

计算机二级VB辅导:重载与隐藏计算机等级考试 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/645/2021_2022__E8_AE_A1_E7_AE_97_E6_9C_BA_E4_c97_645169.htm 大李坐在转椅上左右晃动着，手中的一个硬币在他指间灵巧地翻滚着。

“ hi, Henry。你对重写与重载的意思理解是怎么样子的呢？”我正盯着他手中的硬币发愣呢，“哦，重写，就是您刚才举的示例中，在派生类中用Overrides重新编写有Overridable标识的基类的方法或属性；重载么，就是我们可以用同样的名称，在一个类中用不同的参数列表来创建多个方法和属性，在调用时就可以适应不同参数类型的要求。”“在一个类中创建？”大李的左眉向上一挑，我就知道我可能说错了。但是，好象没有说错什么呀。“那好，你看一下，这样的写法会有什么结果？”

```
Public Class CBaseHenry Public Sub oldY()  
Console.WriteLine("基类的oldY方法") End Sub End Class Public  
Class CDerivedHenry Inherits CBaseHenry Public Overloads Sub  
oldY(ByVal j As Integer) Console.WriteLine("派生类的oldY方法")  
End Sub End Class Public Sub Main() Dim obj As CDerivedHenry  
= New CDerivedHenry() 然后大李写了obj.oldY这时出现的智能  
感知菜单上出现了参数列表：“咦，第二个oldY()无参数过  
程签名应该是基类定义的呀。为什么在派生类的实例中也会  
出现呀！”我不由好奇起来。“没错。还记得我上次曾显式  
编写构造函数的事吗？我用了一个mybase.new()用于继承下基  
类的性质。现在没有写，其实是一种默认的隐式调用。”大  
李一说我就明白了，其实现在的派生类应该是内含两个oldY  
方法了，无参数的oldY()其实就是基类的方法，是由
```

于Overloads和Mybase.new()双重作用的影响。 “ 那你再看这样的变化情况。 ” 大李象是看出我已经明白了这个问题。他在派生类中添加了一个用Overloads标识的与基类形式一致的无参数oldY()。 Public Class CBaseHenry Public Sub oldY()
Console.WriteLine("基类的oldY方法") End Sub End Class Public Class CDerivedHenry Inherits CBaseHenry Public Overloads Sub
oldY(ByVal j As Integer) Console.WriteLine("派生类的oldY方法1") End Sub Public Sub oldY() Console.WriteLine("派生类的oldY方法2") End Sub End Class Public Sub Main() Dim obj As CDerivedHenry = New CDerivedHenry() obj.oldY() End Sub “ 好的，现在你再说说看，现在的obj.oldY()的运行结果会打印出什么？ ” 大李手按在F5运行键上问我。 “ 应该是派生类的，不对，基类，呃，派生类..... ” 我一下子就晕了。 “ 呵呵。 ” 大李也不禁发笑起来。然后点击了一下F5键。结果是：派生类的oldY方法2 “ 这就叫隐藏，我们用overloads方式，隐藏了基类的同名方法。以防用户发生混淆。一般来说，隐藏有两种情况，一种是通过范围来实现。比如你定义一个全局变量x，但在一个方法中，你又定义了一个局部变量x，在方法中使用x时，使用的是局部变量的那一个，也就是用局部变量x在方法中隐藏了全局变量x。另一种情况，就是通过继承来隐藏，方法么，除了刚才用的overloads，还可以用Shadows关键字来实现。 ” “ Shadows？阴影？倒是很贴切的名字。这个是怎么用的呢？ ” 我兴趣由然而生。 “ Shadows功能很强的。 ” 说着，大李又开始修改刚才的代码了。 Public Class CBaseHenry Public Sub oldY() Console.WriteLine("基类的oldY方法") End Sub End Class Public Class CDerivedHenry Inherits

```
CBaseHenry Public Shadows Sub oldY(ByVal j As Integer)
Console.WriteLine("派生类的oldY方法") End Sub End Class
```

```
Public Sub Main() Dim obj As CDerivedHenry = New
```

CDerivedHenry() 再写到obj.oldY时，出现的智能感知菜单上就剩下一个方法了。“哈，真的呢，基类的方法看不到了。但是，这和重写的效果岂不是一样了吗？”我不由地又想起一个问题，如果是一样的作用，要Shadows干什么呀。“还是有区别的。”大李开始扳着指头数起来，“最明显的区别在于，隐藏适用于任何元素类型，你可以在派生类中用Public Shadows oldY as Integer来隐藏基类的oldY()方法；而重写只能适用于方法与属性，而且声明参数与修饰符都要求完全一致。”“还要注意一点，当派生类又派生出一个子类时，重写与隐藏都会被继承下去。当然，如果在派生类中是用private来修饰Shadows成员的话，它的子类就会继承它基类的成员了。”“我倒是感觉，现在VB.NET对继承的处理功能真的很强大，有继承，对于编程者来说，真的是件好事。”我不由感叹道。“不错，继承层次体系是有很多好处，但是事物总是有其双面性，继承也有不少问题，其中最麻烦的，就是‘脆弱的基类’。”大李紧锁眉头认真地说。“哦，什么叫‘脆弱的基类’？”我不解地问。

编辑特别推荐: 全国计算机等级考试二级VB模拟试题及答案解析 100Test 下载频道开通，各类考试题目直接下载。详细请访问 www.100test.com