**Министерство образования и науки Российской Федерации**

*Федеральное государственное бюджетное образовательное учреждение   
высшего профессионального образования*   
**«АЛТАЙСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ИМ. И.И. ПОЛЗУНОВА» (АлтГТУ)**

Факультет информационных технологий

Кафедра прикладной математики

Отчёт по дисциплине «Алгоритмы и структуры данных»

Лабораторная работа №8

«Вычислительная геометрия»

Выполнил:

Студент гр. ПИ-32

Глушков Г.Г.

Проверил:

ст. преподаватель Н.Д. Бубнова

Барнаул 2015

* Формулировка задачи:

На множестве точек проверить возможность провести линию через 3 точки.

* Алгоритм решения:
* Расставим точку на плоскости . Координаты хранятся в классе Line.
* На каждой итерации добавления точек проверяем возможность построения линии.
* Алгоритм поиска возможности расположения линии : Когда количество точек >= 3 Начинается проверка формулы. Пусть n1 , n2, n3 объекты класса Point ,которые хранят координаты точек . Правило выбора точек является полный перебор Формула – если (n3.x – n1.x ) / (n2.x – n1.x) == (n3.y – n1.y)/(n2.y – n1.y) , то существует возможность провести линию через эти точки. На каждой итерации добавления точки записываем её в объект PointArray ,который хранит все точки.
* Текст программы с комментариями:

//Главный класс

import java.awt.\*;

import java.awt.event.\*;

import java.io.File;

import java.util.ArrayList;

import javax.swing.\*;

import javax.swing.event.\*;

import javax.swing.text.html.parser.Parser;

public class TouchScreen extends JPanel implements ActionListener, ChangeListener, MouseListener {

//Создание панели для отрисовки

private static final long serialVersionUID = 1L;

PointArray p = new PointArray();

ArrayList<Integer> Optoin = new ArrayList();

static JPanel contentPane;

private JPanel picture1;

private JPanel picture2;

private JLabel numbers;

private int num = 0;

private JComboBox boxPoint;

private int fx = 0 , fy = 0;

/\*\* Creates a new instance of RunProgram \*/

public TouchScreen() {

setLayout(new BorderLayout());

contentPane = new JPanel(new BorderLayout());

contentPane.setBorder(BorderFactory.createRaisedBevelBorder());

numbers = new JLabel("0");

boxPoint = new JComboBox();

boxPoint.setEditable(true);

boxPoint.addItem("Points");

picture1 = new JPanel();

picture1.addMouseListener(this);

picture1.setBorder(BorderFactory.createLoweredBevelBorder());

picture1.setPreferredSize(new Dimension(850, 600));

picture2 = new JPanel();

MyItemListener actionListener = new MyItemListener();

boxPoint.addItemListener(actionListener);

picture2.add(boxPoint);

picture2.add(numbers);

picture2.setBackground(Color.GREEN);

picture2.setBorder(BorderFactory.createLoweredBevelBorder());

picture2.setPreferredSize(new Dimension(200, 600));

contentPane.add(picture1, BorderLayout.WEST);

contentPane.add(picture2, BorderLayout.EAST);

}

/\*\*59. \* @param args the command line arguments\*/

public static void main(String[] args) {

JFrame frame = new JFrame("ImageViewer");

frame.add(new TouchScreen());

frame.setSize(1100, 600);

frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

frame.setContentPane(contentPane);

frame.setVisible(true);

}

public void mousePressed(MouseEvent e) {

//Получения координат

/\* x = e.getX();

y = e.getY();

System.out.println("x = " + x + "y = " + y);

drawCircle(e.getX()-(radius/2), e.getY()-(radius/2), e.getSource(),false);

repaint();\*/

}

public void mouseReleased(MouseEvent e) {

x = e.getX();

y = e.getY();

Point pp = new Point(x , y);

p.addPoint(pp);

System.out.println("x = " + x + "y = " + y);

drawCircle(e.getX() - (radius / 2), e.getY() - (radius / 2), e.getSource(), true);

//boxPoint.addItem("x = " + x + " y = " + y);

repaint();

}

public void drawCircle(int x, int y, Object source, boolean fill) {

//Отрисовка точек и линий ,если такие существуют.

if(source instanceof JPanel) {

Graphics g = ((JComponent) source).getGraphics();

if(p.getSize() >= 3){

Optoin = p.getRight();

if(p.Result()==0){

System.out.println("The line was built successfully");

// boxPoint.removeAllItems();

numbers.setText(String.valueOf((p.CountLine)));

for(int i =0; i < Optoin.size(); i=i+6){

num ++;

g.setColor(Color.RED);

g.drawLine(Optoin.get(i), Optoin.get(i + 1), Optoin.get(i + 2), Optoin.get(i + 3));

g.drawLine(Optoin.get(i + 2), Optoin.get(i + 3), Optoin.get(i + 4), Optoin.get(i + 5));

boxPoint.addItem(num + " x1= " + Optoin.get(i) + " y1= " + Optoin.get(i + 1));

boxPoint.addItem(num + " x2= " + Optoin.get(i + 2)+ " y2= " + Optoin.get(i + 3));

boxPoint.addItem(num + " x3= " + Optoin.get(i + 4)+ " y3= " + Optoin.get(i + 5));

}

g.setColor(Color.BLACK);

}

else{

System.out.println("This curve is not a line");

}

}

g.drawOval(x - radius, y - radius, 2 \* radius, 2 \* radius);

g.setColor(Color.BLACK);

if (fill) {

g.fillOval(x - radius, y - radius, 2 \* radius, 2 \* radius);

}

}

}

public void mouseClicked(MouseEvent e) {}

public void mouseEntered(MouseEvent e) {

}

public void mouseExited(MouseEvent e) {}

int x, y;

int radius = 2;

@Override

public void stateChanged(ChangeEvent arg0) {

// TODO Auto-generated method stub

}

@Override

public void actionPerformed(ActionEvent arg0) {

// TODO Auto-generated method stub

}

class MyItemListener implements ItemListener {

public void Parse(String s){

fx = Integer.valueOf(s.substring((s.indexOf("=") + 2), s.indexOf("y")-1));

fy = Integer.valueOf(s.substring(s.lastIndexOf("=")+2));

}

@Override

public void itemStateChanged(ItemEvent e) {

JComboBox cb = (JComboBox) e.getSource();

Object item = e.getItem();

if (e.getStateChange() != ItemEvent.SELECTED) {

this.Parse(item.toString());

}

}

}

}

//Класс линия

/\*\*

\* Created by Gleb on 16.11.2015.

\*/

public class Line {

private Point sP;

private Point mP;

private Point eP;

public Line(){}

public Line(Point sp ,Point mp, Point ep){

this.eP = ep; this.mP = mp; this.sP = sp;

}

public Point getsP() {

return this.sP;

}

public Point getmP() {

return this.mP;

}

public Point geteP() {

return this.eP;

}

public void setsP(Point sP) {

this.sP = sP;

}

public void setmP(Point mP) {

this.mP = mP;

}

public void seteP(Point eP) {

this.eP = eP;

}}

//класс Точка

/\*\*

\* Created by Gleb on 14.11.2015.

\*/

public class Point {

private float cX;

private float cY;

public Point( int x ,int y){

this.cX = x;

this.cY = y;

}

public Point(){}

public float getcX(){

return this.cX;

}

public float getcY(){

return this.cY;

}

}

//Класс Массив Точек

import java.util.ArrayList;

/\*\*

\* Created by Gleb on 14.11.2015.

\*/

public class PointArray {

private ArrayList<Point> pointer = new ArrayList<Point>();

private ArrayList<Line> linear = new ArrayList<Line>();

private int Result = 1;

private boolean Repeat = true;

private int Parametr = 1;

public int CountLine =0;

//Добавление в массив

public void addPoint(Point point){

this.pointer.add(point);

this.Result = 1;

}

public Point getPoint(int index){

try {

if (!this.pointer.isEmpty()) {

return this.pointer.get(index);

}

}

catch (Exception e){

// e.printStackTrace();

}

return null;

}

public int getSize(){

if(this.pointer.isEmpty()){

return 0;

}

else{

return this.pointer.size();

}

}

public int Result(){

return this.Result;

}

public ArrayList getRight(){

ArrayList<Integer> arr = new ArrayList();

//Создаем три точки для проверки алгоритма

Point p1 = new Point();

Point p2 = new Point();

Point p3 = new Point();

Line line = new Line();

int iter = 0;

for(int i =0; i < this.pointer.size(); i++){

p1 = this.pointer.get(i);

for(int j =0; j < this.pointer.size();j++){

if(i==j ) continue;

p2 = this.pointer.get(j);

for (int k =0; k < this.pointer.size(); k++){

if(k==i || k==j) continue;

p3 = this.pointer.get(k);

**//Проверка формулы . Если выполняется ,то добавляем эти точки в массив линий.**

try {

if(((p3.getcX()-p1.getcX())/(p2.getcX() - p1.getcX())) == (((p3.getcY() - p1.getcY())/(p2.getcY()-p1.getcY())))){

if(this.Parametr ==1) {

line.setsP(p1);

line.setmP(p2);

line.seteP(p3);

}

if(iter==0 && this.Repeat){

this.Result = 0;

line.setsP(p1);

line.setmP(p2);

line.seteP(p3);

this.CountLine ++;

arr.add((int)p1.getcX());

arr.add((int) p1.getcY());

arr.add((int) p2.getcX());

arr.add((int) p2.getcY());

arr.add((int)p3.getcX());

arr.add((int)p3.getcY());

linear.add(line);

}

this.Repeat = true;

if(this.Parametr ==1){

linear.add(line);

}

this.Parametr = 0;

iter = 0;

}

}

catch (Exception e){

}

}

}

}

return arr;

}

}

