

## 第二周学习笔记

05月24日

### 一：简述类与对象的关系

类就是对现实生活中具体事物的描述，通常情况下只关注事物的行为和属性，而对象是该类事物存在的个体。简单说，对象就是类的一个实例化。

### 二：封装的概念

封装就是把对象的属性和实现细节进行隐藏，仅仅对外提供公共访问方式。一般设置成员变量为私有，通过 `set` 和 `get` 来操作对象。

### 三：对构造函数的理解

在描述事物的时候，该事物已存在就具备的一些内容，这些内容就可以定义在构造函数中。构造函数的作用是给对象进行初始化。如果类中没有明确定义的构造函数，类中会有一个空参数默认构造函数；若为该类定义了构造函数，那么就不会存在空参数默认构造函数。

一般情况下，构造函数具有函数名与类名相同，不用定义返回类型，没有具体的返回值等特点。

如果一个类中存在多个构造函数，那么这些构造函数以重载的形式存在。

构造函数中存在 `this` 关键字，`this` 可以用来区分成员变量和局部变量，并且 `this` 可以用来调用构造函数，要通过 `this (...)` 的形式。如果该类继承于某个父类，那么该函数首先会执行 `super()`。

### 四：JAVA 能不能多继承，若不能，怎么处理这种问题

不可以，多继承会出现调用模糊的情况。Java 中可以多重继承，并且 java 中接口可以多实现，进而起到多继承的作用。

## 五：单例设计模式的理解

就是保证一个类在内存中的对象唯一性，对于多个程序使用同一个配置信息对象的时候，就需要保证该对象的唯一性。实现方法如下：

1. 不允许其他程序用 new 调用该类对象 ==> 私有化该类构造函数
2. 在该类中创建一个本类实例 ==> 通过 new 在创建一个本类对象
3. 对外提供一个方法，让其他程序可以获取该对象 ==> 定义一个共有的方法给返回创建的对象

## 六：解释下这句代码什么意思

```
public static void main(String[] args)
```

首先 main 函数是程序的入口，其是固定不变的并且被 JVM 所识别和调用。  
`public`：权限必须是最大的 `static`：不需要对象，直接用主函数所属类名调用 `void`：没有具体的返回值 `main`：函数名，不是关键字，只是一个被 JVM 识别的固定的名字 `String[] args`：主函数参数列表，是一个数组类型的参数，而且元素都是字符串型(jvm 往 main 中传了 new String[0])

## 七：UML 类图

Unified modeling language (UML)，统一建模语言。在 UML 类图中，类使用包含类名、属性、方法且带有分割线的矩形来表示一个类。

常见符号：+ (public)、- (private)、# (protected)

属性表示方式：可见性 名称：类型

方法表示方式：可见性 名称（参数列表）[:返回类型]

类与类之间的关系的表现形式：

依赖关系（使用）、泛化关系（继承关系）、接口实现关系、关联关系、聚合关系、组合关系

八：语法知识：

1、定义 `int[] x, y[]`；判断下面赋值操作是否正确。（毕4-14）

- a. `x[0] = y`；正确
- b. `y[0] = x`；错误
- c. `y[0][0] = x`；错误
- d. `x[0][0] = y`；错误
- e. `y[0][0] = x[0]`；错误
- f. `x = y`；错误

2、类和对象的关系？（毕5-3）

类就是对现实生活中具体事物的描述，通常情况下只关注事物的行为和属性，而对象是该类事物存在的个体。简单说，对象就是类的一个实例化。

3、局部变量和对象成员变量的区别。（毕5-4）

- 1. 成员变量定义在类中，整个类中都可以访问；局部变量定义在函数，语句和局部代码块中，只在所属的区域内有效
- 2. 局部变量存在于栈内存中，成员变量存在于堆内存中
- 3. 成员变量随着对象的创建而存在，随着对象的消失而消失；局部变量随着所属区域的执行而存在，随着所属区域的结束而释放

4. 成员变量有默认初始化，而局部变量没有默认初始化

#### 4、什么是封装？（毕5-6）

封装就是把对象的属性和实现细节进行隐藏，仅仅对外提供公共访问方式。一般设置成员变量为私有，通过 set 和 get 来操作对象。

#### 5、构造函数合一般函数的区别？（毕5-7）

构造函数：对象创建时就会调用与之对应的构造函数，对对象进行初始化；并且只调用一次，可以调用一般函数

一般函数：对象创建后，需要该函数功能的时候才会调用；可以被调用多次

#### 6、构造代码块和构造函数的区别？（毕5-9）

构造函数给对应的对象进行针对化的初始化，而构造代码块可以给所有的对象进行初始化

#### 7、一般函数怎么调用构造函数？（毕5-12）

This

#### 8、实例变量和类变量的区别？（毕6-1）

成员变量随着对象的创建而存在，随着对象的被回收而释放；静态变量随着类的加载而存在，随着类的消失而消失 ==> 两个变量的生命周期不同

调用方式不同 成员变量只能被对象调用，静态变量可以被类名（推荐）对象调用

数据的存储位置不同 成员变量存储在堆内存的对象中，所以也叫作对象的特有数据；静态变量数据存储在方法区（共享数据区）的静态区中，所以也叫对象的共享数据

## 9、为什么 main 主函数要用 static 修饰。（毕6-2）

因为 main 函数是程序的入口，在没进入主函数之前无法进行类的实例化，静态函数在类定义的时候就存在了，不需要实例化就可以使用，所以要用 static 修饰

## 10、什么时候使用 static。（毕6-3）

静态变量：当分析对象中所具备的成员变量的值都是相同的，并且不需要修改只需要使用即可，就可以被静态修饰

静态函数：该函数是否需要访问到非静态成员变量，如果需要，该功能就是非静态的，如果不需要，就可以将该功能设定为静态函数

## 11、给出下列程序的输出结果顺序？（毕6-6）

```
public class StaticTest2{
    public static void main(String[] args){
        new StaticDome();
        System.out.println("aa");
    }
    static{
        System.out.println("bb");
    }
    {
        System.out.println("cc");
    }
}

class StaticDome{
    public StaticDome(){
        System.out.println("dd");
    }
    static{
        System.out.println("ee");
    }
    {
        System.out.println("ff");
    }
}
```

Bb ee ff dd aa

## 12、为什么 java 不支持多继承？（毕7-2）

继承的模糊性，如果两个子类分别重写父类的某个函数，当继承于这两个子类的类的实例化去调用这个函数的时候，就会出现歧义。C++中允许多继承

## 13、请问下列语句是否可以正常编译执行？如果不能为什么？

请问最简单的修改是怎么样子的？（毕7-6）

```
public class ExtendsTest2{
    public static void main(String[] args){
        Zi z = new Zi();
    }
}
class Fu {
    Fu(int x){
        System.out.println("父类构造函数 x = "+x);
    }
}
class Zi extends Fu{
    Zi(){
        this(3);
    }
    Zi(int x){
        System.out.println("子类构造函数 "+x);
    }
}
```

不可以编译成功，在倒数第三行加上 `super(x);` 或者直接给父类一个空参数构造函数。因为子类的构造函数首先要执行 `super()`，而父类没有空参数构造函数。

## 14、抽象类和一般类的区别？（毕7-9）

相同点：抽象类和一般类都是用来描述事物的，都在内部定义了成员

不同点：

一般类有足够的信息描述事物，抽象类的信息一般不具体

一般类中不能定义抽象方法，只能定义非抽象方法；抽象类中可以定义抽象方法，同时也可以定义非抽象方法

一般类可以被实例化，抽象类不行

## 15、抽象类和接口的区别？（毕7-12，毕7-13）

相同点：都是抽象的，不断向上抽取而来的

不同点：

抽象类需要被继承（不可以多继承），接口需要被实现（可以多实现）

抽象类中可以定义抽象方法和非抽象方法，子类继承后可以直接使用非抽象方法；而接口中全部是抽象方法，必须由子类实现全部方法

抽象类的继承是“is a”关系，定义了体系内的共性内容；接口是“like a”关系，定义了体系的额外功能

## 16、内部类为什么直接可以访问外部类的成员（毕9-1）

因为内部类定义在外部类的里面，就相当于外部类的一个成员。可以访问外部类的其他成员。本质是因为，内部类中存在了外部类的引用，Outer.This（）

## 17、局部内部类和成员内部类有什么区别？（毕9-1，9-2）

局部内部类（在外部类的局部代码中），并且局部内部类只能访问局部中被 final 修饰的局部变量。

## 18、匿名内部类的优缺点？（毕9-4）

匿名内部类只能被使用一次而且必须要继承父类或者实现接口。

当函数的参数是接口类型时，并且接口的方法不超过 3 个，可以使用匿名内部类作为实际参数进行传递。