

关于VisualBasic9.0的动态标识符 PDF转换可能丢失图片或格式，建议阅读原文

[https://www.100test.com/kao\\_ti2020/257/2021\\_2022\\_\\_E5\\_85\\_B3\\_E4\\_BA\\_8EVisu\\_c67\\_257591.htm](https://www.100test.com/kao_ti2020/257/2021_2022__E5_85_B3_E4_BA_8EVisu_c67_257591.htm) 动态语言今年来发展很快，各种脚本语言多少都带有些动态语言的特点。动态语言以其灵活的语法和丰富的运行时行为赢得了许多人的青睐。在.net阵营，C#一直扮演着静态语言的角色，尽管3.0的语法改变很激进，但仍注重于编译时的类型检查和约束。动态特性用的不好会产生更多运行时问题，不易确保程序的可靠性，但其提高开发效率的作用是毋庸置疑的。Visual Basic 9.0是Visual Studio中唯一能让你统领动态和静态两大领域的语言，让你在程序中自由选择喜欢的风格。Visual Basic从1.0便支持后期绑定，即按照对象的运行时类型信息使用对象，而不是定义这个对象时的类型信息。如：`Dim I As Object = New ArrayList()``I.Add(12)``I.RemoveAt(0)` 我们知道Object没有Add或RemoveAt方法，但运行时类型ArrayList具有他们，因此Visual Basic允许到运行时再检查有无相应的操作并支持。既然这个过程已经是在运行时做的了，我们就应该允许这个后期绑定的操作本身也是在运行时确定。这个语法增强叫做动态标识符。比如我们可以和这样写：`Dim I As Object = New ArrayList()``I("Add")(12)``I("RemoveAt")(0)` 我们看到，这个时候方法名称Add和RemoveAt已经是字符串了，它当然也可以是包含字符串的变量!所以你甚至可以写出这样的代码：`I(Console.ReadLine())(12)``I("Remove" & "At")(0)` 这样，作为方法名称的标识符就可以完全在动态期间确定了。方法名称如此，那么类型名称呢?理应如此。我们可以这样写：`Dim`

`l As Object = New ("ArrayList")()` 这样，你的变量l的类型将根据一个运行时字符串的值确定。动态标识符还能将函数的参数也动态化，假设有下列子程序：`Sub Method1(Optional arg1 As String, Optional arg2 As String)` 运行的时候，我们可以动态地指定将数值赋给哪个参数：`If a > 0 Then argString = "arg1" Else argString = "arg2" Method1({argString}:= a.ToString())` 注意在:=符号之前的大括号，表示对参数名称的动态结合。有了动态标识符这一特性，Visual Basic 9.0将不需要利用反射的复杂语法，就可以做很多需要反射和运行时类型信息来做的任务，这使得Visual Basic成为解决你手边小问题的最佳帮手. 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](http://www.100test.com)