# 实验六 类与对象

## 1. 实验目的

(1) 理解类的封装概念。

(2) 掌握Java定义类的格式。

(3) 掌握构造方法的设计方法。

(4) 掌握成员变量和成员方法的设计方法。

(5) 掌握this关键字的使用方法。

## 2. 实验内容

1. 定义长方体类Cuboid，要求如下：（1）私有成员变量包括长length、宽width和高height；（2）构造方法包括一个公共的空构造方法，一个能够初始化所有成员变量的构造方法；（3）包括所有的setter和getter方法，其中setter方法要求对形参进行验证，例如长、宽和高都大于0；（4）功能方法包括计算表面积getArea()、计算体积getCubage()、判断是否正方体isCube()、输出该长方体的长宽高printCuboid()、绘制该长方体（给出提示信息即可）drawing()。

测试一要求如下：(1)利用空构造方法创建一个新的长方体对象cuboidA，然后利用setter方法设置长宽高分别为1.5、2.6和18.6，(2)调用printCuboid()方法输出cuboidA的基本信息，调用isCube()方法判断是否正方体；（3）调用方法drawing()模拟绘制该长方体。

测试二要求如下：(1)利用构造方法创建一个具有长宽高分别为2.5、6.8和8.0的长方体对象cuboidB，然后利用setLength()方法修改length为4.6；(2)调用getArea()方法计算表面积；(3)调用getCubage计算该长方体的体积。

**package** test6;

**class** Cuboid {

**private** Double length;

**private** Double width;

**private** Double height;

**public** Cuboid() {

}

**public** Cuboid(Double length, Double width, Double height) {

**this**.length = length;

**this**.width = width;

**this**.height = height;

}

**public** Double getLength() {

**return** length;

}

**public** **void** setLength(Double length) {

**if**(length > 0)

**this**.length = length;

}

**public** Double getWidth() {

**return** width;

}

**public** **void** setWidth(Double width) {

**if**(width > 0)

**this**.width = width;

}

**public** Double getHeight() {

**return** height;

}

**public** **void** setHeight(Double height) {

**if**(height > 0)

**this**.height = height;

}

**public** Double getArea() {

Double Area = (**this**.width \* **this**.length + **this**.width \* **this**.height + **this**.height \* **this**.length) \* 2;

**return** Area;

}

**public** Double getCubage() {

Double Cubage = **this**.width \* **this**.length \* **this**.height;

**return** Cubage;

}

**public** **void** isCube() {

**if**(**this**.width == **this**.height && **this**.height == **this**.length)

System.*out*.println(" 是正方体 ");

**else**

System.*out*.println(" 不是正方体 ");

}

**public** **void** printCuboid() {

System.*out*.println("长 :" + **this**.length + " 宽 :" + **this**.width + " 高 " + **this**.height);

}

**public** **void** drawing() {

System.*out*.println("已绘制该长方体");

}

}

**public** **class** demo {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

Cuboid cuboidA = **new** Cuboid();

// 测试一

System.*out*.println("测试一");

cuboidA.setLength(1.5);

cuboidA.setWidth(2.6);

cuboidA.setHeight(18.6);

cuboidA.printCuboid();

cuboidA.isCube();

cuboidA.drawing();

// 测试二

System.*out*.println("测试二");

Cuboid cuboidB = **new** Cuboid(2.5, 6.8, 8.0);

cuboidB.setLength(4.6);

System.*out*.println("表面积 " + cuboidB.getArea());

System.*out*.println("体积 " + cuboidB.getCubage());

}

}

2. 定义房间类Room，要求如下：(1)私有成员变量包括房间编号id，面积area，户型type，单价price（元/平米），销售状态state等；(2)定义一个空构造方法；定义第二个构造方法能够根据房间编号，面积，单价创建Room对象；定义第三个构造方法能够根据所有成员变量创建Room对象；(3)定义针对成员变量state和price的setter和getter方法，定义getArea()方法返回房间面积；(3)功能方法包括：showRoom()方法输出Room对象所有成员变量；getMoney()获得房间总价。

楼盘river房间Room对象信息如表6-1中所示。定义测试类，对表6-1完成如下操作：(1)统计3个房间的总面积；(2)统计3个房间的总价；(3)把3-2-502房间的销售状态修改为true；(4)显示3个房间的详细信息。

表6-1 楼盘river的房间信息

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号id | 面积area | 户型type | 单价price | 销售状态state |
| 3-2-502 | 113.5 | 三室一厅 | 15000 | false |
| 4-5-1108 | 156.9 | 四室二厅 | 17500 | true |
| 4-3-1501 | 45.9 | 二室一厅 | 16200 | true |

**class** Room{

// 定义成员变量

**private** String id;

**private** Double area;

**private** String type;

**private** Double price;

**private** **boolean** state;

// 定义空构造方法

**public** Room() {

}

// 定义构造方法能够根据房间编号,面积,单价创建Room对象

**public** Room(String id, Double area, Double price) {

**this**.id = id;

**this**.area = area;

**this**.price = price;

}

// 定义构造方法能够根据所有成员变量创建Room对象

**public** Room(String id, Double area, String type, Double price, **boolean** state) {

**this**.id = id;

**this**.area = area;

**this**.type = type;

**this**.price = price;

**this**.state = state;

}

// 定义针对成员变量 state 和 price 的setter 和getter 方法, 定义getArea()方法返回房间面积

**public** **boolean** getState() {

**return** state;

}

**public** **void** setState(**boolean** state) {

**this**.state = state;

}

**public** Double getPrice() {

**return** price;

}

**public** **void** setPrice(Double price) {

**this**.price = price;

}

**public** Double getArea() {

**return** area;

}

// 功能方法

// showRoom() 方法输出Room 对象所有成员变量 getMoney() 方法获得房间总价

**public** **void** showRoom() {

System.***out***.println(**this**.id + "\t" + **this**.area + "\t" + **this**.type + "\t" + **this**.price + "\t" + **this**.state);

}

**public** Double getMoney() {

**return** **this**.area \* **this**.price;

}

};

**public** **class** debug4 {

**public** **static** **void** main(String[] args) {

// 定义测试类

Room[] river = **new** Room[3];

river[0] = **new** Room("3-2-502 ", 113.5, "三室一厅", 15000D, **false**);

river[1] = **new** Room("4-5-1108", 156.9, "四室一厅", 17500D, **true**);

river[2] = **new** Room("4-3-1501", 45.9, "二室一厅", 16200D, **true**);

// 统计3个房间的总面积

Double totalArea = 0D;

**for**(Room room:river)

totalArea += room.getArea();

System.***out***.printf("3个房间的总面积为 %.2f\n", totalArea);

// 统计3个房间的总价

Double totalPrice = 0D;

**for**(Room room:river)

totalPrice += room.getMoney();

System.***out***.printf("3个房间的总价为 %.2f\n", totalPrice);

// 修改 3-2-502 房间销售状态为 true

river[0].setState(**true**);

// 显示3个房间的详细信息

System.***out***.println("\t楼盘river 的房间信息");

System.***out***.println("编号\t面积\t户型\t单价\t销售状态");

**for**(Room room:river)

room.showRoom();

}

}

3. 定义汽车类Car,要求如下：(1)私有成员变量包括汽车品牌brand，排气量emission和价格price；(2)定义一个构造方法初始化所有成员变量；(3)定义getBrand()方法返回品牌信息;(4)功能方法包括：starting()方法启动发动机。

定义Hand类，要求如下：(1)私有成员变量包括手指长度length，手掌面积area和握力grip；(2)定义构造方法初始化所有成员变量;(3)省略setter和getter方法；(4)功能方法包括：catching(String thing)方法抓东西，getInfo()方法取得Hand的所有信息。

定义Person类，要求如下：(1)私有成员变量包括身份证号码id,姓名name，年龄age和手hand;(2)定义一个空构造方法，另一个构造方法初始化所有成员变量；(3)定义getName()方法返回人的姓名;(4)功能方法包括：showPerson()方法显示某人的所有信息；(2)saying(Person per,String msg)方法表示一个人对另一个人说了一些信息msg;(3)getSomething(String something)表示用手拿东西，需要调用Hand对象的拿方法catching();(4)start(Car car)方法表示某人启动了一台汽车，需要调用Car的starting方法启动发动机。

定义测试类Demo，完成如下任务：(1)创建一台汽车，品牌是红旗L5，排气量是3.0，价格是25万；(2)创建一只手，手指长度8厘米,手掌面积42平方厘米，握力是45kg；(3)创建Person对象孙悟空和猪八戒，孙悟空向猪八戒说“八戒，你这个呆子！”，孙悟空拿了一根金箍棒并启动了一辆红旗L5汽车。

Demo6\_3.java 内容

// 定义测试类 Demo

**class** Demo {

// 创建一台汽车

Car car = **new** Car("红旗L5", 3.0, 250000D);

// 创建一只手

Hand hand = **new** Hand(8D, 42D, 45D);

// 创建 Person 对象孙悟空和猪八戒

Person sunwukong = **new** Person("1", "孙悟空", 500, hand);

Person zhubajie = **new** Person("2", "猪八戒", 480, hand);

}

**public** **class** Demo6\_2 {

**public** **static** **void** main(String[] args) {

Demo demo = **new** Demo();

System.***out***.print("孙悟空对八戒说:");

demo.sunwukong.saying(demo.zhubajie, "八戒,你个呆子");

demo.sunwukong.getSomething("拿起了一根金箍棒");

demo.sunwukong.start(demo.car);

System.***out***.println(demo.car.getBrand());

}

}

Person.java 内容

//定义 Person 类

**public** **class** Person {

// 定义成员变量

**private** String id;

**private** String name;

**private** **int** age;

**private** Hand hand;

// 定义空构造方法

Person() {

}

Person(String id, String name, **int** age, Hand hand) {

**this**.id = id;

**this**.name = name;

**this**.age = age;

**this**.hand = hand;

}

// 定义 getName() 方法返回人的姓名

**public** String getName() {

**return** **this**.name;

}

// 功能方法

// showPerson() 方法显示某个人的信息

// saying(Person per, String msg) 方法表示一个人对另一个人说了一些信息 msg

// getSomething(String something)表示用手拿东西,需要调用 Hand 对象的拿方法 catching()

// start(Car car) 方法表示某人启动了一台汽车,需要调用 Car 的 starting 方法启动发动机

**public** **void** showPerson() {

}

**public** **void** saying(Person per, String msg) {

System.***out***.println(msg);

}

**public** **void** getSomething(String something) {

**new** Hand().catching(something);

}

**public** **void** start(Car car) {

car.starting();

}

}

Hand.java 内容

// 定义 Hand 类

**public** **class** Hand {

// 定义成员变量

**private** Double length;

**private** Double area;

**private** Double grip;

**public** Hand() {

}

// 定义初始化所有成员变量

**public** Hand(Double length, Double area, Double grip) {

**this**.length = length;

**this**.area = area;

**this**.grip = grip;

}

// 省略 setter 和 getter 方法

// 功能方法

// catching(String thing) 方法抓东西

// getInfo() 方法取得Hand的所有信息

**public** **void** catching(String thing) {

System.***out***.println(thing);

}

**public** **void** getInfo() {

System.***out***.println("手指长度 " + **this**.length + "手掌面积 " +

**this**.area + "握力 " + **this**.grip);

}

}

Car.java 内容

//定义汽车类 Car

**public** **class** Car {

// 定义成员变量

**private** String brand;

**private** Double emission;

**private** Double price;

// 定义构造方法初始化所有成员变量

**public** Car(String brand, Double emission, Double price) {

**this**.brand = brand;

**this**.emission = emission;

**this**.price = price;

}

**public** String getBrand() {

**return** **this**.brand;

}

**public** **void** starting() {

System.***out***.println("启动发动机");

}

}

## 3. 实验步骤

【定义类的过程分为四步】：

第一步： 定义成员变量；

第二步： 定义构造方法，构造方法可以重载；

第三步： 定义setter和getter方法；

第四步： 定义功能方法。

3.2定义房间类Room

【解题思路】：

(1) 建立Room.java文件包含公共类Room。

(2) 建立测试文件Demo6\_2.java，该文件的类中有主方法。

3.3定义Person、Car和Hand类

【解题思路】：

(1) 建立Car.java文件包含公共类Car

(2) 建立Hand.java文件包含公共类Hand；

(3) 建立Person.java文件包含公共类Person，该类中的成员变量Hand类型与Person是组成关系，该类中的方法start(Car car)的参数是Car类型，说明Car与Person是依赖关系。【需要画图示意！！！！！！！！】

(4) 建立测试文件Demo6\_3.java，该文件的类中有主方法。

## 4. 实验总结

本章实验中，通过定义类以及类的使用，加深对类和对象概念的理解，通过成员访问控制符加深对类的封装性理解。定义类分为定义成员变量、定义构造方法、定义setter和getter方法，最后定义功能性方法等四步，构造方法可以重载，体现了对象设计中的静态多态性。类与类之间有组成、聚合和依赖等关系，其中Hand对象是Person的成员变量，体现了类之间的组成关系；Person中start(Car car)方法的形参是Car对象，体现了Car与Person之间是依赖关系。