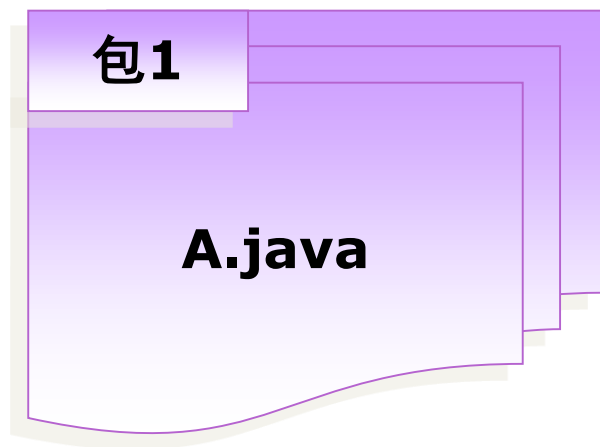


第七章 访问权限



□ 包解决的问题





□ 包

□ 创建

□ 使用：

- import 导入

- 完全限定命名访问

□ 作用



□ Java将类成员的可见度分为四个种类：

访问修饰符	本类	同包	子类	其他
private	√			
默认	√	√		
protected	√	√	√ (不同包子类也可访问)	
public	√	√	√	√



- **static 是一个修饰符**

- **应用：**

- **可以用于修饰属性，方法，块,类**



❑ **static**修饰符 —— ‘静态’

❑ 静态变量：

```
class 类名{  
    public static 数据类型 变量名;  
}
```

问题：

小朋友们交作业，郭靖交一本，
黄蓉交一本，数一数，一共交了2
本作业。



□ 实例变量(非静态成员变量)和类变量(静态变量)的区别：

静态变量	实例变量
静态变量随着类的加载而存在于方法区中	实例变量随着对象的建立而存在于堆内存中
静态变量的生命周期最长,随着类的消失而消失	实例变量的生命周期随着对象的消失而消失
静态变量也称之为类变量,所属于整个类,被整个类所共享	实例变量是对象的特有数据,所属于对象



□ 修饰方法

```
class 类名{  
    public static 返回值 方法名（参数列表）{  
        方法体;  
    }  
}
```


静态方法与实例方法.



□ 区别

静态方法	实例方法
只能直接访问静态成员	可以直接访问静态和非静态成员
不能引用 <code>this</code> 、 <code>super</code>	能引用 <code>this</code> 、 <code>super</code>



定义在类中，用static标识。

```
class Test
{
    staic int num=4;
    //静态代码块
    static{
        System.out.println("我是静态代码块!" + num);
    }
}
```



□ 静态变量初始化方式

□ 静态导入

➤ 格式：

```
import static 包名.类名.静态变量名（方法）；
```



□ 常用方法

`double ceil(double d)`: 获取大于指定参数的最小整数

`double floor(double d)`: 获取小于指定参数的最大整数

`long round(double d)`: 四舍五入

`double pow(double a, double b)` a是底数, b是指数

`double random()`: 产生0到1之间的随机小数



□ 生成某个范围内的随机数字

随机数 = `(int)(Math.random()*(最大值-最小值+1)+ 最小值)` ;

□ 示例：抽奖。

随机产生1-10之间的随机数，输入数字与随机数相同为中奖，奖品为HP笔记本。

