num As Integer) num \* 2

func(11) → 22

If 表达式

在 VB.NET 中，提供了与 Racket 中类似的 If 表达式，有三个参数 / 两个修改两种形式

If(<condition As Boolean, <expression If True>, <expression If False>) As <result>

expressionIfTrue : evaluated and returned if <condition> evaluate to True

expressionIfFalse : evaluated and returned if <condition> evaluate to False

可见使用短路设计有条件地返回两个表达式的值之一

use short-circuit evaluation to conditionally return one of two values

注意与 Java 相比，VB.NET 的 If 表达式不要求两个表达式结果类型相同

如 Dim obj = If(True, 1L, "abc") → obj :: Object

If(<expression>, <expressionIfNothing>) As <result>

expressionIfNothing : evaluated and returned

类似于 C# 中的空合并运算符 (??)

注意二元 If 表达式 (binary If expression) 的第一个运算数有类型限制

可空类型 (nullable value type)

引用类型 (reference type)

无限制的通用类型 (unconstrained generic type)

# VB.NET - P15

## 运算符优先级

|   |                      |
|---|----------------------|
| Await   | Await                |
| Exponentiation (幂)                                  | ^                    |
| unary identity and negation (单目加减)                  | +,-                  |
| multiplication / floating-point division (乘法/浮点数除法) | *,/                  |
| integer division (整数除法)                             | \                    |
| modulus arithmetic (模运算)                            | Mod                  |
| addition / subtraction (加法/减法)                      | +,-                  |
| arithmetic bit shift (位移)                           | <<,>>                |
| comparison (比较)                                     | <=,<,>,>=, Is, IsNot |
| negation (按位非, 逻辑非)                                 | Not                  |
| conjunction (按位与, 逻辑与)                              | And, AndAlso         |
| inclusive disjunction (按位或, 逻辑或)                    | Or, OrElse           |
| exclusive disjunction (按位异或, 逻辑异或)                  | Xor                  |

条件结构 在VB.NET中, ~~if ... then~~ else if ... else 结构的基本语法结构为

If (<boolean\_expression>) Then  
    <statements>  
    [ ElseIf (<boolean\_expression>) Then ]\* 可以有0个或多个 ElseIf 判断  
    判断为真并执行后  
    [ Else                          ] 可以有0个或1个 Else 结构  
    ElseIf / Else  
    [ <statements> ]  
End If 注意VB.NET中必须有 End If 以指明 If 结构结束

Select

~~... case~~ 结构 在VB.NET中, ~~... case~~ 结构的基本语法结构为

Select [Case] <expression>  
    [ Case <expression-list> ]\* 多个表达式的情况下  
    [ <statements> ] 表达式子句间以逗号分隔  
    [ Case Else  
    [ <statements> ] ]类似于C++/Java中的 default 部分

End Select

条件 expression 必须为 VB.NET 中的基本数据类型 /Single/String/UInteger/ULong/UShort  
Boolean/Byte/Char/Date/Double/Decimal/Integer/Long/Object/SByte/Short

# VB.NET - P16

do 循环结构，在VB.NET中，提供 do 循环和 do...while 循环两种结构

do 循环的基本语法结构为

Do {While | Until} <condition>

<statements>

Loop

do...while 循环的基本语法结构为

Do

<statements>

Loop {While | Until} <condition>

While/Until 循环 当 / 直到 <condition> 计算的值为 True

) while 循环结构，在VB.NET中，while 循环结构的基本语法结构为

While <condition>

<statements>

End While

For...next 循环，在VB.NET中，for...next 循环结构的基本语法结构为

For <counter> [As <data-type>] = <start> To <end> [Step <step>]

<statements>

Next [<counter>]

counter：循环的迭代计数器

data-type：迭代计数器，起点，终点，步长的数据类型

start/end：迭代计数器的起点与终点

注意 end 本身也包含在循环中，即 1 To 5 → 1, 2, 3, 4, 5

step：步子步长，默认的值为 1

for each...next 循环，在VB.NET中，for each...next 循环结构的基本语法结构为

For Each <element> [As <data-type>] In <group>

<statements>

Next [<element>]

data-type：element 的数据类型

group：传入一个数据类型为 data-type 的迭代器 (iterator)

# VB.NET - P17

Exit 语句 在 VB.NET 中，用于将控制从过程 / 块立即传递到过程调用 / 块定义之后的语句  
终止循环 / 过程 / try 块 / [选择块]

基本语法为：

Exit {Do | For | Function | Property | Select | Sub | Try | While}

在嵌套循环中，Exit 语句将停止最内层循环的执行

Continue 语句 在 VB.NET 中，工作方式与 Exit 语句类似

使循环跳过其主体中的剩余部分的任何代码

强制循环的下一个迭代发生

并立即重新测试循环的控制条件

基本语法结构为

Continue {Do | For | While}

在 For...Next 循环中，Continue 语句在测试循环控制条件之前先执行递增步距

Goto 语句 在 VB.NET 中，用于将控制无条件地传递到过程中的特定行

基本语法结构为：Goto <label>

如 Sub Main()

...

line1:

...

Goto line1

...

End Sub

With...End With 语句，在 VB.NET 中，用于执行一系列反复引用同一个对象 / 结构的语句

基本语法结构为：With <object> <statements> End With

End With

通常用于在一个代码块中，频繁地使用同一个对象的情形

如 With cube

Console.WriteLine(c.length)

Console.WriteLine(c.width)

Console.WriteLine(c.height)

End With