

C#的修饰符

C#修饰符之类修饰符：public、internal、partial、abstract、sealed、static

C#修饰符之成员修饰符：public、protected、private、internal、sealed、abstract、virtual、override、readonly、const

c#里面一共有五种访问修饰符，五种中每个种的访问权限都是不一样的

public：（可以修饰类也可以修饰成员）公共的，**最高的访问级别**，访问没有限制

Internal：（可以修饰类也可以修饰成员）内部的，**类和方法的默认访问修饰符是internal**，只有在本程序集内的成员可以访问，其他程序集或站点引用其所在的程序集无法访问此类。例如程序集LibraryA写有ClassA,程序集LibraryB引用LibraryA,在LibraryB的类或方法里无法通过LibraryA调用ClassA

Partial：（它只修饰类），部分的，可以将一个类分成几部分写在不同文件中，最终编译时将合并成一个文件，且各个部分不能分散在不同程序集中

Abstract：（可以修饰类也可以修饰成员）抽象的，修饰类的时候表示该类为抽象类，不能够创建该类的实例。修饰方法的时候表示该方法需要由子类来实现，如果子类没有实现该方法那么子类同样是抽象类；且含有抽象方法的类一定是抽象类

Sealed：（可以修饰类也可以修饰成员）密封的，修饰类时表示该类不能够被继承，修饰方法时表示该方法不能被重写【sealed相当于java中的final修饰符】

Static：（可以修饰类也可以修饰成员）静态的，修饰类时表示该类是静态类，不能实例化该类的对象，既然不能实例化该类，那么这个类也就不能含有对象成员，即该类所有成员为静态；类成员只能通过【类.成员名】的方式访问。当static修饰构造函数时，构造函数不能含有任何参数，不能含有修饰符，构造函数不能对对象成员进行初始化操作。但是能够对静态成员进行初始化或者调用。不能保证他在什么时候执行，却能保证在第一次使用类型前执行。在静态构造函数中初始化的静态成员为最终初始化结果

private：（只修饰成员）私有的，**最低的访问级别**，被private访问修饰符修饰的成员只有在当前类的内部可以访问，其他地方一律不得访问

protected：（只修饰成员）受保护的，**只有在当前类内部及所有的子类内部可以访问**。只有继承父类后，才可以调用它的属性

Virtual：修饰方法成员，表示虚方法。父类可以含有该类的实现，子类可以覆写该函数。

Override：表示该方法为覆写了父类的方法。

ReadOnly：修饰字段，表示该字段为只读字段。readonly是运行时只读，内容在运行时确定，所以修改了readonly类型成员后无需重新编译即可生效，ReadOnly不能修饰局部变量

Const: 修饰字段，表示该字段为只读字段。const修饰的字段在编译时必须能够明确知道该字段的值，其值是硬编码到程序中去，修改了该类型成员后需要重新编译才能使修改生效，const可以修饰局部变量