**作业1**

**（考试之前上传至FTP服务器的作业1目录）**

**学号：222022321062009**

**姓名：吴孜远**

1、（30分）**定义类**MyPoint用于表示二维空间中的一个点，包含特征如下：

（1）用x、y两个字段表示坐标，并设计Getter和Setter函数；

（2）无参数构造器用于创建一个坐标为（0,0）的点对象；

（3）带坐标参数的构造器；

（4）计算与其他点之间距离的函数double distance(MyPoint otherPoint)；

（5）计算与特定坐标距离的函数double distance(double x, double y)；

**解题要求**：（1）绘制类图，并编写代码实现该类。（2）编写Test类代码，创建（0，0）和（10，10）两个点对象，并计算两个点之间的距离。

图形用户界面, 应用程序, 表格

描述已自动生成

public class MyPoint {

    private double x;

    private double y;

    //无参构造，初始化为(0,0)

    MyPoint(){

        x = 0;

        y = 0;

    }

    //有参构造器

    MyPoint(double pointX, double pointY){

        x = pointX;

        y = pointY;

    }

    public double distance(MyPoint otherPoint) {

        double distance = Math.sqrt(Math.pow(otherPoint.getX() - x,2) + Math.pow(otherPoint.getY() - y,2));

        return distance;

    }

    public double distance(double x, double y) {

        double distance = Math.sqrt(Math.pow(x - this.x,2) + Math.pow(y - this.y,2));

        return distance;

    }

    /\*

     \* getters and setters

     \*/

    public double getX() {

        return x;

    }

    public void setX(double x) {

        this.x = x;

    }

    public double getY() {

        return y;

    }

    public void setY(double y) {

        this.y = y;

    }

}

Test类：

public class Test {

    public static void main(String[] args) {

        MyPoint a = new MyPoint();

        MyPoint b = new MyPoint(10,10);

        double dist = a.distance(b);

        System.out.println(dist);

    }

}

输出：

图形用户界面, 应用程序, Word

描述已自动生成

2、（30分）设计一个绘画框架，包含画布和形状两个要素，画布（Canvas）通过draw函数将某种形状绘制到界面上。如下代码所示，现在有圆形和矩形两种形状，画布需要根据形状设计绘制函数。请利用接口和多态特性对现有设计进行改进，使画布不依赖于具体形状，提高程序可扩展性和复用性。

|  |
| --- |
| class Canvas{  public void draw(Circle c) {  System.***out***.println("绘制" + c.getShape());  }  public void draw(Rectangle r) {  System.***out***.println("绘制" + r.getShape());  }  }  class Circle {  public String getShape() {  return "圆形";  }  }  class Rectangle {  public String getShape() {  return "矩形";  }  } |

**解题要求**：（1）绘制类图，并编写代码实现该类。（2）编写Test类代码，创建画布、圆形、矩形等对象，模拟绘制两种图形的过程。

图示

描述已自动生成

public interface Shape {

    String getShape();

}

public class Circle implements Shape{

    public String getShape() {

        return "圆神？启动！";

    }

}

public class Rectangle implements Shape{

    public String getShape() {

        return "矩形";

    }

}

public class Canvas {

    public void draw(Shape c) {

        System.out.println("绘制" + c.getShape());

    }

}

public class Test {

    public static void main(String[] args) {

        Canvas canvas = new Canvas();

        Circle c = new Circle();

        Rectangle r = new Rectangle();

        canvas.draw(c);

        canvas.draw(r);

    }

}

**输出：**

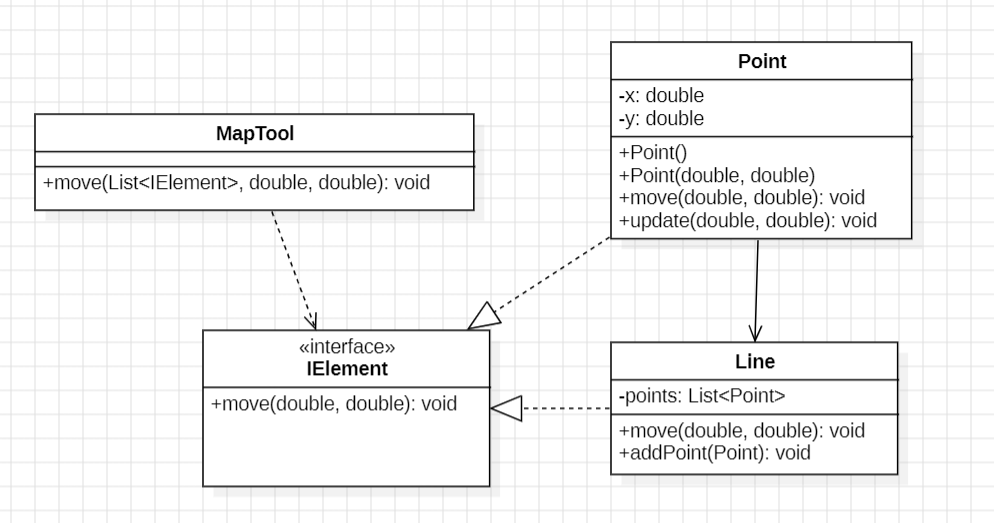
文本

描述已自动生成

3、（40分）设计一个地图程序：（1）地图包含点（Point）和线（Line）两个要素，点要素包含一个坐标点，线要素包含一个有序的坐标列表。（2）在工具类MapTool中设计一个批量平移的工具函数move(List<X> list, double dx, double dy)，表示将list中的所有要素按给定的增量dx,dy进行平移。结合接口、多态等知识思考这里的X应该为什么类型？针对Point和Line两个类，是否需要设计一个接口或抽象类？Point和Line两个类中的移动功能如何实现？

**解题要求**：（1）绘制类图，并编写代码实现。（2）编写Test类代码，创建由多个点、线对象组成的列表，调用move函数对其进行批量平移。

答：Point 和Line都可以作为要素IElement（没想到好的名字，Element好像util中已经有了）接口的实现，X就是IElement接口类型，或者抽象类也行，单纯喜欢写成接口。



public interface IElement {

    void move(double dx, double dy);

}

public class Point implements IElement{

    private double x;

    private double y;

    //无参构造，初始化为(0,0)

    Point(){

        x = 0;

        y = 0;

    }

    //有参构造器

    Point(double pointX, double pointY){

        x = pointX;

        y = pointY;

    }

    @Override

    public void move(double dx, double dy) {

        x += dx;

        y += dy;

    }

    /\*

     \* getters and setters

     \*/

    public void update(double newX, double newY) {

        x = newX;

        y = newY;

    }

    public double getX() {

        return x;

    }

    public void setX(double x) {

        this.x = x;

    }

    public double getY() {

        return y;

    }

    public void setY(double y) {

        this.y = y;

    }

}

import java.util.ArrayList;

import java.util.List;

public class Line implements IElement {

    private List<Point> points = new ArrayList<Point>();

    @Override

    public void move(double dx, double dy) {

        for (Point p : points) {

            p.update(p.getX() + dx, p.getY() + dy);

        }

    }

    /\*

     \* setters and getters

     \*/

    public void addPoint(Point p) {

        points.add(p);

    }

    public List<Point> getPoints() {

        return points;

    }

    public void setPoints(List<Point> points) {

        this.points = points;

    }

}

import java.util.List;

public class MapTool {

    public void move(List<IElement> list, double dx, double dy){

        for(IElement e : list) {

            e.move(dx, dy);

        }

    }

}

import java.util.ArrayList;

import java.util.List;

public class Test {

    public static void main(String[] args) {

        List<IElement> list = new ArrayList<IElement>();

        Point a = new Point();

        Point b = new Point(1,1);

        Point c = new Point(2,2);

        Point d = new Point(3,3);

        Line l = new Line();

        l.addPoint(b);

        l.addPoint(c);

        l.addPoint(d);

        list.add(a);

        list.add(b); //b将会被move两次

        list.add(l);

        MapTool tool = new MapTool();

        tool.move(list, 1, 1);

        System.out.println("x of a: "+a.getX());//输出1

        System.out.println("x of b: "+b.getX());//输出3

    }

}

**输出：**

