Java语言基础

【Day09】

访问控制

access control

访问控制修饰符

- 而public修饰的成员变量和方法可以在任何地方调用。 public修饰的内容是对外提供可以被调用的功能，需要相对稳定；
- 用protected修饰的成员变量和方法可以被子类及同一个包中的类使用。
- 默认访问控制即不书写任何访问控制符。默认访问控制的成员变量和方法可以被同一个包中的类调用。
- private修饰的成员变量和方法仅仅只能在本类中调用； private修饰的内容是对内实现的封装，如果“公开”会增加维护的成本。

访问控制修饰符的对比

- 对于类的修饰可以使用public和默认方式。public修饰的类可以被任何一个类使用；默认访问控制的类只可以被同一个包中的类使用。protected和private可以用于修饰内部类。

修饰符	本类	同一个包中的类	子类	其他类
public	可以访问	可以访问	可以访问	可以访问
protected	可以访问	可以访问	可以访问	不能访问
默认	可以访问	可以访问	不能访问	不能访问
private	可以访问	不能访问	不能访问	不能访问

package语句的由来

- 定义类时需要指定类的名称。但如果仅仅将类名作为类的唯一标识，则不可避免的出现命名冲突的问题。这会给组件复用以及团队间的合作造成很大的麻烦！在Java语言中，用包（package）的概念来解决命名冲突的问题。

package的定义

- 在定义一个类时，除了定义类的名称一般还要指定一个包名，格式如下：

package 包名;

```
package test;
class Point {…… …… ……}
```

package语句必须写在Java源文件的最开始，在类定义之前。例如：下面的语句将为Point类指定包名为“test”：一旦使用的package指定了包名，类的全称应该是“包名.类名”。例如，上述的Point类的全称是test.Point。不同的包中可以定义相同的类名，例如test1.Point和test2.Point是两个截然不同的名称。

package的定义

- 包名也可以有层次结构，在一个包中可以包含另外一个包。格式如下：

  package 包名1.包名2…包名n;

- 如果各个公司或开发组织的程序员都随心所欲的命名包名的话，仍然不能从根本上解决命名冲突的问题。因此，在指定包名的时候应该按照一定的规范。

  org.apache.commons.lang.StringUtil

  StringUtils是类名。而org.apache.commons.lang是多层包名，其含义如下：org.apache表示公司或组织的信息（是这个公司（或组织）域名的反写）；common 表示项目的名称信息；lang 表示模块的名称信息。

final关键字

final关键字修饰类

- final关键字修饰的类不可以被继承。
- 使一个类不能被继承的意义在于：可以控制滥用继承对系统造成的危害。

```
class MyString
        extends String   错误
```

final关键字修饰方法

- final关键字修饰的方法不可以被重写。
- 使一个方法不能被重写的意义在于：防止子类在定义新方法时造成的“不经意”重写。

final关键字修饰成员变量

- final关键字修饰成员变量，意为不可改变。该成员变量需在初始化时赋值，对象一旦创建即不可改变。

```
public class Emp {
    private String name;
    private final int no = 100;  final类型的变量必须在声明时初始化
    public Emp(String name) {
        this.name = name;
    }
   public void setNo(int no) {
        this.no = no;                    final类型的变量不可以被改变
   }
}
```

面向对象的三大特征

— 多态

多态的基本概念

- 多态就是指一个事物的多种形态。
- 理论上的多态： 要一个宠物，结果可能是猫/狗/小强/蜗牛。。。
- 宠物(Pet)是多态的，--- 父类可以用子类对象(Cat/Dog/...)去代表。

- 父类变量/引用指向 一个子类的对象，形成多态。
- Pet p = new Cat();
- Pet p = new Dog();

多态的特点

- Person p = new Student(); p 究竟是当父类用，还是子类用呢?
- 其实是当成父类的对象使用。
- 1 只能使用父类中定义的属性和方法。
- 2 不能直接使用子类中扩展的属性和方法。
- 3 如果子类重写了方法，静态方法调父类的，非静态方法调 子类的。
- 原因:
- 编译时，p被认为是Person类型；但在运行时是Student类型。在内存中其实是子类对象。

多态的特点

- 多态时能不能把扩展的属性和方法调出来?
- 能，但需要做类型转换，Person类型的变量 无法调用Student扩展的成员。
- 引用类型的类型转换：
- 必须发生在父子类之间。
- 父类转子类需要做强制类型转换(向下造型)，子类转父类可以自动完成(向上造型)。
- 所谓强制类型转换，Student s = (Student)p; 不是把p的类型转换了，是定义了一个新的变量s，s的类型是Student，p还是Person。
- 对象强制类型转换是一种还原行为，必须内存中是该类型的对象才能成功。

instanceof运算符

- instanceof 就是判断 对象的类型，如果是该类型返回true，不是返回false。
- 语法格式：对象 instanceof 类型
- obj instanceof Object 或 p instanceof Person
- 严格来说，对象进行强制类型转换之前，都应该instanceof一下。
- instanceof 判断支持本类和父类类型。

总结与答疑

技术资料

IT兄弟连

ITXDL.CN

变态严管　让学习成为一种习惯