import java.applet.Applet;

import java.awt.Canvas;

import java.awt.Color;

import java.awt.Dimension;

import java.awt.Graphics;

import java.util.Random;

class Point\_ extends Canvas {

private static final long serialVersionUID = 1L;

Color clr;

int x, y;

Dimension dim = фnew Dimension(20, 20);

public Point\_ (int appWidth, int appHeight) {

super();

setMaximumSize(dim);

setBackground(null);

setBounds(0, 0, dim.width, dim.height);

clr = Color.black;

x = new Random().nextInt(appWidth - dim.width) + dim.width / 2;

y = new Random().nextInt(appHeight - dim.height) + dim.height / 2;

}

public void paint (Graphics g) {

g.setColor(clr);

g.fillOval(0, 0, dim.width, dim.height);

}

public void setColor (Color clr) {

this.clr = clr;

}

public Dimension getMinimumSize () { return dim; }

public Dimension getPreferredSize () { return dim; }

}

class Line\_ {

Color clr;

int x1, y1, x2, y2;

public Line\_ (Color c, int x1, int y1, int x2, int y2) {

clr = c;

this.x1 = x1;

this.y1 = y1;

this.x2 = x2;

this.y2 = y2;

}

public int compare (Point\_ p) {

double tg = (double)(y2 - y1) / (x2 - x1);

int xt = (int)(x1 - (double)(y1 - p.y) / tg);

if ( p.x < xt )

return -1;

else if ( p.x > xt )

return 1;

int yt = (int)(y1 - tg \* (x1 - p.x));

if ( p.y < yt )

return 1;

else if ( p.y > yt )

return -1;

return 0;

}

}

public class task\_19 extends Applet {

private static final long serialVersionUID = 1L;

static final int CX = 600, CY = 500;

Canvas point\_first;

Canvas point\_second;

Line\_ Line\_;

String s;

public Color getHtmlColor (String rgb, Color def) {

if ( rgb == null || rgb.charAt(0) != '#' )

return def;

try {

return new Color(Integer.parseInt(rgb.substring(1), 16));

} catch (NumberFormatException e) {

return def;

}

}

public void init () {

setSize(CX, CY);

setLayout(null);

setBackground(Color.white);

Color clr = Color.BLACK;

int x1 = 0, y1 = 0, x2 = 0, y2 = 0;

try {

clr = getHtmlColor(getParameter("Line\_Color"), Color.BLACK);

x1 = Integer.parseInt((getParameter("x1") == null)?"50":getParameter("x1"));

y1 = Integer.parseInt((getParameter("y1") == null)?"50":getParameter("y1"));

x2 = Integer.parseInt((getParameter("x2") == null)?"550":getParameter("x2"));

y2 = Integer.parseInt((getParameter("y2") == null)?"450":getParameter("y2"));

} catch (Exception e) {}

Line\_ = new Line\_(clr, x1, y1, x2, y2);

point\_first = new Point\_(this.getWidth(), this.getHeight());

point\_second = new Point\_(this.getWidth(), this.getHeight());

int ans\_first = ((Line\_)Line\_).compare((Point\_)point\_first);

int ans\_second = ((Line\_)Line\_).compare((Point\_)point\_second);

Color cl\_first = Color.red;

Color cl\_second = Color.red;

s = new String();

switch (ans\_first) {

case -1: cl\_first = Color.red; break;

case 1: cl\_first = Color.green; break;

default: cl\_first = Color.yellow;

}

switch (ans\_second) {

case -1: cl\_second = Color.red; break;

case 1: cl\_second = Color.green; break;

default: cl\_second = Color.yellow;

}

if(ans\_second == ans\_first){

if(ans\_second == -1)

s="точки лежат слева от прямой";

if(ans\_second == 1)

s="точки лежат справа от прямой";

if(ans\_second == 0)

s="точки лежат на прямой";

}else{

if(ans\_second + ans\_first == 0)

s="точки лежат по разные стороны от прямой";

else{

s="одна точка лежит на прямой, а вторая";

if(ans\_second==-1 || ans\_first==-1)

s+=" справа";

else

s+=" слева";

s+=" от прямой";

}

}

((Point\_)point\_first).setColor(cl\_first);

point\_first.setLocation(((Point\_)point\_first).x, ((Point\_)point\_first).y);

add(point\_first);

((Point\_)point\_second).setColor(cl\_second);

point\_second.setLocation(((Point\_)point\_second).x, ((Point\_)point\_second).y);

add(point\_second);

repaint();

}

public void paint (Graphics g) {

g.drawString(s, 0, 280);

g.setColor(Line\_.clr);

g.drawLine(Line\_.x1, Line\_.y1, Line\_.x2, Line\_.y2);

}

}