包的定义：

在Java文件的行首

需要用到其他包内的类的时候，需要导入包

1.使用指定类导入方式

import 包名.类名

2.使用\*（只有在代码中真正使用到的时候才会加载）

import 包名.\*

类使用class和public class的区别：public class 定义的类名必须与Java文件名一致，且其他包可访问

class无文件名要求，此类不能被其他包访问

3.静态导入（需访问其他包中的静态属性或方法）

import static 包名.类名.\*

起包名规则：

1. 组织域名的反转：eg:google.com-->com.google
2. 个人开源项目：github.com-->com.github.sweeeeeet//
3. 特殊字符，空格不使用，使用a--z,0--9,字母小写

桌面程序：

xml+html+javascript+java=javaFX

移动程序：

xml+java+**android API**=Android APP

访问控制权限：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 范围 | private | default | protected | public |
| 同包中同类 | √ | √ | √ | √ |
| 同包中不同类 |  | √ | √ | √ |
| 不同包的子类 |  |  | √ | √ |
| 不同包的非子类 |  |  |  | √ |

权限的选择:

有子类继承的类,将其属性设计为protected

private [default] protected 根据现实情况设计

jar命令:

jar本质上也是一种压缩文件，里面保存的都是\*.class文件

单例设计模式:

1.将构造方法私有化,这样就外部就不能使用new 随意创建对象,2.此时提供public的静态方法,返回实例对象3.类的内部实例化对象(final)

特点:

1.一个类只允许实例化一个对象

2.立即初始化(饿汉式单例),立即加载,线程安全,内存可能会浪费

懒汉式单例:静态私有对象不初始化,在静态方法中初始化,延迟初始化时间,但是是线程不安全的,节省了内存空间

单例的八种模式:

1. 饿汉式
2. 懒汉式(线程安全版,必须掌握)
3. 静态内部类
4. 枚举

枚举类默认含有私有构造方法,指定实例对象需在构造方法之前,用逗号分隔

枚举类同样可指定成员属性,通常属性在构造时初始化,构造方法的参数列表需与实例对象对应

访问枚举类 的实例对象通过类名直接访问:类名.实例对象名

获取枚举类的所有实例化对象的对象名:类名.values();(用对象数组接受)

通过枚举的实例化对象名获取对象:类名.valueOf(name);