

Національний технічний університет України  
«Київський політехнічний інститут»  
Факультет інформатики і обчислювальної техніки  
Кафедра обчислювальної техніки

Лабораторна робота №4  
З комп'ютерної графіки  
Бригада №3

*Виконали:*  
Студенти групи ІО-32  
Змеул Є.С.  
Попенко Р.Л.  
*Перевірів:*  
Саверченко В.Г.

Київ - 2015 р.

### 1. Завдання:

Відтворити рух довільно обраного об'єкту за заданою траєкторією по варіанту.

Варіант №3: по верхній частині кола за годинниковою стрілкою.

### 2. Короткі теоретичні відомості:

Параметричне рівняння кола з центром у точці  $(x_0, y_0)$ :

$$\begin{cases} x = R * \cos t + x_0, \\ y = R * \sin t + y_0. \end{cases}$$

### 3. Лістинг програми:

```
package labs.lab4;

import java.awt.Color;
import java.awt.Graphics;
import java.awt.Graphics2D;
import java.awt.event.KeyAdapter;
import java.awt.event.KeyEvent;
import java.awt.image.BufferedImage;
import javax.swing.JComponent;
import javax.swing.JFrame;

public class DrawBoard extends JFrame {
    protected static final int WIDTH = 800;
    protected static final int HEIGHT = 600;
    private static DrawPanel drPanel;

    public DrawBoard() {
        this.addNotify();
        this.setSize(this.getInsets().left + this.getInsets().right + WIDTH,
            this.getInsets().top + this.getInsets().bottom + HEIGHT);
        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        drPanel = new DrawPanel(this);
        this.add(drPanel);
        this.addKeyListener(new AL());
        this.setVisible(true);
    }

    public class DrawPanel extends JComponent implements Runnable {
        private double timer;
        private double timerDelta = 1;
        private int radius = 200;
        private int rad = 100;
        private int x, y;

        private int width, height;
        private BufferedImage dbImage;
        private Graphics2D dbGraphics;

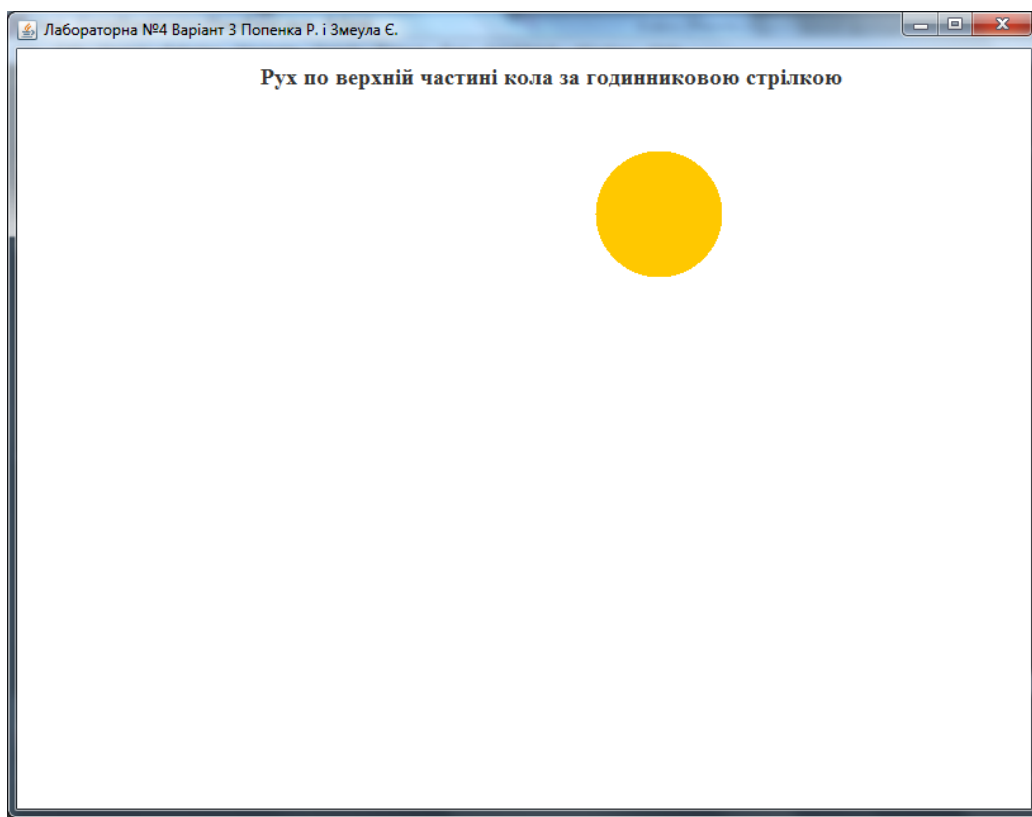
        public DrawPanel(DrawBoard drb) {
            super();
            width = DrawBoard.WIDTH;
            height = DrawBoard.HEIGHT;
            this.setSize(width, height);
            dbImage = new BufferedImage(width, height, BufferedImage.TYPE_INT_RGB);
            dbGraphics = dbImage.createGraphics();
            dbGraphics.setBackground(Color.WHITE);
            dbGraphics.setColor(Color.RED);
            timer = 0;
            new Thread(this).start();
        }
    }
}
```

```

@Override
protected void paintComponent(Graphics g) {
    Graphics2D g2 = (Graphics2D) g;
    dbGraphics.clearRect(0, 0, width, height);
    x = (int) -Math.round(Math.cos(timer*Math.PI/180)*radius);
    y = (int) -Math.round(Math.sin(timer*Math.PI/180)*radius);
    if((timer <= 180)
        dbGraphics.fillOval(x + width/2 - rad/2, y + height/2 - rad/2, rad, rad);
    else
        timer = 0.0;
    g2.drawImage(dblImage, 0, 0, this);
}
@Override
public void run() {
    while (true) {
        repaint();
        timer += timerDelta;
        try {
            Thread.sleep(10);
        } catch (InterruptedException ex) {
            System.out.println("InterruptedException!\n" + ex.getMessage());
        }
    }
}
}
}
}

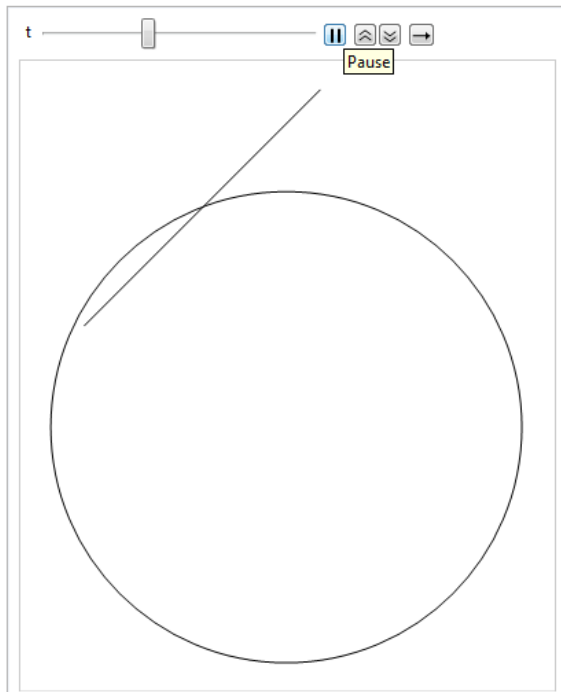
```

## Виконання



Також було розроблено у системі wolfram mathematica

```
tmin = -Pi/2;  
tmax = Pi/2;  
dt = 0.02;  
Animate[Graphics[{ {Circle[],  
  Line [{{Sin[t] - 0.5, Cos[t] - 0.5}, {Sin[t] + 0.5,  
    Cos[t] + 0.5}}]}], {t, tmin, tmax}, AnimationRunning -> False]
```



#### 4. Аналіз результатів (висновки):

Для виконання даної лабораторної роботи було використане параметричне рівняння кола з центром у заданій точці. Для того, щоб фігура (круг) переміщалася лише по верхній частині кола, її параметр, набуваючи значення у 180 градусів, обнуляється.