

第二周学习笔记

07月23日

一：简述类与对象的关系

1. 类

事物的描述。是具备某些共同特征的实体的集合，它是一种抽象的数据类型，它是对所具有相同特征实体的抽象。在面向对象的程序设计语言中，类是对一类“事物”的属性与行为的抽象。

2. 对象

正文该类事物的实例。在 Java 中通过 new 进行创建。是一个真实世界中的实体，对象与实体是一一对应关系的，意思就是现实世界的每一个实体都是一个对象，所以对象是一个具体的概念。对象是一种个性的表示，表示一个独立的个体，每个对象拥有自己独立的属性，依靠属性来区分不同对象。；

3. 区别

- a). 类是一个抽象的概念，它不存在于现实中的时间/空间里，类只是为所有的对象定义了抽象的属性与行为。就好像“Person（人）”这个类，它虽然可以包含很多个体，但它本身不存在于现实世界上。
- b). 对象是类的一个具体。它是一个实实在在存在的东西。类是一个静态的概念，类本身不携带任何数据。当没有为类创建任何对象时，类本身不存在于内存空间中。
- c). 对象是一个动态的概念。每一个对象都存在着有别于其它对象的属于自己的独特的属性和行为。对象的属性可以随着它自己的行为而发生改变。；

二：封装的概念

1. 基本概念

正文封装是面向对象的方法遵循的一个重要原则，有时也叫做数据隐藏。

从形式上看，封装将数据和行为组合在一个包中，并对对象的使用者隐藏了数据的实现形式。

对象中的数据称为实例域,操纵数据的过程叫做方法,对于每个特定的对象,都有一组特定的实例域值。这些值的集合就是这个对象的当前状态. 无论何时,只要向对象发送一个消息,它的状态就有可能发生改变。

封装的关键在于不能让类中的方法直接地访问其他类的实例域。程序仅通过对象的方法,与对象数据进行交互。这意味着一个类可以全面地改变存储数据的方式,只要仍旧使用同样的方法操作数据,其他对象就不知道所发生的变化

三：对构造函数的理解

1. 概念

构造函数是一种特殊的函数。其主要功能是用来在创建对象时初始化对象, 即为对象成员变量赋初始值, 总与 new 运算符一起使用在创建对象的语句中。构造函数与类名相同, 可重载多个不同的构造函数。在 JAVA 语言中, 构造函数普遍被称之为构造方法。;

2. 特点

a). 构造方法的方法名必须与类名相同。

构造方法没有返回值类型, 可以有 return, 但是不能有返回值, 在这里只是表示结束, 并不是返回的表示, 一般也不使用 return。

b). 构造方法的主要作用是完成对象的初始化工作, 它能够把定义对象时的参数传给对象的域。

c). 一个类可以定义多个构造方法, 如果在定义类时没有定义构造方法, 则编译系统会自动插入一个无参数的默认构造方法, 这个构造方法不执行任何代码。

d). 构造方法可以重载, 以参数的个数, 类型, 顺序。

3. 理解

构造一个新对象的方法。

四：JAVA 能不能多继承, 若不能, 怎么处理这种问题

1. 不能

其问题在于子类的功能由谁实现, 使用时能否准确定位是谁做。子类继承父类, 没有重写方法, 调用的是父类的功能; 假如可

以继承多个父类，那么当父类 A, B 中有同样的方法或参数时，子类 C 调用父类方法或属性时很难判断要调用谁；

2. 二解决方法

通过接口，接口不同，接口没有方法体，也就是没有功能，功能是实现类来做的，因此不用考虑让谁做的问题；定义的属性也是常量，要通过接口名来调用

五：单例设计模式的理解

1. 概念

一个类中只提供一个实例对象让外面访问。；

2. 实现方法

- a). 在类初始化时，已经自行实例化；
- b). 在第一次调用的时候实例化
- c). 将类名注册，下次从里面直接获取

3. 理解

由类的访问模式定义的编程设计模式；

六：解释下这句代码什么意思

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

1. 答：

- A)、String[] args：字符串数组，接收程序执行时来传进来的参数；
- B)、public：程序访问权限，表示该程序在任何场合都可以被引用；
- C)、static：表示方法是静态的，不依赖类的对象，是属于类的；
- D)、void：表示方法无返回值；

七：UML 类图

1. 作用：

识别出所有必须的类，并分析这些类之间的关系。

2. 知识：

- a) 类图中包含类之间的关系分别有有关联、聚合、组合、泛化、依赖，

- b) 聚合表示整体与部分的关系；
- c) 关联是模型元素之间的一种语义联系，是类之间的一种很弱的联系
- d) 组合表示类之间整体和部分的关系
- e) 泛化定义了一般元素和特殊元素之间的分类关系，类之间的这种泛化关系也就是继承关系。

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`;
错
- b. `y[0] = x`;
对
- c. `y[0][0] = x`;
错
- d. `x[0][0] = y`;
错
- e. `y[0][0] = x[0]`;
对
- f. `x = y`;
错

2、类和对象的关系？(毕5-3)

类是没有具体分配储存空间的，是一种概念或规范，对象为类的具体实现，类为对象定义属性和行为。

3、局部变量和对象成员变量的区别。(毕5-4)

局部变量：

在方法内部定义，在方法的内部有效，没有默认值，位于栈内存，随着方法进栈而诞生，随着方法出栈而消失

成员变量：

在方法的外部类的内部定义，整个类中都有效，如果没有赋值，则有默认值，规则和数组一样，位于堆内存，随着对象创建而诞生，随着对象被垃圾回收而消

4、什么是封装？（毕5-6）

封装：数据隐藏，对使用者隐藏对象数据的实现形式，仅能通过对应的方法进行操作

5、构造函数合一般函数的区别？（毕5-7）

构造函数用于初始化对象，为对象的成员进行赋值。

6、构造代码块和构造函数的区别？（毕5-9）

相同点：每创建一个对象，构造代码块和构造函数都执行一次，
不同点：执行的顺序不同，先执行构造代码块，再执行构造方法。

7、一般函数怎么调用构造函数？（毕5-12）

一般函数不能调用构造函数。

8、实例变量和类变量的区别？（毕6-1）

实例变量：

实例变量也称为对象变量，实例变量为所属对象所私有；

类变量：

也叫做静态变量，为所有对象所共有；

区别：

如果一个对象将类变量的值进行改变，那么其他对象将不能改变他的值；而实例变量则是为对象所私有，一个对象改变不影响其他对象；

9、为什么 main 主函数要用 static 修饰。（毕6-2）

静态，一上来就装入内存，而 Main 主函数是一个程序的入口，如果写成非静态的，那么就必须实例化一个对象再来调用它；静态方法是属于类的，直接用类名就可以用 static 方法不需要创建对象 可以直接访问，这样 main 主函数就不需要做特殊处理，

10、什么时候使用 static。（毕6-3）

用 static 修饰的方法可以用类名直接调用，所以当方法或者变量需要初始化加载，或者是经常被调用的时候可以加上 static。

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");
    }
}
```

答：

bb
aa

12、为什么 java 不支持多继承？（毕7-2）

多继承容易在子类调用父类方法上引起混乱，因此不支持

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){
        Super(x);
        System.out.println("子类构造函数 "+x);
    }
}
```

修改：

不能，子类继承父类不会继承父类的构造函数，每个类都有独立的构造函数，若子类无构造函数则会默认调用父类的无参构造、或通过 `super` 调用父类的构造。这里子类无构造，父类无无参构造。

14、抽象类和一般类的区别？（毕7-9）

- a). 抽象类不能像普通化一样实例化出对象。
- b). 抽象类的访问权限限于 `Public` 和 `Protected`，因为抽象类的方法是继承之后让子类去实现的，如果为 `Private`，则无法被子类继承，子类也无法实现该方法。
- d). 如果一个类继承于抽象类，则该子类必须实现父类的抽象方法。如果子类没有实现父类的抽象方法，则必须将子类也定义为 `abstract` 类。

15、抽象类和接口的区别？（毕7-12，毕7-13）

抽象类可以包含具体方法，接口只能是抽象方法

16、内部类为什么直接可以访问外部类的成员（毕9-1）

- a) 内部类对象的创建依赖于外部类对象；
- b) 内部类对象持有指向外部类对象的引用，该引用的实现由编译器自动完成。

17、局部内部类和成员内部类有什么区别？（毕9-1, 9-2）

- a) 成员内部类中不能定义静态变量，但可以访问外部类的所有成员。
- b) 局部内部类中不可定义静态变量，可以访问外部类的局部变量（即方法内的变量），但是变量必须是 `final` 的。

18、匿名内部类的优缺点？（毕9-4）

优点：

- 1、内部类方法可以访问该类定义所在的作用域中的数据，包括私有的数据；
- 2、内部类可以对同一个包中的其他类隐藏起来；
- 3、使用匿名内部类定义回调函数，节省代码；

缺点：

- 1、使用有很多限制，如类中方法多的类匿名内部类表达式会很长，没有阅读性；或者是需要多次访问内部类的情况，就不适合；
- 2、容易造成内存泄漏