

第4章 类和对象

Java 是一种纯面向对象的程序设计语言，目前比较流行。在进行编程之前，必须理解类、对象、消息、封装、继承和多态等基本概念。通过本章的习题解答和实验练习，读者应该能够进一步理解这些概念，并且在此基础上掌握类、对象的定义和使用方法。

4.1 典型例题解答

一、选择题

1. 以下关于构造方法的描述错误的是（ ）。

- A) 构造方法的返回类型只能是 void 型
- B) 构造方法是类的一种特殊方法，它的方法名必须与类名相同
- C) 构造方法的主要作用是完成对类的对象的初始化工作
- D) 一般在创建新对象时，系统会自动调用构造方法

【答案】A

2. 假设类 A 定义如下，设 a 是 A 类的一个实例，下列语句调用错误的是（ ）。

```
class A{
    int i;
    static String s;
    void method1() {}
    static void method2() {}
}
```

- A) System.out.println(a.i);
- B) a.method1();
- C) A.method1();
- D) A.method2();

【答案】D

3. 以下关于对象的说法不正确的是（ ）。

- A) 对象是一个具有封装性和信息隐藏的独立模块
- B) 对象能更好地模拟计算机工作方式，体现计算机运行规律，提高程序执行效率
- C) 组成客观世界的不同实体都可以看成是对象
- D) 对象可以分解和组合，还可以进行分类和抽象

【答案】B

4. 下列哪一项不是成员变量的非访问控制符（ ）。

- A) static
- B) final
- C) native
- D) public

【答案】D

5. 关于被保护访问控制符 protected 修饰的成员变量，以下说法正确的是（ ）。

- A) 可以被三种类所引用：该类自身、与它在同一个包中的其他类、在其他包中的该

类的子类

- B) 可以被两种类访问和引用：该类本身、该类的所有子类
- C) 只能被该类自身所访问和修改
- D) 只能被同一个包中的类访问

【答案】A

6. 下列关于修饰符混用的说法，错误的是（ ）。

- A) `abstract` 不能与 `final` 并列修饰同一个类
- B) `abstract` 中不可以有 `private` 的成员
- C) `abstract` 方法必须在 `abstract` 类中
- D) `static` 方法中能处理非 `static` 的属性

【答案】D

7. 在 Java 中，一个类可同时定义许多同名的方法，这些方法的形式参数个数、类型或顺序各不相同，传回的值也可以不相同。这种面向对象程序的特性称为（ ）。

- A) 隐藏
- B) 覆盖
- C) 重载
- D) Java 不支持此特性

【答案】C

8. 在 Java 源代码中有如下声明对象的方法：

- ① A a0=new A();
- ② A a1;
a1=new A();
- ③ a2=new A();

问以下哪个说法是正确的？（ ）

- A) 只有第 1 行能通过编译
- B) 第 1、2 行能通过编译，但第 3 行编译出错
- C) 第 1、2、3 行能通过编译，但第 2、3 行运行时出错
- D) 第 1 行、第 2 行和第 3 行的声明都是正确的

【答案】B

9. `main` 方法是 Java Application 程序执行的入口点，关于 `main` 方法的方法头，以下哪项是合法的（ ）。

- A) `public static void main()`
- B) `public static void main(String[] args)`
- C) `public static int main(String[] arg)`
- D) `public void main(String arg[])`

【答案】B

10. 有如下类：

```
public class Simple{
    public static int i;
}
```

若 `s1`、`s2` 均为 `Simple` 的对象，则（ ）。

- A) `s1`、`s2` 的 `i` 值不一定相同
- B) `s1`、`s2` 的 `i` 值一定不同
- C) `s1`、`s2` 的值没有任何关系，但当赋值后均不可改变

D) s1、s2 的 i 值永远相同

【答案】D

11. 下列描述正确的是（ ）。

A) 静态变量的值不能变化

B) 静态变量就是常量，其值一旦被初始化就不能再改变

C) 静态变量只能做一次赋值运算

D) 一个类的所有对象共享静态变量

【答案】D

二、填空题

1. 程序中定义类使用的关键字是 `class`，每个类的定义由类头定义、类体定义两部分组成，其中类体部分包括_____、_____。

【答案】属性、方法

2. 把对象实例化可以生成多个对象，使用_____运算符为对象分配内存空间。

【答案】`new`

3. 定义类的构造方法不能有返回值类型，其名称与_____名相同。

【答案】类

4. 在 Java 中有一种叫做_____的特殊方法，在程序中用它来对类成员进行初始化。

【答案】构造方法

5. Java 语言以_____为程序的基本单位，它是具有某些共同特性实体的集合，是一种抽象的概念。

【答案】类

6. `new` 是_____对象的操作符。

【答案】创建

7. 在 Java 程序里，同一类中重载的多个方法具有相同的方法名和_____的参数列表。重载的方法可以有不同的返回值类型。

【答案】不同

8. Java 是面向对象语言。对象是客观事物的_____，对象与之是一一对应的，它是很具体的概念。

【答案】实例

9. Java 语言通过接口支持_____继承，使类继承具有更灵活的扩展性。

【答案】多

10. 创建一个类就是创建一个新的数据类型，而类在 java 中属于_____数据类型。

【答案】引用

11. `abstract` 方法_____（不能或能）与 `final` 并列修饰同一个类。

【答案】不能

12. 如果一个类中定义了几个名为 `method` 的方法，这些方法的参数都是整数，则这些方法的_____必须是不同的，这种现象称为方法的重载。

【答案】参数个数

三、简答题

1. 什么是类？写出 Java 中类定义的格式？

【解答】在面向对象程序设计中，类实质上是一种对象类型，它是对具有相同属性和相同行为对象的一种抽象。类是 Java 程序中最小的组成单位。

Java 中类定义的格式为：

```
[类修饰符] class 类名 [extends 基类][ implements 接口]
{
    //成员变量声明 (Member variable declaration)
    //成员方法声明 (Member method declaration)
}
```

其中，class、extends 和 implements 都是 Java 的关键字。类修饰符、extends 和 implements 是关于修饰符、继承和接口的内容。类名是用户定义的标识符。

2. 请举例说明什么是数据成员？什么是成员方法？

【解答】例如：定义点 Point 类。

```
class Point {
    long xCoordinate,yCoordinate;
    setx(long x){ xCoordinate=x; }
    sety(long y){ yCoordinate=y; }
    setp(long x,long y){ setx(x); sety(y);}
}
```

对于上述类，其中 xCoordinate 和 yCoordinate 是数据成员，setx()、sety()和 setp()是成员方法；数据成员一般描述对象的属性；成员方法描述对象所具有的功能或操作，反映对象的行为，是具有某种独立功能的程序模块。

四、编程题

1. 编程：计算 n 的阶乘，用递归方法实现。

【解答】解题思路：

- (1) 定义方法实现 n 的阶乘。
- (2) $n! = n * (n-1)!$ 。
- (3) 输出计算结果。

参考程序：

```
public class C04E1 {
    public static void main(String args[]){
        int i=8;
        System.out.println("8 的阶乘为: "+fac(8));
    }
    public static int fac(int n){
        if(n==1)return 1;
        else return n*fac(n-1);
    }
}
```

2. 输入一个十进制数，输出相应的十六进制数。

【解答】 解题思路：

- (1) 定义方法实现十进制数转换成十六进制数。
- (2) 使用递归方法完成相应功能。
- (3) 十进制数转换成十六进制数使用相除取余法。
- (4) 在输出时，十六进制数 10~15 要转换为 a~f 或 A~F。

参考程序：

```
import java.io.*;
public class C03E2{
    public static void main(String args[]){
        byte buf[]=new byte[50];
        String str;
        int a=0;
        try { System.out.print("输入圆的半径: ");
            System.in.read(buf);
            str=new String(buf,0);
            a=Integer.parseInt(str.trim());
        }catch (Exception e) {}
        change(a,16);
    }
    public static void change(int n,int base) {
        if(n>16)
            change(n/16,base);
        if(n%16<10)
            System.out.print(n%16);
        else{
            switch(n%16){
                case 10:System.out.print('A');break;
                case 11:System.out.print('B');break;
                case 12:System.out.print('C');break;
                case 13:System.out.print('D');break;
                case 14:System.out.print('E');break;
                case 15:System.out.print('F');break;}
        }
    }
}
```

3. 编写程序，实现 Hanoi 塔游戏。Hanoi 塔游戏的具体规则：有 n 个直径不同的盘子套在第一个针上，要求将这些盘子借助第二个针，移动到第三个针上。在移动过程中，每次只能移动一个盘子，每个针上任意时刻保证小盘子在大盘子上面，并且只能在这三个针上移动，要求编程打印出移动的步骤。

【解答】 解题思路：

- (1) 定义方法完成 Hanoi 塔游戏。
- (2) 使用递归的形式来完成，盘子移动分三个步骤：第一步，借助第三个针，把第一个

针上的 $n-1$ 个盘子移动到第二个针上；第二步，把第一个针上的最后 1 个盘子移动到第三个针上；第三步，把第二个针上的 $n-1$ 个盘子，借助于第一个针移动到第三个针上。

(3) 输出移动步骤。

参考程序：

```
public class C04E3 {
    public static void main(String args[]){
        byte buf[]=new byte[50];
        String str;    int n=0;
        try {System.out.print("输入盘子的个数: ");
            System.in.read(buf);
            str=new String(buf,0);
            n=Integer.parseInt(str.trim());
        } catch (Exception e) {}
        System.out.println("移动步骤为: ");
        hanoi(n,'a','b','c');}
    public static void hanoi(int n,char a,char b,char c) {
        if(n==1)
            System.out.println("From "+a+" to "+c);
        else {hanoi(n-1,a,c,b);
            System.out.println("From "+a+" to "+c);
            hanoi(n-1,b,a,c);}
        }
    }
}
```

4. 创建一个具有 `radius`、`area` 和 `diameter` 等成员类 `Circle`。添加一个把 `radius` 设置为 1 的构造方法，添加 `setRadius()`、`getRadius()`、`computeDiameter()` 等方法用来计算圆周的直径，添加方法 `computeArea()` 计算圆周的面积。把程序保存为 `Circle.java`。

另外，创建一个名称为 `TestCircle` 的类，该类的 `main()` 方法声明了 3 个 `Circle` 对象。使用 `setRadius()` 方法，给一个 `Circle` 赋一个小的半径值，给另一个 `Circle` 赋一个较大的半径值，不要给第三个 `Circle` 的半径赋值；在程序中保留所赋值。调用每个圆的 `computeDiameter()` 和 `computeArea()` 并显示计算结果。把程序保存为 `TestCircle.java`。

【解答】 解题思路：

(1) 定义圆类，并定义成员变量（在类中定义的变量），并定义题目中要求的各种成员方法。

(2) 定义 `TestCircle` 类，在其 `main()` 方法中，定义几个不同的圆类对象，并输出每个圆的计算后的信息。

参考程序：

```
//Circle.java
public class Circle {
    private double radius,area,diameter;
    public void setRadius(double r){ radius=r;}
    public double getRadius(){return radius;}
    public double computeDiameter(){return 2*radius;}
    public double computeArea(){
```

```

        return Math.PI*radius*radius;
    }
}
//TestCircle.java
public class TestCircle{
    public static void main(String args[]){
        Circle c1,c2,c3;
        c1=new Circle();          c2=new Circle();
        c1.setRadius(10);         c2.setRadius(100.0);
        c3=c1;
        System.out.println("Circle1's Diameter is : "+c3.computeDiameter());
        System.out.println("Circle1's Area is : "+c3.computeArea());
        c3=c2;
        System.out.println("Circle2's Diameter is : "+c3.computeDiameter());
        System.out.println("Circle2's Area is : "+c3.computeArea());
    }
}

```

5. 编写显示员工 ID 和员工姓名的程序。使用两个类，第一个类包括用来设置 ID 和姓名的员工数据和单独的方法。另一个类创建员工对象，并使用对象调用方法。创建若干员工并显示它们的数据。把程序保存为 Employee.java。

【解答】 解题思路：略。

参考程序：

```

//Employee.java
class EmployeeInfo{
    long ID;
    String name;
    void setID(long id){ID=id;}
    void setname(String strName){name=strName;}
    void displayInfo(){
        System.out.println("雇员的 ID: "+ID);
        System.out.println("雇员的姓名: "+name);
    }
}
public class Employee{
    public static void main(String args[]){
        EmployeeInfo e1,e2;
        e1=new EmployeeInfo();e2=new EmployeeInfo();
        e1.setID(12345);e1.setname("John");
        e1.displayInfo();
        e2.setID(28822);e2.setname("Tom");
        e2.displayInfo();
    }
}

```

程序运行结果：

雇员的 ID: 12345

雇员的姓名: John

雇员的 ID: 28822

雇员的姓名: Tom

6. 编写显示若干项货物发货单的程序。该程序应在对应于数量和物品价值的各行上包括物品名称、数量、价格和总价等项。使用两个类, 第一个类包括物品数据及用来获得和设置物品名称、数量和价格的方法。另一个类创建物品的对象并使用对象调用设置和获取方法。

【解答】解题思路:

(1) 按照题意定义两个类。

(2) 定义一个含有 main() 方法的类, 在其中组织上述的两个类, 完成题目要求的功能。

参考程序:

```
//TestargoItem.java
class cargoItem{
    String cargoName;
    long num;double price;double total;
    void SetcargoName(String strc){cargoName=strc;}
    void Setprice(double p){price=p;}
    void Setnum(long n){num=n;}
    String getname(){return cargoName;}
    double Setprice(){return price;}
    double Setnum(){return num;}
    double Gettotal(){return price*num;}
    void displayInfo(){
        System.out.println("Cargo name:"+cargoName);
        System.out.println("Cargo price:"+price);
        System.out.println("Cargo num:"+num);
        System.out.println("Cargo total price:"+Gettotal());
    }
}

public class TestargoItem{
    public static void main(String args[]){
        cargoItem c1;
        c1=new cargoItem();
        c1.SetcargoName("steel");
        c1.Setprice(2004);
        c1.Setnum(20088);
        c1.displayInfo();
    }
}
```

7. 定义一个表示学生的类(Student)。Student 类包括表示学生的学号、姓名、性别、年龄和 3 门课程成绩的信息数据及用来获得和设置学号、姓名、性别、年龄和 3 门课程成绩的方法。创建 TestStudent 类, 在 TestStudent 类中生成 5 个学生对象, 计算 3 门课程的平均成绩, 以及某门课程的最高分和最低分。

【解答】解题思路:

(1) 按照要求定义学生类, 其中包括学生的学号、姓名、性别、年龄和 3 门课程成绩等

成员变量，并定义各种获取和设置成员变量值的方法。

(2) 定义一个主类，在 main()方法中定义 5 名学生对象，求出每门课程的平均成绩、最高分和最低分。

参考程序：

```
class Student{
    String num;String name;
    char sex;int age;
    float score[]=new float[3];
    void setNum(String n){num=n;}
    void setName(String na){name=na;}
    void setSex(char s){sex=s;}
    void setAge(int a){age=a;}
    void setScore(float s1,float s2,float s3){score[0]=s1;score[1]=s2;score[2]=s3;}
    String getNum(){return num;}
    String getName(){return name;}
    char getSex(){return sex;}
    int getAge(){return age;}
    float getScore1(){return score[0];}
    float getScore2(){return score[1];}
    float getScore3(){return score[2];}
}
public class TestStudent{
    public static void main(String args[]){
        Student s[]=new Student[5];
        float ave[]=new float[3];
        float max[]=new float[3];
        float min[]=new float[3];
        int i;
        for(i=0;i<5;i++){
            s[i]=new Student();
            s[i].setNum("1000"+i);s[i].setName("Jsj"+i);
            s[i].setSex('F');s[i].setAge(19);
        }
        s[0].setScore(87,67,99);    s[1].setScore(97,89,78);    s[2].setScore(88,99,96);
        s[3].setScore(76,77,89);    s[4].setScore(90,92,69);
        for(i=0;i<5;i++){
            System.out.println("student"+(i+1));
            System.out.println("Num:"+s[i].getNum());
            System.out.println("Name:"+s[i].getName());
            System.out.println("Sex:"+s[i].getSex());
            System.out.println("Age:"+s[i].getAge());
            System.out.println("Score:"+s[i].getScore1()+"\t"+ s[i].getScore2()+"\t"+s[i].getScore3());
        }
        ave[0]=0;ave[1]=0;ave[2]=0;
        max[0]=s[0].getScore1();max[1]=s[0].getScore2();max[2]=s[0].getScore3();
        min[0]=s[0].getScore1();min[1]=s[0].getScore2();min[2]=s[0].getScore3();
    }
}
```

```

for(i=0;i<5;i++){
    ave[0]+=s[i].getScore1();ave[1]+=s[i].getScore1();ave[2]+=s[i].getScore1();
    if(max[0]<s[i].getScore1())max[0]=s[i].getScore1();
    if(max[1]<s[i].getScore2())max[1]=s[i].getScore2();
    if(max[2]<s[i].getScore3())max[2]=s[i].getScore3();
    if(min[0]>s[i].getScore1())min[0]=s[i].getScore1();
    if(min[1]>s[i].getScore2())min[1]=s[i].getScore2();
    if(min[2]>s[i].getScore3())min[2]=s[i].getScore3();
}
ave[0]/=5;ave[1]/=5;ave[2]/=5;
System.out.println("*****");
System.out.println("三门课程平均成绩:"+ave[0]+" \t"+ ave[1]+" \t"+ave[2]);
System.out.println("课程 1: 最高分: "+max[0]+"    最低分: "+min[0]);
System.out.println("课程 1: 最高分: "+max[1]+"    最低分: "+min[1]);
System.out.println("课程 2: 最高分: "+max[2]+"    最低分: "+min[2]);
}
}
}

```

8. 创建一个名称为 `Pay` 的类，该类包括工作小时、每小时工资、扣缴率、应得工资总额和实付工资等 5 个双精度变量。创建 3 个重载的 `computeNetPay()` 方法。应得工资是工时乘以每小时工资的计算结果。当 `computeNetPay()` 接收代表小时、工资率和扣缴率的数值时，计算出应得工资并从中扣除合理的扣缴金额得出实付工资。当 `computeNetPay()` 接收两个参数时，扣缴率假定为 15%，当 `computeNetPay()` 接收一个参数时，扣缴率假定为 15%，每小时工资率为 4.65%。编写一个测试所有 3 个重载方法的 `main()` 方法。

【解答】解题思路：略。

参考程序：

```

class Pay{
    double WorkHours;    //工作小时
    double PayPerHour;   //每小时工资
    double Discount;     //扣缴率
    double TotalPay;     //应得工资总额
    double NetPay;       //实付工资
    Pay(){ PayPerHour=100.0;}
    void computeNetPay(double wh,double pp,double dc){
        WorkHours=wh;
        TotalPay=WorkHours*PayPerHour;
        NetPay=TotalPay*(1-Discount)*(1-pp);
    }
    void computeNetPay(double wh,double pp){
        WorkHours=wh;
        TotalPay=WorkHours*PayPerHour;
        NetPay=TotalPay*(1-0.15)*(1-pp);
    }
    void computeNetPay(double wh){
        WorkHours=wh;
        TotalPay=WorkHours*PayPerHour;
    }
}

```

```

        NetPay=TotalPay*(1-0.15)*(1-0.0465);
    }
}
public class PayTest {
    public static void main(String args[]){
        Pay p1=new Pay();
        p1.computeNetPay(50,0.05,0.1);
        System.out.println("应发工资: "+p1.TotalPay+"\t 实发工资: "+p1.NetPay);
        System.out.println("*****");
        p1.computeNetPay(50,0.05);
        System.out.println("应发工资: "+p1.TotalPay+"\t 实发工资: "+p1.NetPay);
        System.out.println("*****");
        p1.computeNetPay(50);
        System.out.println("应发工资: "+p1.TotalPay+"\t 实发工资: "+p1.NetPay);
    }
}

```

9. 创建一个名为 `Shirt` 的类，该类具有表示衣领尺寸和袖子长度的数据成员。添加一个接收每个数据成员为参数的构造方法。添加一个命名为 `material` 的 `String` 类变量，将类变量初始值设定为"棉"。编写一个命名为 `TestShirt` 的程序，用来生成具有衣领尺寸和袖子长度的 `Shirt` 的类对象。显示包括材料在内的所有衬衫的数据。

【解答】解题思路：略。

参考程序：

```

class Shirt {
    double collar;    double salve;String material="cotton";
    Shirt(double c,double s){collar=c;salve=s;}
    void display(){
        System.out.println("领子尺寸: "+collar);
        System.out.println("袖子尺寸: "+salve);
        System.out.println("面料: "+material);
    }
}
public class TestShirt {
    public static void main(String args[]){
        Shirt s=new Shirt(10,12);
        s.display();
    }
}

```

4.2 实验四 类与对象

4.2.1 实验目的

- (1) 理解类的封装。
- (2) 掌握类的定义格式。

- (3) 熟悉类的成员：数据成员、方法及构造方法等。
- (4) 熟练掌握方法重载。
- (5) 熟练掌握类变量与实例变量，类方法与实例方法的区别。
- (6) 熟练掌握对象、对象的创建方式。
- (7) 掌握 this 关键字的使用。

4.2.2 实验内容

1. 计算 1 个长方体、1 个截面为正方形的长方体和 1 个立方体的体积和表面积之和，编写一个基于对象的应用程序。定义长方体类，数据成员包括 length（长）、width（宽）、height（高）。要求用成员函数实现下列功能：

- (1) 定义构造函数重载为普通长方体、截面为正方形的长方体、立方体。
- (2) 设置长方体的长、宽、高。
- (3) 计算长方体的体积。
- (4) 计算长方体的表面积。
- (5) 输出体积和表面积。

2. 编写一个完整的 Java Application 程序。包含类 Student、TestStudent，具体要求如下：

(1) Student 类：

1) 属性。

number: String 对象，表示学号。

name: String 对象，表示一个人姓名。

sex: char 类型，用来表示性别。

sepciality: String 对象，表示专业。

address: String 对象，表示家庭地址。

2) 方法。

Student (String number,String name, char sex): 构造函数。

String getName(): 返回姓名。

void setNumber(String num): 设置学号。

void setSepciality(String sep): 设置专业。

void setAddress(String add): 设置家庭地址。

public String toString(): 返回学生的各项信息，包括姓名、性别等上述属性。

(2) 类 TestStudent 作为主类要完成测试功能。

1) 用以下信息生成一个 Student 对象 stu1：

姓名：刘美丽

性别：女

学号：0251102

2) 设置家庭地址：河北邯郸

设置专业信息：计算机软件

3) 输出对象 stu1 的各项信息。

3. 商店销售某一件商品，商店每天公布统一的折扣（discount）。同时允许销售人员在销

售时灵活掌握价格（price），在统一折扣的基础上，对一次购买 10 件以上者，还可以享受 9.5 折优惠。现已知当天 5 名售货员的销售情况为：

售货员编号（num）	销售件数（quantity）	销售单价（price）
101	3	126.8
102	8	125.6
103	10	124.8
104	45	123.4
105	100	121.5

编写程序，计算当天此商品的总销售额 sum，以及每件商品的平均售价。

4.2.3 实验步骤

1. 实现思路：略。

参考程序代码：

```
import java.io.*;
public class Cuboid{
    int length, width, height;
    public Cuboid(int l, int w, int h){ length = l; width = w; height = h;}
    public Cuboid(int l, int h){ length = width = l; height=h;}
    public Cuboid(int h){ length=width=height=h;}
    public void setLWH(int l, int w, int h){ length = l; width = w; height = h;}
    public int volume(){ return length * width * height; }
    public int surfacearea(){return 2 * (length * width + width * height + length * height); }
    public void display(){
        System.out.println("Length = "+length+" , Width = "+width+" , Height =" +height);
        System.out.println("Volume = "+volume()+" , SurfaceArea =" +surfacearea());
    }
    public static void main(String args[]){
        Cuboid c1, c2, c3;
        int vol, area;
        c1 = new Cuboid (1,2, 3); c2=new Cuboid (1, 3); c3=new Cuboid (3);
        vol = c1.volume() + c2.volume() + c3.volume();
        area = c1.surfacearea() + c2.surfacearea() +c3.surfacearea();
        c1.display(); c2.display(); c3.display();
        System.out.println("Total Volume=" +vol +" , Total surfacearea=" +area);
    }
}
```

2. 实现思路：略。

参考程序代码：

```
class Student{
    String number; String name; char sex;
    String speciality; String address;
    public Student(String num, String nam, char sex) {number = num; name = nam; this.sex = sex;}
    void setNumber(String num){number = num; }
```

```

void setSpeciality(String spe){speciality = spe;}
void setAddress(String add){ address = add; }
String getName(){ return name; }
public String toString(){
    return "学号: "+number+ "\n 姓名: " + name + "\n 性别: " + sex+ "\n 专业: " + speciality+ "\n
家庭地址: " + address ;
}
}
}
public class T04E02{
    public static void main(String[] args){
        Student stu1 = new Student("0251102", "刘美丽", '女');
        stu1.setAddress("河北邯郸");
        stu1.setSpeciality("计算机软件");
        System.out.println(stu1.toString());
    }
}
}

```

程序执行结果为：

```

学号： 0551102
姓名： 刘思嘉
性别： 女
专业： 计算机软件
家庭地址： 河北邯郸

```

3. 解题思路：略。

参考程序：

```

import java.text.DecimalFormat;
class Sale{
    static double discount=1.0;           //统一折扣
    static double price;                 //标价
    String salesmanNum;                  //销售员编号
    double salePrice;                    //销售价
    int quantity;                         //销售数量
    public Sale(String num){ salesmanNum = num;}
    static void setDiscount(double d) {discount = d;}
    static void setPrice(double p){price = p;}
    void setQuantity(int qty){quantity = qty;}
    void setSalePrice(double p){
        salePrice = price * discount;
        if (p < price * discount * 0.95)
            System.out.println(salesmanNum+"\s SalePrice is not permit !!!");
        salePrice = p;
    }
    double getSalePrice(){ return salePrice;    }
    int getQuantity(){ return quantity;}
}
public class T04E03{
    public static void main(String[] args){

```

```
Sale s1,s2,s3,s4,s5;
double sum=0;      int qty = 0;      int i;
Sale.setDiscount(0.75);      Sale.setPrice(168);
s1=new Sale("101");  s1.setQuantity(3);  s1.setSalePrice(126.8);
s2=new Sale("102");  s2.setQuantity(8);  s2.setSalePrice(125.6);
s3=new Sale("103");  s3.setQuantity(10);  s3.setSalePrice(124.8);
s4=new Sale("104");  s4.setQuantity(45);  s4.setSalePrice(123.4);
s5=new Sale("105");  s5.setQuantity(100);  s5.setSalePrice(121.5);
sum += s1.getSalePrice() * s1.getQuantity();  sum += s2.getSalePrice() * s2.getQuantity();
sum += s3.getSalePrice() * s3.getQuantity();  sum += s4.getSalePrice() * s4.getQuantity();
sum += s5.getSalePrice() * s5.getQuantity();
qty=s1.getQuantity()+s2.getQuantity()+s3.getQuantity()+s4.getQuantity()+s5.getQuantity();
System.out.println("当天此商品的总销售额: "+sum);
System.out.println("每件商品的平均售价: " +new DecimalFormat("#.##").format(sum/qty));
}
}
```

程序执行结果为：

当天此商品的总销售额：20336.2

每件商品的平均售价：122.51

4.2.4 实验总结

在这次实验中，通过对类的使用，加深对类、对象概念的理解。通过使用类控制符，充分理解类的封装性特征。构造方法、成员方法可以重载，体现了静态多态性，静态数据成员可以通过对象或类名调用，被所有此类对象共享。