**.NET程序集初级-反射**

**反射是.NET用于动态编程获取和使用程序集的元数据类型的过程，能够显示和ildasm.exe相同的元数据类型。反射需要导入命名空间System.Reflection。**

* **反射能做什么？**
* 反射能够获取程序集的类型，以展示类的细节，即类的自定义浏览器：
* 方法：GetMethods()，包括方法名、参数、返回值
* 属性：GetProperties()
* 字段：GetFields()
* 接口：GetInterfaces()
* 事件：GetEvents()

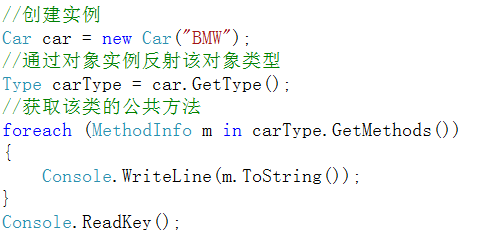
注意，只有Public公共的细节才能被获取。这也很容易理解，毕竟类被封装的私有部分在运行时是被隐藏的。但ildasm.exe可以看到所有的细节，包括私有部分。

* **晚期绑定**

晚期绑定指使用System.Activator类获取未引用程序集反射类型的实例。

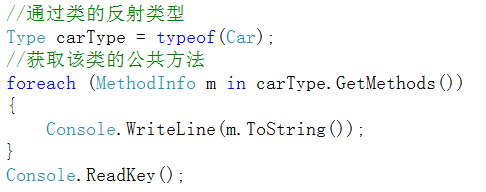
* **如何通过反射获取已加载程序集的类型引用Syetem.Type？**
* 在实例上反射类型

使用System.Object.GetType()反射类型引用：



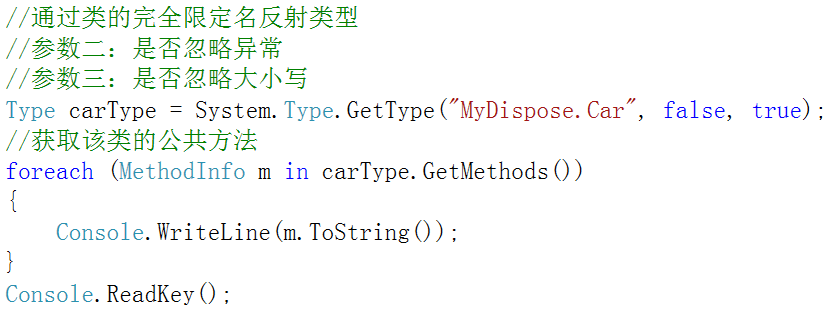
* 用类反射类型

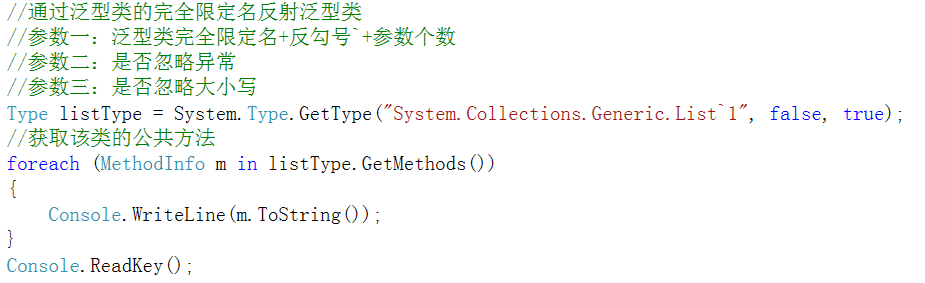
使用typeof()反射类型引用：



* 用类的完全限定名反射类型

使用System.Type.GetType()反射类型引用：

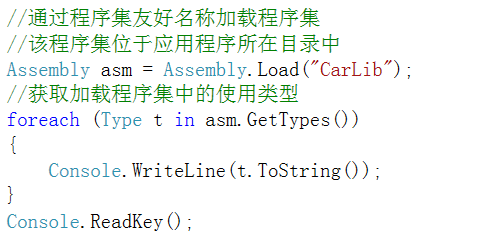




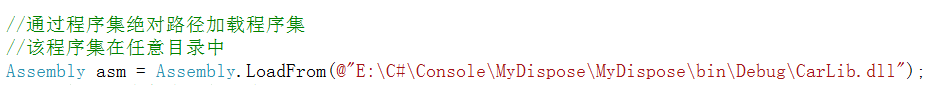
* **如何通过反射获取未加载程序集的类型引用Syetem.Type？**

使用动态程序集加载类Assembly加载目标程序集，并通过Assembly.GetTypes()反射类型引用:

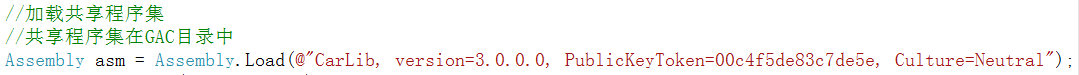
* 使用位于应用程序所在目录里面的程序集：



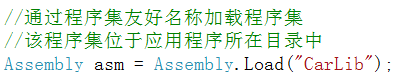
* 使用任意位置的程序集：



* 使用共享程序集：



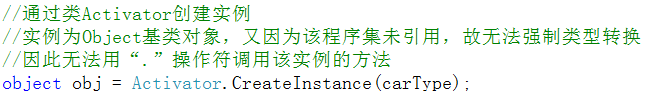
* **如何在晚期绑定创建反射类型的实例？**
* 加载程序集：

****

* 通过反射获取指定的类型引用：



* 使用类Activator创建实例：



* **如何调用晚期绑定创建反射类型的实例的方法？**

由于晚期绑定创建的实例是Object的实例，因此无法用“.”来

获取反射类型对象的成员，但可以通过反射获取：

