# 1.包

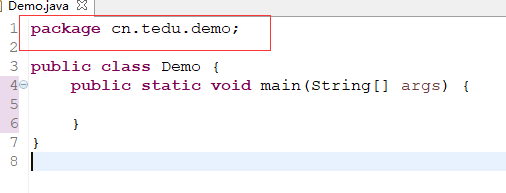
## 声明语句

声明包用的package，包的产生是为了解决同名文件的问题。

声明格式如下：

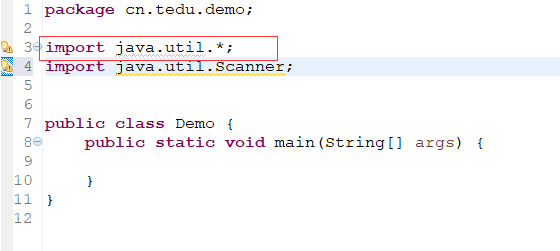
package m.n.c

一个Java文件中只允许存在一个package语句，而且这个package语句必须放在整个Java文件的首行。



## 导包语句

导入包用的是import。



import java.util.\* //表示导入util包下的所有的类而不包括子包下的类。

//\*是一个通配符。

## 分类

java.lang：核心包（核心类库），在程序启动的时候就已经加载到了方法区中了，在使用的时候不需要导包。

java.util —— 工具包，存放了一系列简化操作的工具类。

java.io —— 数据传输。

java.nio —— 高并发。

java.net —— 网络传输。

java.math —— 数学运算。

java.sql —— 数据库。

java.awt —— 图形用户界面。（几乎不用）

java.security —— 数据安全。

java.text —— 格式化。

javax.\* —— 扩展包。

org.\* —— 第三方厂商和开源社区提供的包。

同包类和java.lang包不用写导包语句。

注意：包名不能以java,javax,org开头

# 2.垃圾分代回收机制

垃圾回收针对的是堆内存。

对象在堆内存中存储，对象在使用完成之后会在不定的某个时刻被垃圾回收器（GC --- Garbage Collector）解析掉。现阶段回收过程无法手动控制。

当调用构造方法的时候，创建好一个对象，因为Java中对每种数据类型都明确给定了大小，在创建对象的时候，会自动计算大小分配内存，所以在内存的回收和释放的时候也是由Java自己管理。

堆内存分为了新生代/年轻代和老生代。新生代划分为伊甸园区和幸存区。一个对象新创建是放到了伊甸园区，如果这个对象在栈内存中没有引用，那么会在扫描的时候被解析，释放内存；在伊甸园区经过了一次回收如果依然存活则标记到幸存区。幸存区的扫描频率要略低于伊甸园区。如果在幸存区中经过了多次扫描这个对象依然没有被解析，则标记到老生代。老生代的扫描频率要远远低于新生代。**如果老生代的对象发生了回收，导致程序的卡顿甚至崩溃。**

发生在新生代的回收称之为minor gc，即初代回收。

发生在老生代的回收称之为full gc，即完全回收。

