

一、案例分析(每题参考分值 5 分)

1、编写一个名为 **Rectangle** 的类表示矩形，他有两个双精度浮点型私有属性：**width** 、**height** ，一个静态字符串型类属性 **color** ，一个默认长和宽都为 1 的构造函数，一个指定长和宽的构造函数，公有的求面积 **calArea()** 和求周长 **calPerimeter()** 的方法，并编写测试该类的 **main()** 方法。



```

public class Exercise {
    + public static void main(String[] args) {
    +     + Rectangle r1 = new Rectangle(4,40);
    +     + r1.setColor("red");
    +     + Rectangle r2 = new Rectangle(3.5,35.9);
    +     + r2.setColor("red");
    +     + System.out.println(r1.toString());
    +     + System.out.println(r2.toString());
    + }
}

class Rectangle {
    + private double width = 1;
    + private double height = 1;
    + private String color = "white";
    + public Rectangle() {
    +     + this(1.0,1.0);
    + }
    + public Rectangle(double width,double height) {
    +     + this.width = width;
    +     + this.height = height;
    +     + color = "white";
    + }
}

```

正确答案:

```

→ public double getWidth() {
→     → return width;
→ }
→ public void setWidth(double width) {
→     → this.width = width;
→ }
→ public double getHeight() {
→     → return height;
→ }
→ public void setHeight(double height) {
→     → this.height = height;
→ }
}
public double getHeight() {
→     → return height;
→ }
→ public void setHeight(double height) {
→     → this.height = height;
→ }
→ public String getColor() {
→     → return color;
→ }
→ public void setColor(String color) {
→     → this.color = color;
→ }
→ public double calArea() {
→     → return width * height;
→ }
→ public double calPerimeter() {
→     → return (width + height) * 2;
→ }
→ public String toString() {
→     → String output = "";
→     → output = "矩形的属性:\n\t宽: " + width + "\n\t高: " + height + "\n\t颜色: " +
→     →     → color + "\n\t面积: " + calArea() + "\n\t周长: " + calPerimeter();
→     → return output;
→ }
}

```

2、编写一个 Java 应用程序 Test 类，实现成员方法 max(a,b) 的重载。具体要求如下：(1) 编写 void max(int a,int b) 成员方法，对两个整数进行大小比较，输出较大的那个整数。(2) 编

写 void max(float a,float b) 成员方法,对两个 float 数进行大小比较,输出较大的那个 float 数。(3) 在主方法中调用成员方法 max(a,b) 。



```
public class Test {  
    → void max(int a,int b){  
    →     → System.out.println(a>b?"a:b");  
    → } * * * * *  
    → void max(float a,float b){  
    →     → System.out.println(a>b?"a:b");  
    → } * * * * *  
    → public static void main(String[] args){  
    →     → Test t=new Test();  
    →     → t.max(3,4);  
    → } * * * * *  
    → }  
}
```

正确答案：

3、创建一个桌子类 Table，该类中有桌子名称，重量（weight ），桌面宽度（width ），长度（length ）和桌子高度（height ）属性，并含有以下几个方法。(1) 构造方法：初始化所有成员变量(2) area(): 计算桌面地面积(3) display(): 在屏幕上输出所有成员变量的值(4) changeWeight(int w): 改变桌子重量在 main() 方法中实现创建一个桌子对象,计算桌面地面积，改变桌子重量，并在屏幕上输出所有桌子的属性值



正确答案：

```
public class Table {  
    * private String name;  
    * private int weight;  
    * private int length;  
    * private int width;  
    * private int height;  
    * public Table() {  
    * name="桌子";  
    * weight=100;  
    * length=10;  
    * width=10;  
    * height=10;  
    * }  
    public void display() {  
    System.out.print("name="+name+"weight="+weight+"length="+length+"width="+width+"height  
    ="+height+"\n");  
    * }  
    * public int area() {
```



```

    * * * * int area;
    * * * * area=length*width;
    * * * * return area;
    * * }
    * * void changeWeight(int w) * * {
    * * weight=w;
    * * }
    public static void main(String[] args) * * {
    * * Table ta=new Table();
    * * ta.dispaly();
    * * int x;
    * * x=ta.area();
    * * System.out.print(" * 面积 *="+x+"\n");
    * * ta.changeweight(10);
    * * ta.dispaly();
    * * }
    }

```

二、论述（每题参考分值 5 分）

4、按以下要求编写程序(1) 编写抽象类 **Animal**，其中抽象方法 **breathe()**。(2) 编写 **Fly** 接口，接口中声明 **fly()** 方法。(3) 编写 **Swim** 接口，接口中声明 **swim()** 方法。(4) 定义 **Bird** 类和 **Fish** 类继承于 **Animal** 类，并分别实现 **Fly** 和 **Swim** 接口。(5) 编写 **Bird** 类和 **Fish** 类的测试程序，并调用其中的 **breathe()**，**fly()**，**swim()** 方法。备注：方法体中使用 **System.out.println()** 方法输出相应提示信息即可。



```
public abstract class Animal {
    → public abstract void breathe();
}
public interface Fly {
    → void fly();
}
```

正确答案：

```
public interface Swim {
    → void swim();
}
public class Bird extends Animal implements Fly {
    → public void fly() {
    → → System.out.println("鸟儿空中飞");
    → }
    → public void breathe() {
    → → System.out.println("鸟儿用肺呼吸");
    → }
}
public class Fish extends Animal implements Swim {
    → public void swim() {
    → → System.out.println("鱼儿水中游");
    → }
    → public void breathe() {
    → → System.out.println("鱼儿用鳃呼吸");
    → }
}
public class Main {
    → public static void main(String[] args) {
    → → Animal bird=new Bird();
    → → bird.breathe();
    → → Animal fish=new Fish();
    → → fish.breathe();
    → → Fly flyingBird=new Bird();
    → → flyingBird.fly();
    → → Swim swimmingFish=new Fish();
    → → swimmingFish.swim();
    → }
}
```

5、编程实现一个复数类 Complex，实现复数的加减乘除运算。该复数类中至少应该有表示实部和虚部的成员数据（成员变量、属性），两个构造函数（带参数，不带参数），加减

运算分别用两个带参数的成员方法 `add` 、 `minus` 来实现， `toString` 方法实现将数据转换为类似 “`3+4i` ” 形式。并计算： `a+b` 、 `a-b` ， 其中 `a=3+4i`,`b=4+3i`.




```
class Complex {
    + private double m; // 实部
    + private double n; // 虚部
    + public Complex() {
        +     m=0;
        +     n=0;
    }
```

正确答案:

```
    }
    + public Complex(double m, double n) {
        +     this.m = m;
        +     this.n = n;
    }
    + // add
    + public Complex add(Complex c) {
        +     return new Complex(m + c.m, n + c.n);
    }
    + // minus
    + public Complex minus(Complex c) {
        +     return new Complex(m - c.m, n - c.n);
    }
    + public String toString() {
        +     String rtr_str = "";
        +     if (n > 0)
        +         rtr_str = "(" + m + " + " + n + "i" + ")";
        +     if (n == 0)
        +         rtr_str = "(" + m + ")";
        +     if (n < 0)
        +         rtr_str = "(" + m + n + "i" + ")";
        +     return rtr_str;
    }
    + public static void main(String[] a) {
        +     Complex b = new Complex(3, 4);
        +     Complex c = new Complex(4, 3);
        +     System.out.println(b + " + " + c + " = " + b.add(c));
        +     System.out.println(b + " - " + c + " = " + b.minus(c));
        + }
}
```

6、假定在当前工作目录下有文件 data.txt ，文件以文本方式存储一系列由空格分隔的实数

（整数或小数形式），要求编程实现从文件中读取数据，计算出这些数的和与平均值。



```
import java.io.*;
import java.util.*;
public class Sum_Avg {
    → public static void main(String[] args) {
    →     → double sum=0, avg=0;
    →     → int n=0, m=0;
    →     → String line, temp;
    →     → StringTokenizer fenxi;
    →     → try {
    →         →     → BufferedReader fin=new BufferedReader(
    →         →     →     →     →     → new InputStreamReader(
    →         →     →     →     →     →     → new FileInputStream("data.txt"));
    →         →     → while((line=fin.readLine())!=null) {
    →         →         →     → fenxi=new StringTokenizer(line, " ");
    →         →         →     → n=fenxi.countTokens();
    →         →         →     → m=m+n;
    →         →         →     → for(int i=0; i<n; i++) {
    →         →             →     → temp=fenxi.nextToken();
    →         →             →     → sum=sum+Double.parseDouble(temp);
    →         →             →     → }
    →         →         →     → }
    →         →     → }
    →         →     → fin.close();
    →         →     → if(m>0)
    →         →         →     → avg=sum/m;
    →         →         →     → System.out.print("和为: "+sum+"均值为: "+avg);
    →         →     → } catch (IOException e) {
    →             →     → e.printStackTrace();
    →             →     → }
    →     → }
    → }
```

正确答案:

```
    →     →     → e.printStackTrace();
    →     →     → }
    →     → }
    → }
```

7、编程实现求正方形和圆形的面积，用户输入相关数据后，可以求得相应的面积数值。要求先设计一个公共的父类 Shape，在其中定义求面积的方法 calArea()，再通过继承，在各子类中覆盖父类相关方法实现面积的计算。



```

import java.util.Scanner;
abstract class Shape {
    → abstract double calArea();
}
class Circle extends Shape {
    → private double radius;
    → public Circle(double radius) {
    →     → this.radius = radius;
    → }
    → double calArea() {
    →     → return Math.PI * radius * radius;
    → }
}
class Square extends Shape {
    → private double side;
    → public Square(double side) {
    →     → this.side = side;
    → }
    → double calArea() {
    →     → return side * side;
    → }
}

```

```

public class CalShape {
    → public static void main(String[] args) {
    →     → Scanner scanner = new Scanner(System.in);
    →     → String tip1 = "计算面积：1-圆形，2-正方形，-0-退出";
    →     → String tip2 = "选择错误，请重新选择！";
    →     → String tip3 = "请输入半径：";
    →     → String tip4 = "请输入边长：";
    →     → String input;
    →     → while (true) {
    →         → System.out.println(tip1);
    →         → input = scanner.next().trim();
    →         → if (!input.matches("0|1|2"))
    →             → System.out.println(tip2);
    →         → else {
    →             → switch (input) {
    →                 → case "1":
    →                     → System.out.println(tip3);
    →                     → double r = scanner.nextDouble();
    →                     → Shape sh = new Circle(r);
    →                     → System.out.println("圆的面积是" + sh.calArea());

```

```
→ → → → → break;
→ → → → case "2":
→ → → → → System.out.println("2");
→ → → → → double s=scanner.nextDouble();
→ → → → → Shape t=new Square(s);
→ → → → → System.out.println("正方形的面积是"+t.calArea());
→ → → → → break;
→ → → → case "0":
→ → → → → scanner.close();
→ → → → → System.exit(0);
→ → → → } → → → →
→ → → }
→ → }
→ }
```

8、定义一个类 Student ，属性包括学号、姓名和成绩；方法为包括构造方法 Student 、增加记录 SetRecord 、得到记录 GetRecord 和主方法 main 。要求构造方法 Student 初始化一条记录(学号、姓名、成绩)，SetRecord 给出学号、姓名和成绩的赋值，GetRecord 通过学号得到考生的成绩。




```

public class Student {
    → private int ID;
    → private String name;
    → private float score;
    → Student(int ID, String name, float score) {
    →     → this.ID = ID;
    →     → this.name = name;
    →     → this.score = score;
    → }
    → public void SetRecord(int ID, String name, float score) {
    →     → this.ID = ID;
    →     → this.name = name;
    →     → this.score = score;
    → }
    → public float getRecord(int ID) {
    →     → if (ID == this.ID)
    →     →     → return this.score;
    →     → else
    →     →     → return -1;
    → }
    → public static void main(String[] args) {
    →     → Student s = new Student(0, "sdfs", 12);
    →     → float Sco = s.getRecord(0);
    →     → System.out.print(Sco);
    → }
}

```

9、定义一个“点”(Point)类用来表示三维空间中的点，类体的成员变量 x,y,z 分别表示三维空间的坐标。类体中具有如下成员方法的定义：(1) 构造方法 Point () 可以生成具有特定坐标的点对象。(2) setX()，setY()，setZ() 分别为设置三个坐标的方法。(3) getDistance() 为计算该点到原点距离的方法。



```

class Point{
    → double x,y,z; .....
    → Point(double _x,double _y,double _z){ ...
    →     → x=_x;
    →     → y=_y;
    →     → z=_z; ...
    → } ...
    → void setX(double _x){
    →     → x=_x; ...
    → } .....
    → void setY(double _y){
    →     → y=_y; ...
    → } .....
    → void setZ(double _z){
    →     → z=_z; ...
    → } .....
    → double getDistance(){
    →     → return Math.sqrt(x*x+y*y+z*z); ...
    → } ...
}

```

10、编写一个字符界面的 Java Application 程序，接受用户输入的 10 个整数，并输出这 10 个整数的最大值和最小值，要求在程序中处理 IOException 和用户误操作引起的 NumberFormatException 。



以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：<https://d.book118.com/768127005062006134>