МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ УКРАЇНИ «КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО Факультет прикладної математики
Кафедра програмного забезпечення комп'ютерних систем

ЗВІТ

з лабораторної роботи № 2

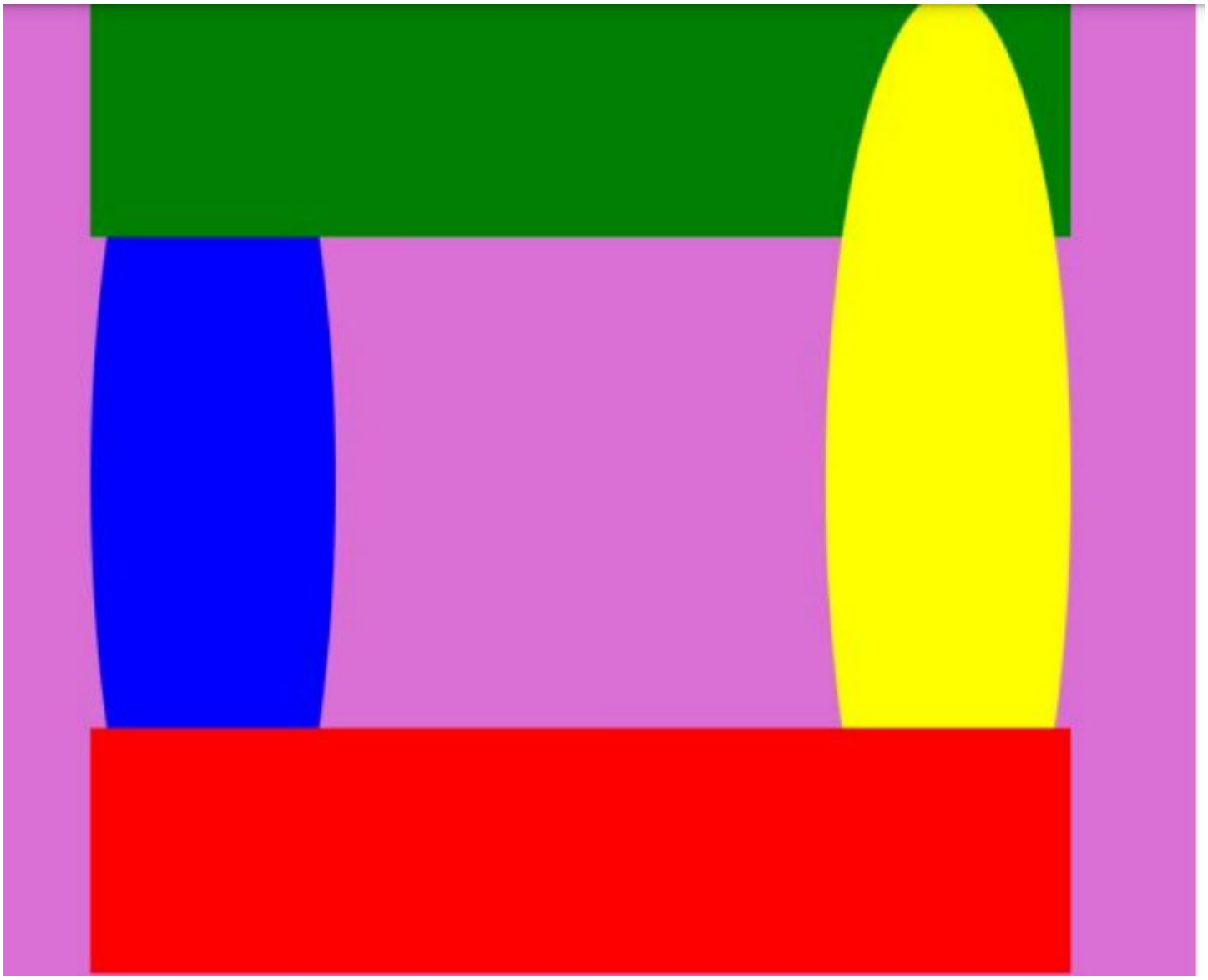
Варіант 1

Виконав:
студент 3-го курсу,
групи КП-83,
спеціальності 121 – Інженерія програмного забезпечення
Бойчук Владислав Андрійович

Київ – 2020

Тема: Побудова та анімація зображень за допомогою Java2D

Мета: Ознайомитися з можливостями побудови зображень та їх анімації у Java2D (Картинка з Лабораторної 1)Варіант 1:



Завдання:

За допомогою Java 2D намалювати картинку з лабораторної роботи №1 (за варіантом). Додатково виконати:

1. Хоча б 1 стандартний примітив, та хоча б 1 фігуру, побудовану по точкам (ламаною).
2. Хоча б 1 фігуру залити градієнтною фарбою за вибором (в цьому випадку колір може не співпадати з варіантом із лабораторної роботи № 1). 11
3. На достатній відстані від побудованого малюнку намалювати прямокутну рамку, всередині якої відбуватиметься анімація. Тип лінії рамки задано за варіантом. 4. Виконати анімацію малюнку, за варіантом. При цьому рамка повинна залишатися статичною. Взаємодія з рамкою не обов’язкова, якщо не передбачено варіантом.

11	3, 9	JOIN_MITER
12	1, 8	JOIN_ROUND
13	2, 9	JOIN_BEVEL
14	1, 6	JOIN_MITER
15	3, 6	JOIN_ROUND
16	4, 10	JOIN_BEVEL
17	5, 9	JOIN_MITER
18	5, 10	JOIN_ROUND
19	2, 10	JOIN_BEVEL
20	6, 10	JOIN_MITER
21	8, 9	JOIN_ROUND
22	3, 8	JOIN_BEVEL
23	6, 9	JOIN_MITER
24	8, 10	JOIN_ROUND
25	4, 9	JOIN_BEVEL
26	2, 5	JOIN_MITER
27	4, 6	JOIN_ROUND
28	2, 7	JOIN_BEVEL
29	4, 8	JOIN_MITER
30	2, 6	JOIN_ROUND

Типи анімації:

1. Рух по колу проти годинникової стрілки
2. Рух по колу за годинниковою стрілкою 12
3. Рух по квадрату проти годинникової стрілки
4. Рух по квадрату за годинниковою стрілкою
5. Обертання навколо центру малюнка за годинниковою стрілкою
6. Обертання навколо центру малюнка проти годинникової стрілки
7. Обертання навколо кута малюнка за годинниковою стрілкою
8. Обертання навколо кута малюнка проти годинникової стрілки
9. Зміна прозорості
10. Масштабуванняimport java.awt.*;

```
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.geom.GeneralPath;
import java.awt.geom.Line2D;

import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.Timer;

public class lab2 extends JPanel implements ActionListener {
double arrowPoligon[][] = {
{235+400, 174+300}, {330+400,55+300}, {424+400,174+300}, {377+400,174+300},{377+400,295+300},{283+400,294+300},{284+400,174+300}
```

```
};

public void crearRendering(Graphics2D g2d)
{

        g2d.setRenderingHint(RenderingHints.KEY_ANTIALIASING,
        RenderingHints.VALUE_ANTIALIAS_ON);

        g2d.setRenderingHint(RenderingHints.KEY_RENDERING,
        RenderingHints.VALUE_RENDER_QUALITY);
}
public void createField(Graphics2D g2d)
{
        g2d.setColor(new Color(255,255,255));
        g2d.fillRect(0, 0, 500, 350);
        g2d.setColor(new Color(200,200,200));
        g2d.fillRect(500, 0, 480, 350);
        g2d.setColor(new Color(130,130,130));
        g2d.fillRect(0, 350, 500, 350);
        g2d.setColor(new Color(255,130,130));
        g2d.fillRect(500, 350, 480, 350);
}



public GeneralPath createPoligon(double arr[][])
{
        GeneralPath poligon = new GeneralPath();
        poligon.moveTo(arr[0][0], arr[0][1]);

        for (int k = 1; k < arr.length; k++)
                poligon.lineTo(arr[k][0], arr[k][1]);
        poligon.closePath();
        return poligon;
}
public int[][] createCircle(){
        int[][] circleCoordinate = new int[628][2];
        for(double i = 0; i < 2 * Math.PI;i+=0.01){
                circleCoordinate[(int)(i*100)][0] = (int)(2 * 50*Math.cos(i));
                circleCoordinate[(int)(i*100)][1] = (int)(2 * 50*Math.sin(i));

                System.out.println((int)(i*100) + "=i"+"x" + "="+ circleCoordinate[(int)i*100][0]  + "y" + "="+ circleCoordinate[(int)i*100][1]);

        }
        System.out.println("x" + "="+ circleCoordinate[25][0]  + "y" + "="+ circleCoordinate[25][1]);
        return circleCoordinate;
```

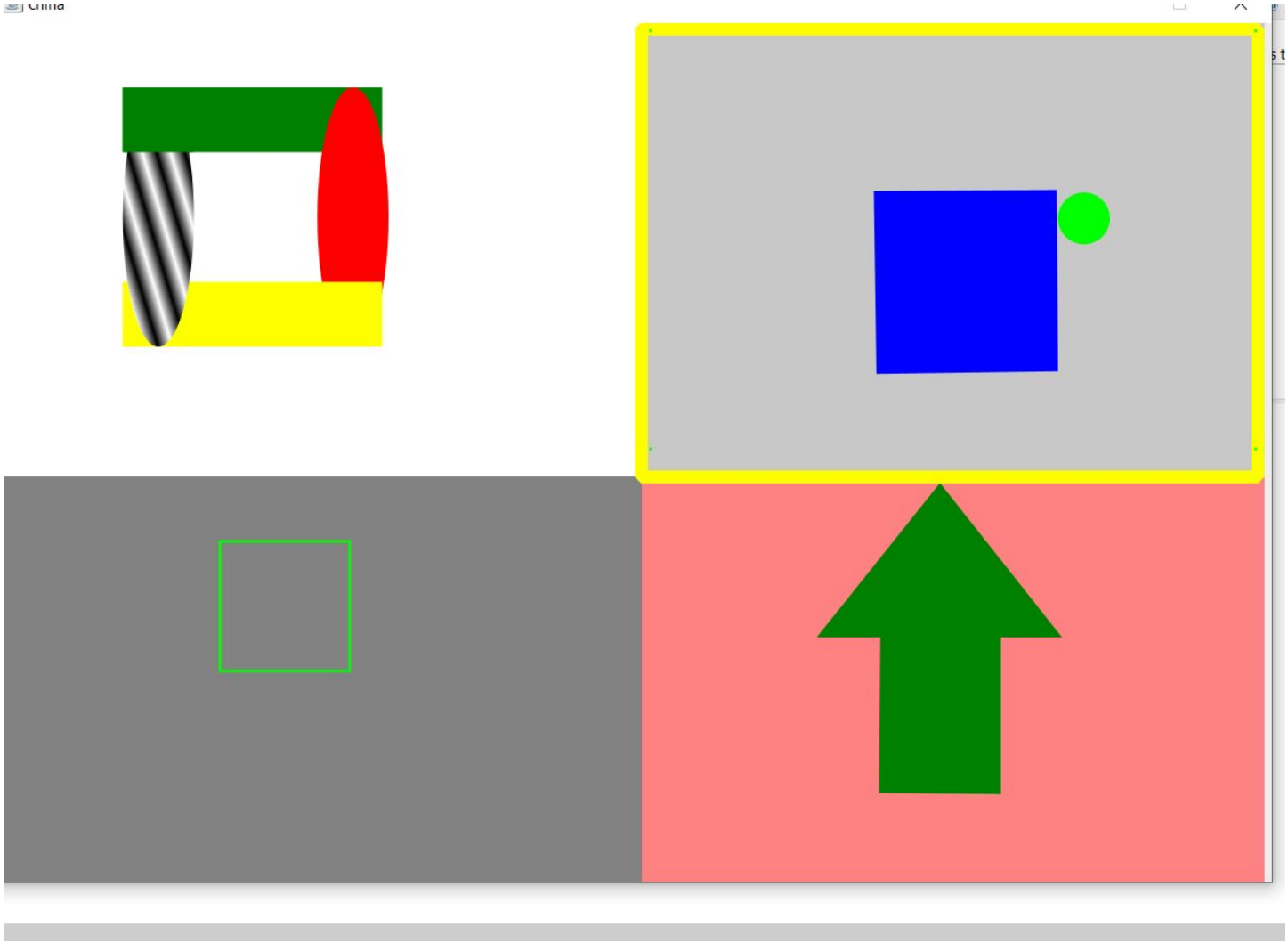
```
}

Timer timer;
int diamentrOfAnimanitionCircle = 40;



int minX =750;
int minY =200;
int maxX =918+50;
int maxY =289+56;
int counter = 627;
int [][] circleCoordinate = createCircle();

int xAnimation= minX +  circleCoordinate[counter][0];
int yAnimation= minY +circleCoordinate[counter][1];

int rectCounter = 0;
int rightBottomPoint  = rectCounter+0;
int rightTopPoint  = (rectCounter+156)%627;
int leftTopPoint  = (rectCounter+314)%627;
int leftBottomPoint  = (rectCounter+471)%627;
int [] rightBottomCorner = circleCoordinate[rightBottomPoint];
int [] rightTopCorner = circleCoordinate[rightTopPoint];
int [] leftTopCorner = circleCoordinate[leftTopPoint];
int [] leftBottomCorner = circleCoordinate[leftBottomPoint];

private static int maxWidth;
private static int maxHeight;


  public lab2() {

                timer = new Timer(20, this);
                timer.start();
                }



  public void paint(Graphics g) {
                //0-2ST  TASK START
                super.paint(g);
                Graphics2D g2d = (Graphics2D)g;
                createField(g2d);
                crearRendering(g2d);
               g2d.setColor(new Color(0,128,0));
```

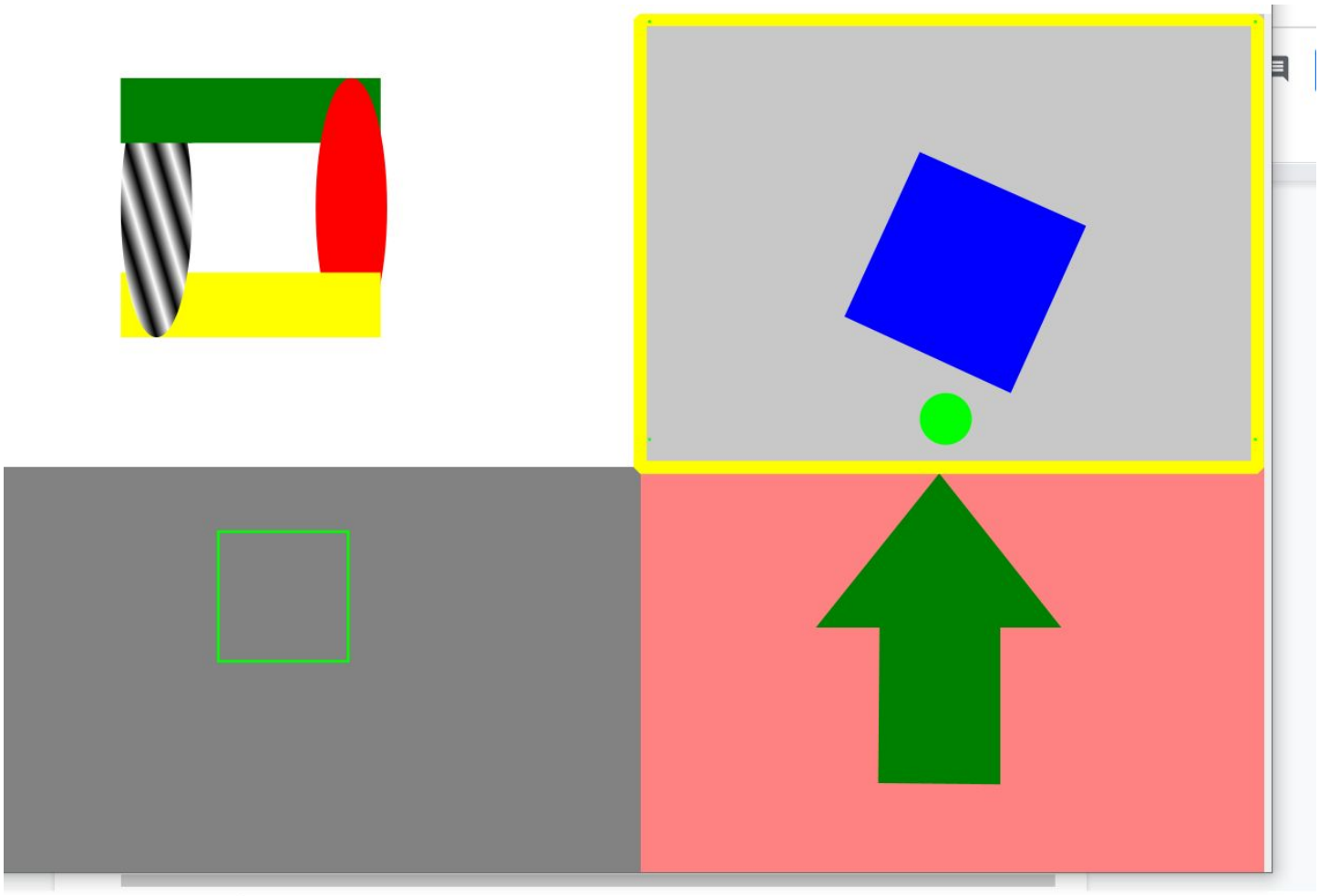
```
g2d.fillRect(100,50,200,50);
g2d.setColor(new Color(255,0,0));
g2d.fillOval(250,50,55,200);
g2d.setColor(new Color(255,255,0));
g2d.fillRect(100,200,200,50);
GradientPaint gp = new GradientPaint(10, 5,
Color.BLACK, 20, 2, Color.WHITE, true);
g2d.setPaint(gp);
g2d.fillOval(100,50,55,200);
g2d.setColor(new Color(0,128,0));
g2d.fillRect(100,50,55,50);
//0-2st task completed
//1st task completed
GeneralPath arrow =createPoligon(arrowPoligon);
g2d.fill(arrow);
//1st task completed
GradientPaint blueToBlack = new GradientPaint(330+400,55+300, Color.BLUE,
283+400,234+300, Color.white);
g2d.setPaint(blueToBlack);
g2d.setColor(new Color(255,255,0));
BasicStroke bs1 = new BasicStroke(10, BasicStroke.CAP_ROUND,
BasicStroke.JOIN_BEVEL);
g2d.setStroke(bs1);




g2d.drawRect(500, 5, 475, 345);
g2d.setStroke(new BasicStroke(2f));
g2d.setColor(Color.green);
g2d.draw(new Line2D.Double(507, 6, 507, 6));
g2d.setStroke(new BasicStroke(2f));
g2d.draw(new Line2D.Double(973, 6, 973, 6));
g2d.setStroke(new BasicStroke(2f));
g2d.draw(new Line2D.Double(973, 329, 973, 329));
g2d.draw(new Line2D.Double(507, 329, 507, 329));
g2d.drawRect(175, 400, 100, 100);

g2d.setComposite(AlphaComposite.getInstance(AlphaComposite.SRC_OVER));

g2d.fillOval(xAnimation, yAnimation,  (int)(diamentrOfAnimanitionCircle), (int)(diamentrOfAnimanitionCircle));
GeneralPath ownRectangle = new GeneralPath();
g2d.setColor(Color.BLUE);
ownRectangle.moveTo(750+rightBottomCorner[0], 200+rightBottomCorner[1]);
```

```
ownRectangle.lineTo(750+rightTopCorner[0], 200+rightTopCorner[1]);
ownRectangle.lineTo(750+leftTopCorner[0], 200+leftTopCorner[1]);
ownRectangle.lineTo(750+leftBottomCorner[0],200+ leftBottomCorner[1]);
ownRectangle.closePath();
g2d.fill(ownRectangle);
}

public static void main(String[] args) {
JFrame frame = new JFrame("chiha");
frame.add(new lab2());
frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
frame.setSize(1000, 700);
frame.setResizable(false);
frame.setLocationRelativeTo(null);
frame.setVisible(true);

Dimension size = frame.getSize();
Insets insets = frame.getInsets();
maxWidth = size.width - insets.left - insets.right - 1;
maxHeight = size.height - insets.top - insets.bottom - 1;
}

public void actionPerformed(ActionEvent e) {
counter--;
rectCounter++;
// System.out.println(counter);
if(counter == 0) {
counter=627;
}
if(rectCounter == 627) {
rectCounter=0;
}

xAnimation= minX +  circleCoordinate[counter][0];
yAnimation= minY +circleCoordinate[counter][1];

rightBottomPoint  = rectCounter+0;
rightTopPoint  = (rectCounter+156)%627;
leftTopPoint  = (rectCounter+314)%627;
leftBottomPoint  = (rectCounter+471)%627;
rightBottomCorner = circleCoordinate[rightBottomPoint];
rightTopCorner = circleCoordinate[rightTopPoint];
leftTopCorner = circleCoordinate[leftTopPoint];
leftBottomCorner = circleCoordinate[leftBottomPoint];
```

```
                // System.out.println(xAnimation);
                // System.out.println(yAnimation);


                  repaint();


        }
```

Приклад роботи програми :





Лаб роб 2.pdf

Висновок:Ознайомився з можливостями побудови зображень та їх анімації у Java2D.