НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

КАФЕДРА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ

Лабораторна робота №1

з дісципліни **«**Комп’ютерна графіка**»**

Виконав:

студент 2 курсу

ФІОТ гр. ІО-21

Кузьменко Володимир

Перевірив:

Саверченко В. Г.

Київ 2014 р.

Завдання:

Відтворити рух довільно обраного об’єкту за заданою траєкторією.

Траєкторія: по квадрату.

**import** javax.swing.JFrame;

**public** **class** Lab4 {

**public** **static** **void** main(String[] args) {

AnimationKG frame = **new** AnimationKG();

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

frame.setVisible(**true**);

}

}import java.awt.Color;

import java.awt.Dimension;

import java.awt.Graphics;

import java.awt.Graphics2D;

import java.awt.geom.Rectangle2D;

import javax.swing.JPanel;

public class AnimationPane extends JPanel implements Runnable {

/\*\*

\*

\*/

private static final long serialVersionUID = 1L;

private int leftX = 120;

private int topY = 120;

private int weight = 200;

private int height = 200;

private int radius = 20;

Rectangle2D rect = new Rectangle2D.Double(leftX, topY - radius, radius,

radius);

Graphics2D g2;

Rectangle2D rectWhite = new Rectangle2D.Double(leftX - radius - 1, topY

- radius - radius - 1, radius + 2, radius + 2);

public AnimationPane() {

setPreferredSize(new Dimension(600, 600));

}

public void paintComponent(Graphics g) {

Graphics2D g2 = (Graphics2D) g;

Rectangle2D rectangle2d = new Rectangle2D.Double(leftX, topY, weight,

height);

g2.draw(rectangle2d);

g2.draw(rect);

this.g2 = g2;

g2.setPaint(getBackground());

g2.fill(rectWhite);

g2.setPaint(Color.black);

g2.fill(rect);

g2.setColor(getBackground());

}

@Override

public void run() {

for (int j = 0; j < 4; j++) {

if (j == 0) {

for (int i = 0; i < 200 + (radius + 2) / 8; i++) {

rect.setRect(leftX + i + 2, topY - radius, radius, radius);

try {

Thread.sleep(5);

} catch (InterruptedException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

rectWhite.setRect(leftX - radius + i + 2, topY - radius,

radius + 2, radius);

repaint();

}

}

if (j == 1) {

for (int i = 0; i < 200 + 3; i++) {

rect.setRect(leftX + height + 2, topY + i,

radius - 1, radius - 1);

rectWhite.setRect(leftX + height + 2,

topY - radius + i, radius + 2, radius);

repaint();

try {

Thread.sleep(5);

} catch (InterruptedException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

}

if (j == 2) {

for (int i = 0; i < 200 + (radius + 7) ; i++) {

rect.setRect(leftX + height-i+3, topY + height+2, radius, radius);

try {

Thread.sleep(5);

} catch (InterruptedException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

rectWhite.setRect(leftX + radius + height-i+3, topY + height+2,

radius, radius+2);

repaint();

}

}

if (j==3){

for (int i = 0; i < 200 + 3; i++) {

rect.setRect(leftX - radius-2, topY +height-+ radius- i,

radius, radius );

rectWhite.setRect(leftX - radius-3,

topY + height - i+2, radius+2, radius+2);

repaint();

try {

Thread.sleep(5);

} catch (InterruptedException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

}

}

}

}

**import** java.util.concurrent.Executor;

**public** **class** AnimationKG **extends** JFrame {

/\*\*

\* Create the frame.

\*/

**public** AnimationKG() {

setTitle("Lab4 KG");

AnimationPane pane = **new** AnimationPane();

add(pane);

Executor executor = Executors.*newSingleThreadExecutor*();

executor.execute(pane);

pack();

}

}