

您的潜力，我们的动力

**Microsoft**  
微软(中国)有限公司

# 课程主题：VB.NET面向对象编程

讲师：蒋林峰

# 概要

- 定义类
- 创建和销毁对象
- 继承
- 接口
- 用类完成工作

# 定义类

- 定义类的过程
- 使用修饰符
- 声明方法
- 声明属性
- 使用Attribute(特征)
- 重载方法
- 构造函数的使用
- 析构函数的使用

# 定义类的过程

- 1** 在项目中添加类
- 2** 确定一个合适的类的名字
- 3** 根据需要创建构造函数
- 4** 如果有必要，创建析构函数
- 5** 定义属性
- 6** 定义方法和事件

# 使用修饰符

- 指定变量或者过程的可见性

| 关键字              | 定义                                   |
|------------------|--------------------------------------|
| Public           | 任意                                   |
| Private          | 只在当前类或者模块的内部可见                       |
| Friend           | 在当前类或者模块内部，以及同一程序集中的任意代码块中           |
| Protected        | 只有当前类以及继承自该类的子类中可见                   |
| Protected Friend | <b>Friend</b> 和 <b>Protected</b> 的结合 |

# 声明一个过程

- 语法

```
Public Sub TestIt(ByVal x As Integer)
...
End Sub

Public Function GetIt( ) As Integer
...
End Function
```



# 声明属性

- 声明属性语法

```
Public Property MyData( ) As Integer
    Get
        Return intMyData          'Return local variable value
    End Get
    Set (ByVal Value As Integer)
        intMyData = Value         'Store Value in local variable
    End Set
End Property
```

- ReadOnly, WriteOnly and Default 关键字

```
Public ReadOnly Property MyData( ) As Integer
    Get
        Return intMyData
    End Get
End Property
```

# 使用Attribute(特征)

- 通过<>提供的特殊数据
- 支持：
  - 程序集、类、方法、属性等
- 一般用来：
  - 程序集版本、Web服务、组件、安全和自定义设置

```
<Obsolete("Please use method M2")> Public Sub M1( )  
    'Results in warning in IDE when used by client code  
End Sub
```



# 重载方法

- 相同名称的方法，带有不同的参数类型

```
Public Function Display(s As String) As String
    MsgBox("String: " & s)
    Return "String"
End Sub

Public Function Display(i As Integer) As Integer
    MsgBox("Integer: " & i)
    Return 1
End Function
```

- 通过指定参数来确定调用哪个方法
- 除了重载继承而来的方法，一般可以忽略 **Overloads** 关键字。

# 使用构造函数

- Sub New() --- Class\_Initialize
- 当对象初始化的时候被执行

```
Public Sub New( )  
    'Perform simple initialization  
    intValue = 1  
End Sub
```

- 能够被重载，但是不能使用Overloads关键字

```
Public Sub New(ByVal i As Integer) 'Overloaded without Overloads  
    'Perform more complex initialization  
    intValue = i  
End Sub
```

# 使用析构函数

- Sub Finalize()---Class\_Terminate事件
- 用来释放资源
- 当对象被GC清除的时候被执行
  - Attention: 不一定马上执行

```
Protected Overrides Sub Finalize( )  
    'Can close connections or other resources  
    conn.Close  
End Sub
```

# 创建和销毁对象

- 实例化和初始化对象
- GC (Garbage Collection)
- 使用Dispose方法

# 实例化和初始化对象

- 示例:

```
'Declare but do not instantiate yet
Dim c1 As TestClass
'Other code
c1 = New TestClass( )           'Instantiate now

'Declare, instantiate & initialize using default constructor
Dim c2 As TestClass = New TestClass( )

'Declare, instantiate & initialize using default constructor
Dim c3 As New TestClass( )

'Declare, instantiate & initialize using alternative constructor
Dim c4 As New TestClass(10)
Dim c5 As TestClass = New TestClass(10)
```

# Garbage Collection

- 用来清理不使用变量的后台进程
- 通过 `X = Nothing`，使得X可以被清理
- 检测那些不可能被访问的对象或者内存
- 允许销毁对象
  - 并不确定什么时候发生
  - 释放一些频繁使用的资源（Connection、File）
  - 能够通过GC提供的系统类强制清理资源。



# 使用Dispose方法

- 创建Dispose方法手工的释放资源

```
'Class code  
Public Sub Dispose( )  
    'Check that the connection is still open  
    conn.Close          'Close a database connection  
End Sub
```

- 客户端调用Dispose方法

```
'Client code  
Dim x as TestClass = New TestClass( )  
...  
x.Dispose( )          'Call the object's dispose method
```

您的潜力，我们的动力

**Microsoft**<sup>®</sup>  
微软(中国)有限公司

# Demo: 创建类

## Creating the Customer Class

# 继承

- 什么是继承？
- Overrides
- 继承的例子
- 屏蔽
- 使用MyBase关键字
- 使用MyClass关键字

# 什么是继承？

- 一个类继承自一个基类
- 属性、方法、事件能够被继承
- 关键字
  - Inherits
  - NotInheritable
  - MustInherit
  - Protected

# Overrides

- 子类能够覆盖（替代）父类的方法
  - Overridable
  - MustOverride
  - Overrides
  - NotOverridable

# 继承例子

```
Public Class BaseClass  
    Public Overridable Sub OverrideMethod( )  
        MsgBox("Base OverrideMethod")  
    End Sub  
    Public Sub Other( )  
        MsgBox("Base Other method – not overridable")  
    End Sub  
End Class
```

```
Public Class DerivedClass  
    Inherits BaseClass  
    Public Overrides Sub OverrideMethod( )  
        MsgBox("Derived OverrideMethod")  
    End Sub  
End Class
```

```
Dim x As DerivedClass = New DerivedClass( )  
x.Other( )           'Displays "Base Other method – not overridable"  
x.OverrideMethod( ) 'Displays "Derived OverrideMethod"
```



# 屏蔽

- 隐藏基类的方法

```
Class aBase
    Public Sub M1( )    'Non-overridable by default
        ...
    End Sub
End Class

Class aShadowed
    Inherits aBase
    Public Shadows Sub M1(ByVal i As Integer)
        'Clients can only see this method
        ...
    End Sub
End Class
```

```
Dim x As New aShadowed( )
x.M1( )        'Generates an error
x.M1(20)       'No error
```

# 使用MyBase关键字

- 在子类中使用基类的方法成员
- 能够调用Public/Protected/Friend修饰符的方法
- 不是一个真正的对象

```
Public Class DerivedClass
    Inherits BaseClass

    Public Overrides Sub OverrideMethod( )
        MsgBox("Derived OverrideMethod")
        MyBase.OverrideMethod( )
    End Sub
End Class
```

# 使用MyClass关键字

- 确保基类的方法得到调用

```
Public Class BaseClass
    Public Overridable Sub OverrideMethod( )
        MsgBox("Base OverrideMethod")
    End Sub

    Public Sub Other( )
        MyClass.OverrideMethod( ) 'Will call above method
        OverrideMethod( )         'Will call derived method
    End Sub
End Class
```

```
Dim x As DerivedClass = New DerivedClass( )
x.Other( )
```

# Demo: Inheritance

您的潜力，我们的动力

**Microsoft®**  
微软(中国)有限公司

# 接口

- 定义接口
- 实现多态性

# 定义接口

- 定义了公共方法、属性和事件的一个签名
- 使用Interface关键词
- 作为类重载的一种方法

```
Interface IMyInterface  
    Function Method1(ByRef s As String) As Boolean  
    Sub Method2( )  
    Sub Method2(ByVal i As Integer)  
End Interface
```

- 接口也可以继承



# 实现多态性

- 多态性：
  - 多个类有相同的方法和属性
  - 调用者不需要知道对象是基于哪个类的
- 两种方法：
  - 接口
    - 类实现接口的成员
  - 继承
    - 子类重载父类的成员

您的潜力，我们的动力

**Microsoft®**  
微软(中国)有限公司

# Demo: 接口和多态性

# 用类完成工作

- 使用Shared数据成员
- 使用Shared方法成员
- 事件处理
- 什么是代理？
- 使用代理
- 比较类和结构

# 使用Shared数据成员

- 允许一个类的多个实例共享一个成员

```
Class SavingsAccount
    Public Shared InterestRate As Double
    Public Name As String, Balance As Double
    Sub New(ByVal strName As String, ByVal dblAmount As Double)
        Name = strName
        Balance = dblAmount
    End Sub
    Public Function CalculateInterest( ) As Double
        Return Balance * InterestRate
    End Function
End Class
```

```
SavingsAccount.InterestRate = 0.003
Dim acct1 As New SavingsAccount("Joe Howard", 10000)
MsgBox(acct1.CalculateInterest, , "Interest for " & acct1.Name)
```

# 使用**Shared**方法成员

- 使用不需要声明类实例
- 一般只处理**Shared**数据成员

```
'TestClass code  
Public Shared Function GetComputerName( ) As String  
    ...  
End Function
```

```
'Client code  
MsgBox(TestClass.GetComputerName( ))
```

# 事件处理

- 声明和触发事件：和VB6一致。
- WithEvents关键字：跟vb6一样处理事件
  - Vb.net中，**Handles**关键字用来指定处理事件的方法
- AddHandler关键字：允许动态关联处理方法和事件

```
Dim x As New TestClass( ), y As New TestClass( )  
AddHandler x.anEvent, AddressOf HandleEvent  
AddHandler y.anEvent, AddressOf HandleEvent  
...  
  
Sub HandleEvent(ByVal i As Integer)  
...  
End Sub
```

- RemoveHandler关键字：取消关联



# Demo: 事件处理

您的潜力，我们的动力

**Microsoft**  
微软(中国)有限公司

# 什么是代理

- 对象能够调用其它对象的方法
- 类似于C++中的指针
- 基于System.Delegate 引用类型
- 类型安全、可靠性、可管理
- 例子：
  - 调用程序和被调用程序中间的一个桥梁

# 使用代理

- **Delegate**关键字声明一个代理，包括参数和返回类型

```
Delegate Function CompareFunc( _  
    ByVal x As Integer, ByVal y As Integer) As Boolean
```

- 实际关联的方法必须有相同的参数和返回类型
- 调用代理调用方法

# 比较类和结构

| 类             | 结构            |
|---------------|---------------|
| 能定义数据成员、属性和方法 | 能定义数据成员、属性和方法 |
| 支持构造器，初始化成员   | 不支持           |
| 支持析构函数        | 不支持           |
| 通过继承可扩展       | 不支持           |
| 引用类型          | 值类型           |

# 回顾

- 定义类
- 创建和销毁对象
- 继承
- 接口
- 用类完成工作



您的潜力，我们的动力

**Microsoft®**

微软(中国)有限公司

**Microsoft®**