

【济南中心】JAVA 编程阶梯：基础篇之第十章

面向对象之 package 关键字:

包的作用：a.将字节码（.class）进行分类存放 b.包其实就是文件夹

创建包的原则：a.按功能分 b.按模块来分

例子：

学生：增加，删除，修改，查询

老师：增加，删除，修改，查询

按功能分：

com.heima.add;com.heima.delete;com.heima.update;com.heima.find

按模块分：com.heima.teacher; com.heima.student

定义包的注意事项：

A.package 语句必须是程序的第一条可执行代码

B.package 在一个 java 文件中只能有一个

C.如果没有 package，默认表示无包名

带包的类编译和运行：

A.javac 编译带上-d 即可 如：javac -d . HelloWorld.java

B.通过 java 命令执行 如：java 包名.HelloWord

如何引入包：

```
import com.heima.add;
```

四种修饰符：

修饰符权限：

访问权限	类	包	子类	其他包
public	✓	✓	✓	✓
protect	✓	✓	✓	×
default	✓	✓	×	×
private	✓	×	×	×

类及其组成所使用的修饰符：

A：修饰符

权限修饰符：private, default , public , protected

状态修饰符：static, final

抽象修饰符：abstract

B：类

权限修饰符：默认修饰符，public

状态修饰符：final

抽象修饰符：abstract

C：成员变量

权限修饰符：private（常用），默认，protected , public

状态修饰符：static , final

D：构造方法

权限修饰符：private，默认 , protected , public（常用）

E：成员方法

权限修饰符：**private**，默认的，**protected**，**public**（常用）

状态修饰符：**static**，**final**

抽象修饰符：**abstract**

F：除此以外的组合规则：

成员变量：**public static final**

成员方法：**public static**

public final

public abstract

内部类

概述：内部类是指在一个外部类的内部再定义一个类；

内部类访问特点：

a:内部类可以直接访问外部类的成员，包括私有。

b:外部类要访问内部类的成员，必须创建对象。

c:外部类名.内部类名 对象名 = 外部类对象.内部类对象;

例子：

```
class Outer {  
    public int num = 10;  
    class Inner {  
        public int num = 20;  
        public void show() {  
            int num = 30;  
            System.out.println(num);  
        }  
    }  
}
```

静态成员内部类：修饰符 **static**

成员内部类被静态修饰后的访问方式是：

外部类名.内部类名 对象名 = 外部类名.内部类对象;

局部内部类

局部内部类访问局部变量的问题：

局部内部类访问局部变量必须用 final 修饰；

局部内部类在访问他所在方法中的局部变量必须用 final 修饰,为什么?

因为当调用这个方法时,局部变量如果没有用 final 修饰,他的生命周期和方法的生命周期是一样的,当方法弹栈,这个局部变量也会消失,那么如果局部内部类对象还没有马上消失想用这个局部变量,就没有了,如果用 final 修饰会在类加载的时候进入常量池,即使方法弹栈,常量池的常量还在,也可以继续使用

但是 jdk1.8 取消了这个事情,所以我认为这是个 bug

匿名内部类

概述：就是内部类的简化写法。

前提：存在一个类或者接口（这里的类可以是具体的类也可以是抽象类）

格式：new 类名或者接口（）{

重写方法；

}

本质：是一个继承了该类或者实现了该接口的子类匿名对象。

例子：

```
//这里写抽象类，接口都行
abstract class Person {
public abstract void show();
}
class PersonDemo {
public void method(Person p) {
p.show();
}
}
class PersonTest {
public static void main(String[] args) {
//如何调用 PersonDemo 中的 method 方法呢?
PersonDemo pd = new PersonDemo ();
}
}
```



识别二维码 关注黑马程序员视频库
免费获得更多 IT 资源