

# 定义封装的类类型

面向对象的三个基本特征是封装、继承、多态。

概念：封装是实现面向对象程序设计的第一步，封装就是将数据或函数等集合在一个个的单元中( 我们称之为类 )。被封装的对象通常被称为抽象数据类型。

意义：封装的意义在于保护或者防止代码（数据）被我们无意中破坏。防止对实现细节的访问。

我们只是提供调用类的方法，而调用者不必了解到类内部怎样处理相关数据。

C#中通常将方法或者其他数据成员封装在一个类中，具体地，封装使用访问修饰符来实现。一个访问修饰符定义了一个类成员的范围和可见性。

访问修饰符：

public、private、protected、internal、protected internal

public 公开的

允许一个类将其成员变量和成员函数暴露给其他的函数和对象。任何公有成员可以被外部的类访问。

private 只能在当前类的内部访问 类成员的默认访问修饰符

允许一个类将其成员变量和成员函数对其他的函数和对象进行隐藏。只有同一个类中的函数可以访问它的私有成员。即使是类的实例也不能访问它的私有成员。

protected 受保护的，只能在当前类内部及子类中访问

允许子类访问它的基类的成员变量和成员函数。这样有助于实现继承。

internal 只能在当前项目中访问，在同一个项目中，internal 和 public 的权限是一样的。

允许一个类将其成员变量和成员函数暴露给当前程序中的其他函数和对象。  
换句话说,带有 internal 访问符的任何成员可以被定义在该成员所定义的应用程序内的任何类或方法访问。

protected internal

允许一个类将其成员变量和成员函数对同一应用程序内的子类以外的其他的类对象和函数进行隐藏。这也被用于实现继承。

1.能够修饰类的访问修饰符只有两个: public、internal。

2.可访问性不一致

子类的访问权限不能高于父类的访问权限。