

VisualBasic9.0前沿播报动态篇（二）动态接口 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/220/2021_2022_VisualBasi_c97_220772.htm 计算机等级考试训练软件《百宝箱》在谈到动态语言与静态语言的区别时，有一句很经典的话：Static typing when possible, dynamic typing when needed。Visual Basic支持后期绑定和动态标识符，因此完全可以按照对象的运行时类型进行操作。只是后期绑定或动态标识符缺乏对参数或返回值类型的约束，仅仅按照成员的名称进行操作。举一个例子：在静态篇我们介绍过匿名类型，假设有3个匿名类型的实例（Tuple）：
`Dim a = New { Name := "Robbins", Age := 25 }`
`Dim b = New { Name := "Andrei", ID := 12, Address := "Beijing" }`
`Dim c = New { Name := New NameType("Mark"), Title := "PM" }`如你所见，三个Tuple具有不同的类型，由于匿名类型没有办法实现接口，因此没有办法用统一的语法操作a和b相同的属性Name。用后期绑定可以统一访问Name属性，但是c的Name属性与a，b的不同，但是后期绑定却能让他们用统一的方式访问，这就让动态的隐患扩大了。有没有既可以动态访问成员，又有部分强类型的约束呢？Visual Basic 9.0引入了动态接口。动态接口与普通接口不同，类型无需声明实现动态接口，只要具有相应动态接口定义的成员即可转换为动态接口类型。这个特性也称为鸭子类型判定。因为流传这样一种动态语言类型判断的手法：如果一个东西走路像鸭子，说话也像鸭子，那它就是鸭子。动态接口的定义需要用Dynamic修饰符

`: Dynamic Interface IHasName Property Name As StringEnd Interface` 这个接口规定了一个String类型的属性，名为Name。

任何类型的实例，只要具有String类型的属性Name，就被推定为实现IHasName：
`Dim a = New { Name := "Robbins", Age := 25 }`
`Dim b = New Button() { Name := "Button1", Text := "Button1" }`
`Dim i As IHasName = a`
`Dim j As IHasName = b`
`i.Name = "Howard"`
`j.Name = "Button2"`
尽管a的匿名类型和Button类型都没有声明实现IHasName接口，但他们都有类型为String名为Name的属性，因此都可以使用IHasName接口。动态接口提供了有类型约束和IDE智能感知的后期绑定，再次提高了Visual Basic动态编程的能力。不过要提示一点，动态接口并非类型安全性的特性。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