

设计Point 类并用此类构建一个三角形类 (Triangle) , 编程求三角形的面积。请把程序代码和输出结果的截图提交到答案中。

```
1  import java.util.Scanner;    //用于读取输入
2
3  class Point{
4      double x;
5      double y;
6      public Point(double i,double j){
7          x=i;
8          y=j;
9      }
10     public double distance(Point p) {
11         return Math.sqrt((p.x-this.x)*(p.x-this.x)+(p.y-this.y)*(p.y-
12         this.y));
13     }
14 }
15
16 class Triangle {
17     Point a;
18     Point b;
19     Point c;
20     public Triangle(Point x, Point y,Point z){
21         a=x;b=y;c=z;
22     }
23     public void Area() {
24         double s1 = a.distance(b);    //计算a和b之间的距离
25         double s2 = b.distance(c);
26         double s3 = c.distance(a);
27         if (s1 + s2 > s3 && s1 + s3 > s2 && s2 + s3 > s1) {
28             double p = (s1 + s2 + s3) / 2;
29             double area = Math.sqrt(p * (p - s1) * (p - s2) * (p - s3));
30             System.out.println("此三角形面积为: " + area);
31         } else
32             System.out.println("此三边不能构成三角形");
33     }
34 }
35
36 public class E1_03{
37     public static void main(String[] args) {
38         Scanner sc = new Scanner(System.in);
39         double x1,x2,x3,y1,y2,y3;
40         System.out.println("请输入第一个点的坐标(x,y): ");
41         x1=sc.nextDouble();
42         y1=sc.nextDouble();
43         System.out.println("请输入第二个点的坐标(x,y): ");
44         x2=sc.nextDouble();
45         y2=sc.nextDouble();
46         System.out.println("请输入第三个点的坐标(x,y): ");
47         x3=sc.nextDouble();
48         y3=sc.nextDouble();
49         Point a = new Point(x1,y1);
50         Point b = new Point(x2,y2);
```

```

50         Point c = new Point(x3,y3);
51         Triangle test = new Triangle(a,b,c);
52         test.Area();
53     }
54 }

```

2. (论述题, 5分)输入若干整数，并求和，直到结束标志999为止。在输入过程中，自动忽略掉输入的非整数。提示：用Scanner的nextInt()方法转换输入信息到整数，如果输入信息为非整数，方法将抛出名为InputMismatchException的异常，该异常封装在java.util包中。请把程序代码和输出结果的截图提交到答案中。

```

1  import java.util.InputMismatchException;
2  import java.util.Scanner;
3
4  public class catchFloat {
5      public static void main(String[] args){
6          Scanner sc=new Scanner(System.in);
7          int number = 0;
8          int sum = 0;
9          System.out.println("请输入整数进行累加（结束标志为999）：");
10         while(true) {
11             try {
12                 int num = sc.nextInt(); //读到非整数则抛出
InputMismatchException异常
13                 if (num == 999)
14                     break;
15                 sum += num;
16             } catch (InputMismatchException e) {
17                 System.out.println("输入的不是整数，已自动忽略。");
18                 sc.next(); // 忽略非整数输入
19             }
20         }
21         System.out.println("所有输入整数的和为：" + sum);
22     }
23 }

```

为某研究所编写一个通用程序，用来计算每一种交通工具行驶1000公里所需的时间。已知每种交通工具的参数都是3个整数A、B、C的表达式，现有两种工具Car007和Plane，其中Car007的速度计算公式为 $A*B/C$ ，Plane的速度计算公式为 $A+B+C$ 。需要编写三个类程序ComputeTime.java、Plane.java、Car007.java和一个接口程序Common.java，要求在未来如果增加第三种交通工具的时候，不必修改以前的任何程序，只需要编写新的交通工具的程序。其运行过程如下，从命令行输入ComputeTime的四个参数，第一个是交通工具的类型，第二、三、四个参数分别是整数A、B、C，例如，计算Plane的时间“java ComputeTime Plane 20 30 40”，计算Car007的时间“java ComputeTime Car007 23 34 45”，如果第三种交通工具为Ship，则只需要编写Ship.java，运行时输入“java ComputeTime Ship 22 33 44”。提示：充分利用接口的概念，接口对象作为参数。实例化一个对象的另外一种办法是“Class.forName(str).newInstance()”，例如需要实例化一个Plane对象的话，则只需调用Class.forName(“Plane”).newInstance()。请把程序代码和输出结果的截图提交到答案中。

```

1  public interface Common {
2      public float speed();

```

```

3     public void init(int x,int y,int z);
4 }
5
6 public class Plane implements Common{
7     int A;
8     int B;
9     int C;
10    public float speed(){
11        return (float)(A+B+C);
12    }
13    public void init(int x,int y,int z){
14        A=x;B=y;C=z;
15    }
16 }
17
18 import java.util.Scanner;
19
20 public class ComputeTime {
21     public static void main(String[] args)throws InstantiationException,
IllegalAccess Exception, ClassNotFoundException{
22         String type;
23         Scanner sc = new Scanner(System.in);
24
25         System.out.println("java ComputeTime ");
26
27         type=sc.next();
28         int a,b,c;
29         a=sc.nextInt();
30         b=sc.nextInt();
31         c=sc.nextInt();
32         sc.close();
33
34         Common vehicle = (Common) Class.forName(type).newInstance();
35         vehicle.init(a, b, c);
36
37         float time = 1000 / vehicle.speed();
38         System.out.println("此交通工具行驶1000公里所需的时间为: " + time + "小
时");
39     }
40 }
41

```

## GUI

```

1 import java.awt.*;
2 import java.awt.event.*;
3 import javax.swing.*;
4 //继承了JFrame（用于创建窗口）并实现了ActionListener接口（用于处理按钮点击事件）
5 public class ColorPane extends JFrame implements ActionListener {
6     private JButton buttons[];
7     private String names[]=
{"blue","cyan","green","magenta","orange","pink","red","white","yellow"};
8     private boolean toggle=true;

```

```

9     private Container container;
10    private GridLayout grid;
11    public ColorPane(){ //初始化窗口和按钮
12        super("ColorPane");
13        grid=new GridLayout(3,3,5,5);
14        container=getContentPane();
15        container.setLayout(grid);
16        buttons=new JButton[names.length];
17        for(int count=0;count<names.length;count++){
18            buttons[count]=new JButton(names[count]);
19            buttons[count].addActionListener(this);
20            container.add(buttons[count]);
21        }
22        setSize(800,800);
23        setVisible(true);
24
25    }
26    public void actionPerformed(ActionEvent e){ //添加监听器
27        if(e.getSource()==buttons[0]){
28            buttons[0].setBackground(Color.BLUE);
29            buttons[0].updateUI();
30        }
31        if(e.getSource()==buttons[1]){
32            buttons[1].setBackground(Color.CYAN);
33            buttons[1].updateUI();
34        }
35        if(e.getSource()==buttons[2]){
36            buttons[2].setBackground(Color.GREEN);
37            buttons[2].updateUI();
38        }
39        if(e.getSource()==buttons[3]){
40            buttons[3].setBackground(Color.MAGENTA);
41            buttons[3].updateUI();
42        }
43        if(e.getSource()==buttons[4]){
44            buttons[4].setBackground(Color.ORANGE);
45            buttons[4].updateUI();
46        }
47        if(e.getSource()==buttons[5]){
48            buttons[5].setBackground(Color.PINK);
49            buttons[5].updateUI();
50        }
51        if(e.getSource()==buttons[6]){
52            buttons[6].setBackground(Color.RED);
53            buttons[6].updateUI();
54        }
55        if(e.getSource()==buttons[7]){
56            buttons[7].setBackground(Color.WHITE);
57            buttons[7].updateUI();
58        }if(e.getSource()==buttons[8]){
59            buttons[8].setBackground(Color.YELLOW);
60            buttons[8].updateUI();
61        }
62    }
63
64    public static void main(String[] args){

```

```

65         ColorPane application=new ColorPane();
66         application.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
67
68     }
69 }

```

用HashMap模拟一个网上购物车。要求：从键盘输入5本书的名称、单价、购买数量，将这些信息存入一个HashMap，然后将该HashMap作为参数调用方法getSum(HashMap books)，该方法用于计算书的总价并返回。提示：键盘输入可以使用Scanner类。

```

1  //class Book {
2  //    private String name;
3  //    private double price;
4  //    private int quantity;
5  //    public Book(String name, double price, int quantity) {
6  //        super();
7  //        this.name = name;
8  //        this.price = price;
9  //        this.quantity = quantity;
10 //    }
11 //    public void setName(String name) {
12 //        this.name = name;
13 //    }
14 //    public double getPrice() {
15 //        return price;
16 //    }
17 //    public void setPrice(double price) {
18 //        this.price = price;
19 //    }
20 //    public int getQuantity() {
21 //        return quantity;
22 //    }
23 //    public void setQuantity(int quantity) {
24 //        this.quantity = quantity;
25 //    }
26 //
27 //}
28 //
29 //
30 //
31

```

实现文件HelloWorldDemo.java的读写，要求如下：

- (1) 打开文件HelloWorldDemo.java，并在控制台显示文件内容；
- (2) 将文件HelloWorldDemo.java的内容，复制写入文件HelloWorldDemo.txt中。

```

1  import java.io.*;
2
3  public class HelloWorldDemo
4  {

```

```

5      public static void main(String[] args)
6      {
7          try
8          {
9              //读取文件内容
10             FileInputStream file=new FileInputStream("D:\\-COOH\\卓班\\大二下
\\Java程序设计\\实验课\\5.1第三次实验\\E3\\src\\HelloWorldDemo.java");
11             BufferedInputStream inputFile=new BufferedInputStream(file);
12             StringBuilder content= new StringBuilder();        //存取文件内容
13             byte[] buffer=new byte[2048]; // 创建一个字节数组作为缓冲区，用于临时
存储读取的数据
14             int test=0; //存取每次读的字节数
15             while((test=inputFile.read(buffer))!=-1)
16             {
17                 assert false;
18                 content.append(new String(buffer, 0, test));
19             }
20             System.out.println(content);
21
22             try {
23                 FileOutputStream file2=new
FileOutputStream("HelloWorldDemo.txt");
24                 BufferedOutputStream outputFile=new
BufferedOutputStream(file2);
25                 String content2=content.toString();        //转为字符串
26
27                 outputFile.write(content2.getBytes(),0,content2.getBytes().length); // 将字
符串转换为字节数组，并写入到文件中
28                 outputFile.flush(); // 刷新缓冲区，确保所有的数据都被写入到文件中
29                 outputFile.close(); //释放资源
30             } catch (Exception e){
31                 e.printStackTrace();
32             }
33             inputFile.close();
34         } catch (Exception e) {
35             e.printStackTrace();
36         }
37     }
38 }

```

设计一个火车售票模拟程序Ticket.java。假如火车站有100张火车票要卖出，现在有5个售票点同时售票，用5个线程模拟这5个售票点的售票情况。要求如下：

- (1) 打印出每个售票点所卖出的票号；
- (2) 各售票点不能售出相同票号的火车票。

输出结果示例如下图所示，左侧数字为售票点编号，右侧数字为卖出火车票编号。

5\_\_\_\_100

2\_\_\_\_99

4\_\_\_\_98

1\_\_\_\_97

3\_\_\_\_96

.....

4\_\_\_\_3

3\_\_\_\_2

2\_\_\_\_1

```
1  import java.util.Random;
2  public class Ticket implements Runnable
3  {
4      public int num=100;
5
6      //重写run () 函数
7      public void run() {
8
9          while(num>0) {
10             try {
11                 Thread.sleep(100);
12                 synchronized(this) { //使用同步代码块
13                     if(num>0) {
14
15                         System.out.println(Thread.currentThread().getName()+"_____"+num);
16                         num--;
17                     }
18                 } catch (Exception e) {
19                     e.printStackTrace();
20                 }
21             }
22         }
23         public static void main(String []args)
24         {
25             Ticket test=new Ticket();
26             for(int thread=1;thread<=5;thread++)
27             {
28                 new Thread(test,""+ thread).start();
29             }
30         }
31     }
```

设计基于流套接字的客户机/服务器通信程序。要求如下：

- (1) 分别编写服务端程序和客户端程序（端口号1888）。
- (2) 客户端向服务器发送信息“地瓜、地瓜，我是土豆，请回答！”，服务器接收到信息并显示出来。
- (3) 同时服务器将信息反馈到客户端“土豆，土豆，我是地瓜，信息已收到！”，客户端收到信息并显示。

```

1  import java.net.*;
2  import java.io.*;
3
4  public class Server {
5      public static void main(String[] args) throws Exception
6      {
7          ServerSocket serversocket = new ServerSocket(1888);
8          System.out.println("等待客户端的连接请求");
9          Socket socket= serversocket.accept();
10         BufferedReader buffer=null;
11         PrintWriter print=null;
12         try{
13             buffer = new BufferedReader(new
14             InputStreamReader(socket.getInputStream()));
15             print = new PrintWriter(socket.getOutputStream(), true);
16             String line = buffer.readLine();
17             System.out.println("客户端: "+line);
18             print.println("土豆, 土豆, 我是地瓜, 信息已收到! ");
19             buffer.close();
20             print.close();
21             socket.close();
22         }
23         catch(IOException e){
24             e.printStackTrace();
25         }
26     }
27
28
29     //client
30     import java.net.*;
31     import java.io.*;
32
33     public class Client {
34         public static void main(String[] args) {
35             try{
36                 Socket socket =new Socket("127.0.0.1",1888);
37                 System.out.println("客户端启动");
38                 PrintStream print = new PrintStream(socket.getOutputStream());
39                 BufferedReader buffer = new BufferedReader(new
40                 InputStreamReader(socket.getInputStream()));
41                 print.println("地瓜、地瓜, 我是土豆, 请回答! ");
42                 String response = buffer.readLine();
43                 System.out.println("服务端: " + response);
44                 print.close();
45                 buffer.close();
46                 socket.close();
47             }
48             catch(Exception e){
49                 e.printStackTrace();
50             }
51         }
52     }

```



