

МIНIСТЕРСТВО ОСВIТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ”

Факультет прикладної математики

Кафедра програмного забезпечення комп’ютерних систем

**Лабораторна робота № 2**

з дисципліни “ Математичні та алгоритмічні основи компютерної графіки”

тема “ Побудова та анімація зображень за допомогою Java2D”

|  |  |  |
| --- | --- | --- |
| Виконала  студентка III курсу  групи КП-52  Левошко Катерина Василівна  варіант № 7  Сумарний бал: |  | Перевірила  “\_\_\_\_” “\_\_\_\_\_\_\_\_\_\_\_\_” 20\_\_\_ р.  викладач  Шкурат Оксана Сергіївна |

|  |  |
| --- | --- |
|  |  |

Київ 2018

**Мета:** Ознайомитися з можливостями побудови зображень та їх анімації у Java2D

**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).
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.

**Варіант 7:**

Тип анімації: Рух по квадрату проти годинникової стрілки та Обертання навколо центру малюнка за годинниковою стрілкою

**Текст коду програм**

|  |
| --- |
| Main.java |
| public class Main extends JPanel implements ActionListener{  private static int maxWidth;  private static int maxHeight;  private double angle = 0;  private double tx = -205;  private double ty = -140;  Timer timer;  public Main(){  timer = new Timer(10, this);  timer.start();  }  public void paint(Graphics g){  Graphics2D g2d = (Graphics2D)g;  RenderingHints rh = new RenderingHints(RenderingHints.KEY\_ANTIALIASING,  RenderingHints.VALUE\_ANTIALIAS\_ON);  rh.put(RenderingHints.KEY\_RENDERING,RenderingHints.VALUE\_RENDER\_QUALITY);  g2d.setRenderingHints(rh);  setAnimation(g);  }  public void setAnimation(Graphics g){  Graphics2D g2d = (Graphics2D)g;  g2d.setBackground(new Color(59,135,143));  g2d.clearRect(0,0,maxWidth,maxHeight);  BasicStroke basicStroke = new BasicStroke(5,BasicStroke.CAP\_ROUND,BasicStroke.JOIN\_BEVEL);  g2d.setStroke(basicStroke);  g2d.drawRoundRect(5,5,780,750,0,0);  double points [][] = {  {280,295}, {355,220},  {530,295}, {455,333},  {480,395}, {330,408}  };  GeneralPath body = new GeneralPath();  body.moveTo(points[0][0],points[0][1]);  for(int i = 0; i<points.length; i++){  body.lineTo(points[i][0],points[i][1]);  }  body.closePath();  g2d.translate(tx,ty);  g2d.rotate(angle,body.getBounds2D().getCenterX(),body.getBounds2D().getCenterY());  g2d.setColor(Color.GRAY);  g2d.fill(body);  printImage(g2d);  BasicStroke reset = new BasicStroke();  g2d.setStroke(reset);  }  public void printImage(Graphics2D g2d){  GradientPaint gradient = new GradientPaint(455,333,Color.black,505,395,Color.white);  g2d.setPaint(gradient);  Polygon tail = new Polygon();  tail.addPoint(475,340);  tail.addPoint(525,330);  tail.addPoint(495,383);  g2d.fillPolygon(tail);  g2d.fillRect(320,278,8,8);  g2d.fillRect(315,328,8,8);  g2d.drawLine(282,295,455,333);  g2d.drawLine(265,220,305,270);  g2d.drawLine(245,375,305,345);  GradientPaint resetGradient = new GradientPaint(0,0,Color.black,0,0,Color.black);  g2d.setPaint(resetGradient);  }  @Override  public void actionPerformed(ActionEvent e) {  if(tx <= -200 && ty <= 250){  ty += 1;  } else if (tx < 250 && ty > 200 ){  tx +=1;  }else if(tx > 200 && ty >= -140){  ty -= 1;  } else if (tx > -200 && ty < -140 ){  tx -=1;  }  angle += 0.01;  repaint();  }  public static void main(String[] args) {  JFrame frame = new JFrame("Hello, Laba\_2");  frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);  frame.setSize(800,800);  frame.setLocationRelativeTo(null);  frame.setResizable(false);  frame.setVisible(true);  Dimension size = frame.getSize();  Insets insets = frame.getInsets();  frame.add(new Main());  frame.setVisible(true);  maxWidth = size.width - insets.left - insets.right - 1;  maxHeight = size.height - insets.top - insets.bottom - 1;  }  } |

**Висновки**

Виконавши дану лабораторну роботу я навчилась працювати з бібліотекою JavaFX та ознайомилась з можливостями побудови зображень та їх анімації у Java2D.