**[C# using 三种使用方式](http://www.cnblogs.com/fashui/archive/2011/09/29/2195061.html)**

1.using指令。using + 命名空间名字，这样可以在程序中直接用命令空间中的类型，而不必指定类型的详细命名空间，类似于Java的import，这个功能也是最常用的，几乎每个cs的程序都会用到。   
例如：using System; 一般都会出现在\*.cs中。

2.using别名。using + 别名 = 包括详细命名空间信息的具体的类型。   
这种做法有个好处就是当同一个cs引用了两个不同的命名空间，但两个命名空间都包括了一个相同名字的类型的时候。当需要用到这个类型的时候，就每个地方都要用详细命名空间的办法来区分这些相同名字的类型。而用别名的方法会更简洁，用到哪个类就给哪个类做别名声明就可以了。注意：并不是说两个名字重复，给其中一个用了别名，另外一个就不需要用别名了，如果两个都要使用，则两个都需要用using来定义别名的。

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54 | using System;  using aClass = NameSpace1.MyClass;  using bClass = NameSpace2.MyClass;    namespace NameSpace1  {      public class MyClass      {          public override string ToString()          {              return "You are in NameSpace1.MyClass";          }      }  }    namespace NameSpace2  {      class MyClass      {          public override string ToString()          {              return "You are in NameSpace2.MyClass";          }      }  }    namespace testUsing  {      using NameSpace1;      using NameSpace2;      /// <summary>      /// Class1 的摘要说明。      /// </summary>      class Class1      {          /// <summary>          /// 应用程序的主入口点。          /// </summary>          [STAThread]          static void Main(string[] args)          {              //              // TODO: 在此处添加代码以启动应用程序              //                aClass my1 = new aClass();              Console.WriteLine(my1);              bClass my2 = new bClass();              Console.WriteLine(my2);              Console.WriteLine("Press any key");              Console.Read();          }      }  } |

3.using语句，定义一个范围，在范围结束时处理对象。   
场景：   
当在某个代码段中使用了类的实例，而希望无论因为什么原因，只要离开了这个代码段就自动调用这个类实例的Dispose。   
要达到这样的目的，用try...catch来捕捉异常也是可以的，但用using也很方便。

|  |  |
| --- | --- |
| 1  2  3  4 | using (Class1 cls1 = new Class1(), cls2 = new Class1())  {    // the code using cls1, cls2  } // call the Dispose on cls1 and cls2 |