内部类

Java的四种内部类包括如下：

成员内部类 静态内部类 局部内部类 匿名内部类

内部类的好处：

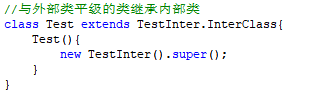
1.内部类可以很好的实现隐藏

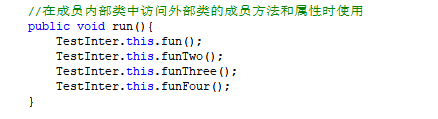
一般的非内部类，是不允许有 private 与protected权限的，但内部类可以

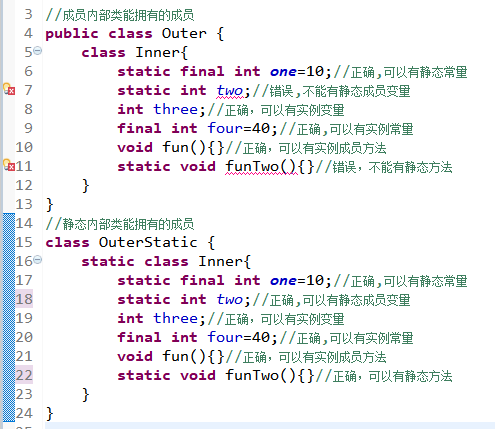
2．内部类拥有外围类的所有元素的访问权限

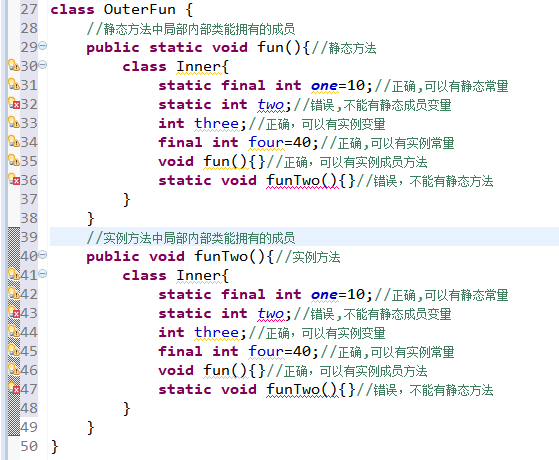
3.每个内部类都能独立的继承一个接口的实现，所以无论外部类是否已经继承了某个(接口的)实现，对于内部类都没有影响。内部类使得多继承的解决方案变得完整，







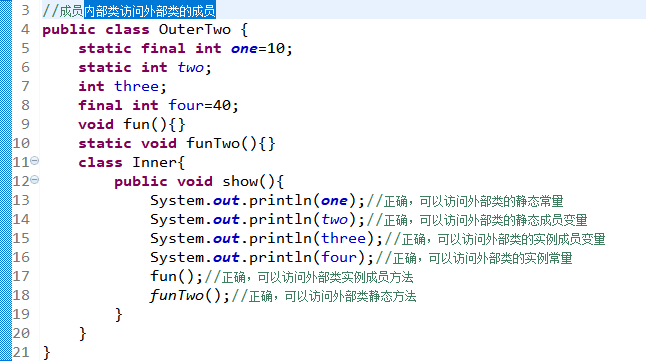


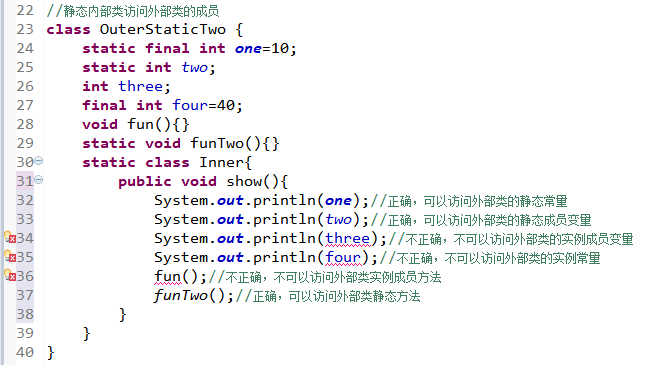


**内部类能拥有的成员**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 说明 | 静态常量 | 静态变量 | 实例变量 | 实例常量 | 静态方法 | 实例方法 |
| 成员内部类 | √ |  | √ | √ |  | √ |
| 静态内部类 | √ | √ | √ | √ | √ | √ |
| 局部内部类 | √ |  | √ | √ |  | √ |

**内部类访问外部类的成员时的限制**

****

****



|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 说明 | 静态常量 | 静态变量 | 实例变量 | 实例常量 | 静态方法 | 实例方法 |
| 静态内部类 | √ | √ | √ | √ | √ | √ |
| 成员内部类 | √ | √ |  |  | √ |  |
| 实例方法内部类 | √ | √ | √ | √ | √ | √ |
| 静态方法内部类 | √ | √ |  |  | √ |  |

**成员内部类：**

1. 定义在另一个类(外部类)的内部，而且与成员方法和属性平级叫成员内部类
2. 成员内部类中不能存在static关键字，即，不能声明静态属性、静态方法、静态代码块等，但是可以声明静态常量
3. 创建成员内部类的实例使用：外部类名.内部类名 实例名 = 外部类实例名.new 内部类构造方法(参数)
4. 在成员内部类中访问外部类的成员方法和属性时使用：外部类名.this.成员方法/属性。
5. 内部类在编译之后生成一个单独的class文件，里面包含该类的定义，所以内部类中定义的方法和变量可以跟父类的方法和变量相同。
6. 外部类无法直接访问成员内部类的方法和属性，需要通过内部类的一个实例来访问。
7. 与外部类平级的类继承内部类时，其构造方法中需要传入父类的实例对象。且在构造方法的第一句调用“外部类实例名.super(内部类参数)”。

**静态内部类：**

静态内部类与静态变量和静态方法定义的位置一样，也带有static关键字，只是它定义的是类，静态内部类除了位置放在别的类内部外，它与一个独立的类差别不大，可以有静态变量、静态方法、成员方法、成员变量、构造方法等，它可以访问外部类的静态变量和方法，但不可以访问实例变量和方法。在类内部，可以直接使用内部静态内部类：例如：外部类为Outer，静态内部类为StaticInner，在Outer类中可以直接使用StaticInner，StaticInner si = new StaticInner();也可以通过:外部类.静态内部类:Outer.StaticInner si = new Outer.StaticInner();

**方法内部类**

1. 方法内部类只能在定义的方法内被使用。如果方法是实例方法，则除了静态变量和方法，内部类还可以直接访问外部类的实例变量和方法,如果方法是静态方法，则方法内部类只能访问外部类的静态变量和方法。
2. 方法内部类还可以直接访问方法的参数和方法中的局部变量，不过，这些变量必须被声明为final,
3. 可以使用abstract修饰，声明为抽象类。

**匿名内部类**

1. 只能使用一次，创建实例之后，类定义会立即消失
2. 必须继承一个类（抽象的、非抽象的都可以）或者实现一个接口。如果父类（或者父接口）是抽象类，则匿名内部类必须实现其所有抽象方法。
3. 不能是抽象类，因为匿名内部类在定义之后，会立即创建一个实例。
4. 不能定义构造方法，匿名内部类没有类名，无法定义构造方法，但是，匿名内部类拥有与父类相同的所有构造方法。
5. 可以定义代码块，用于实例的初始化，但是不能定义静态代码块。
6. 可以定义新的方法和属性(不能使用static修饰)，但是无法显式的通过“实例名.方法名(参数)”的形式调用，因为使用new创建的是“上转型对象”（即父类声明指向子类对象）。
7. 是局部内部类，所以要符合局部内部类的要求。