# Java类修饰符

## 一  类的修饰符

java中的文件结构由大到小为：一个工程，一个工程下可以有许多包，每个包中可以有许多类。

类的修饰符分为         访问权限修饰符          和             非访问权限修饰符

（1）访问权限修饰符：

1 public 公共类修饰符，public修饰的类可以被同一个包中的类访问，也可以被其他包中的类访问。即在此工程下的所有类都可以访问被public修饰的类。Java语言规定，程序的主类必须是公共类。

例如  public class Animal {...}

2 默认说明。某个类前面没有public修饰，则认为该类使用默认说明，这样的类称为友好类，这样的类只能被同一个包中的类使用，不能被其他包中的类使用。

例如  class Animal {...}

(2)非访问权限修饰符：

1  abstract 抽象类修饰符。用abstract修饰的类是抽象类。其他的类会继承抽象类并实现抽象类中的抽象方法。抽象类不能用new运算符创建对象。

2  final 终结类修饰符，用final修饰的类表示终结类，被final修饰的类不能被继承，而用abstract修饰的类是用来被其他类继承的，所以一个类不可能同时被abstract和final修饰。

## 二 类中成员变量和成员方法的修饰符

### 成员变量修饰符

#### (1)访问权限修饰符：

1  public 公有访问修饰符。该修饰符修饰的变量称为公有变量，如果公有变量又在一个公有类（被public修饰的类）中，那么这个变量可以被所有包中的所有类访问；

如果共有变量在友好类（没有被public修饰的类）中，那么该变量可以被它   所在的类   所在的包中的所有类访问。

2 protected 保护访问修饰符。 该修饰符修饰的成员变量若在一个公有类中，那么它可以被所在的类本身，同一个包中的所有类，其他包中该类的子类访问。

3 默认访问修饰符。如果成员变量前没有访问修饰符，那么它为友好成员，他可以被同一个包中的所有类访问。

4 private 私有访问修饰符。 该修饰符修饰的成员只能被他所在的类访问，任何其他的类都不能访问，包括它的子类。在实际项目中，最好把一个类的实例变量（不被static修饰的变量）设置为private，并在方法中设置setXXX() 和 getXXX()这样的方法进行访问。这样做有助于对客户隐蔽类的实现细节，减少错误，提高程序可修改性。

#### （2）非访问权限修饰符

1 static 修饰的变量称为静态变量，静态变量是属于类的。可以用类名或是该类的对象名引用静态变量

2 final 修饰的变量称为最终变量，实际上就是java中的符号常量

3 volatile 修饰的变量称为易失变量，这种变量可以被多个线程所控制和修改。

### 成员方法的修饰符

#### （1） 访问权限修饰符

1 公有访问控制符 public 使用public修饰的方法可以作为该类对外的接口

2 默认访问控制符。具有包访问属性，可以被同一个包内的其他类访问。

3 保护访问控制符 protected 。使用该访问修饰符的方法可以被该类自身，同一个包内其他类，其他包中该类的子类访问。

4 私有访问控制符 private。使用该访问控制符的方法是能被该类自身访问，任何其他类不能访问，包括该类的子类。

#### （2）非访问权限修饰符

1 abstract 用abstract修饰的方法为抽象方法，抽象方法只有类的声明，没有类的实现。

2 static 用static修饰的方法为静态方法，静态方法属于整个类。

3 final 用final修饰的方法为最终方法，最终方法不能被当前类的子类重新定义。

4 native 用native修饰的方法为本地方法，本类方法通常用其他语言书写方法体，并实现方法功能。

5 synchronized 用synchronized修饰的方法为同步方法，同步方法主要用于多线程共存的程序中的协调和同步。

# Java接口修饰符

Java的interface中，成员变量的默认修饰符为：public static final  
所以我们在interface中定义成员变量的时候，可以  
1：public static final String name = "张三";  
  
2：String name = "张三";  
  
以上两种都可以，老司机一般都是第二种。既然是静态最终的变量，也就意味着在外面访问的时候不能修改这个成员变量的值。所以在接口中定义成员变量的，一般都是常量。不会修改的。如果要进行修改的话，定义在接口具体实现类中。  
说完成员变量的默认修饰符，顺便也提下方法的默认修饰符，方法的默认修饰符是：public abstract  
即：公共抽象的，就是用来被实现该接口的类去实现该方法。所以在接口中定义方法时候，也有两种方式  
1：public abstract List<String> getUserNames(Long companyId);  
  
2：List<String> getUserNames(Long companyId);  
  
同样老司机都是第二种。