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 № 5**

**«СТВОРЕННЯ АНІМАЦІЙ У ЗАСТОСУНКАХ ANDROID»**

**Виконав:**

студент 3-го курсу, групи КП-93, спеціальності 121 – Інженерія програмного забезпечення

*Інюшев Артем Владиславович*

**Перевірив:**

к. т. н, старший викладач

*Хайдуров Владислав Володимирович*

Київ – 2021

**Завдання до лабораторної роботи**

1. Ознайомитись із усіма теоретичними відомостями до лабораторної роботи.

2. Переглянути усі практичні завдання при роботі з графічними об’єктами та

методами їх створення.

3. Створити проекти з усіма розглянутими у лабораторній роботі анімаціями,

продемонструвати їх роботу. Виконати відповідні скріншоти та додати їх

до звіту лабораторної роботи.

4. Створити мобільний застосунок, у якому квадрат рухається (заданий

довжиною сторони у пікселях у програмі) по колу (параметри кола

задаються розробником у програмі), причому центр квадрата лежить на

самому колі і квадрат не обертається відносно свого центра. Кольори

обрати на власний розсуд. У додатку передбачити зміну швидкості (кроку

пересування) квадрата по колу.

5. Створити програмний додаток, відтворює «килим» Серпинського:

http://hpc-education.ru/files/lectures/2011/ershov/ershov\_2011\_slides02.pdf.

Відтворення «килиму» можна виконувати як з анімацією, так і без неї (за

бажанням розробника). За додаткові бали можна реалізувати й інші (по 1

балу за кожен) клітинні автомати з даного джерела.

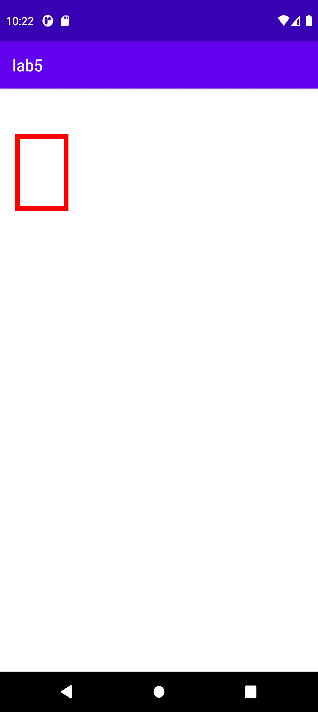
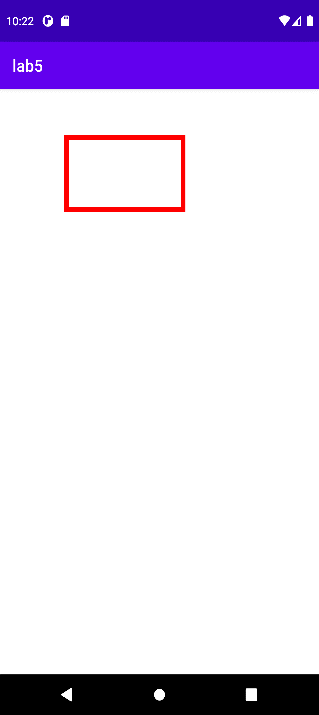
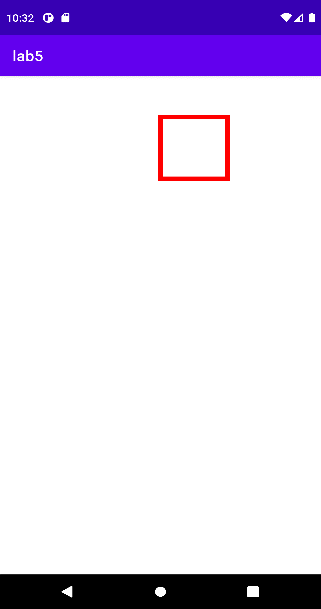
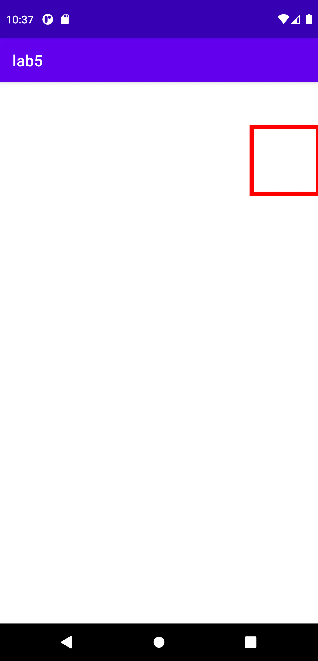
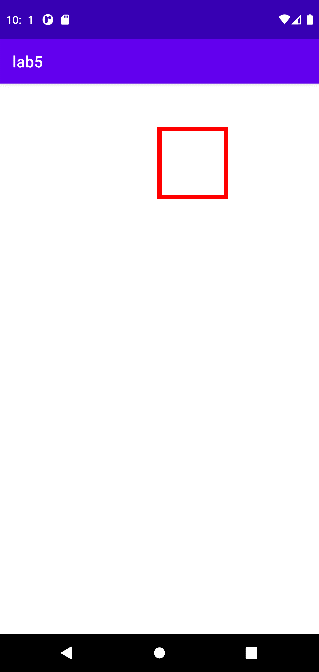
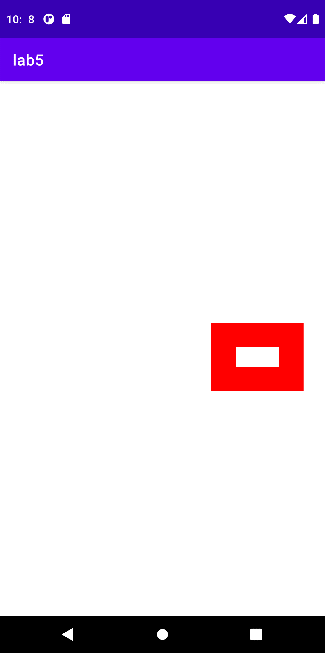
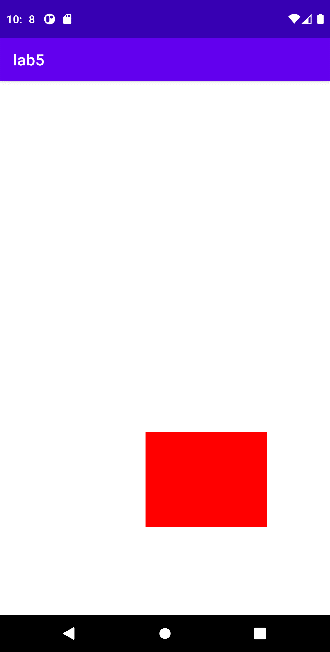
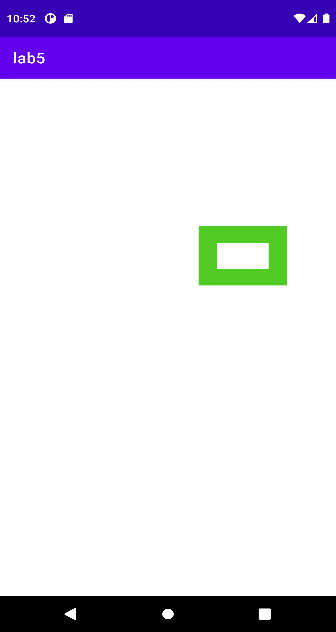
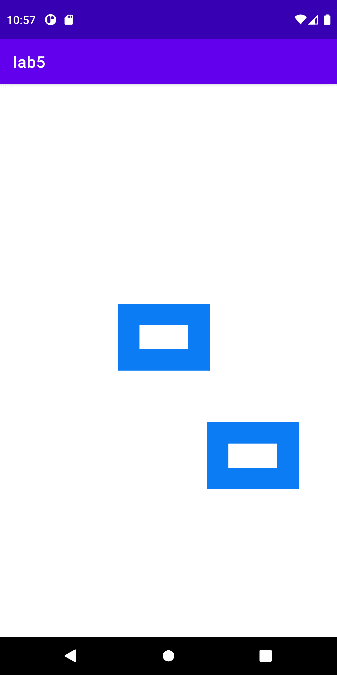
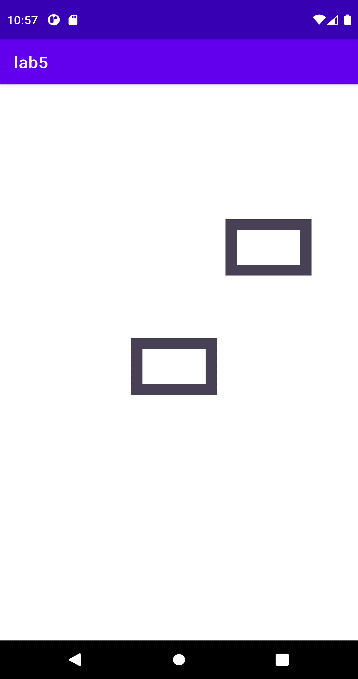
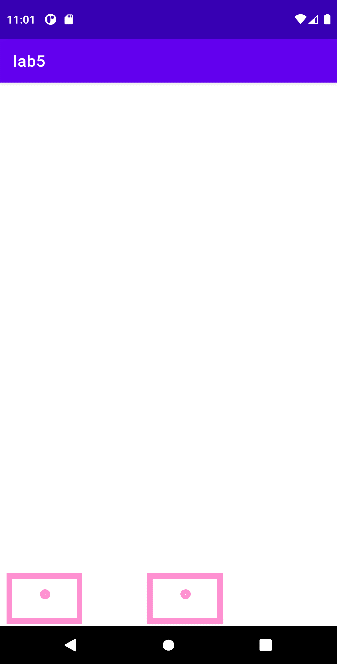
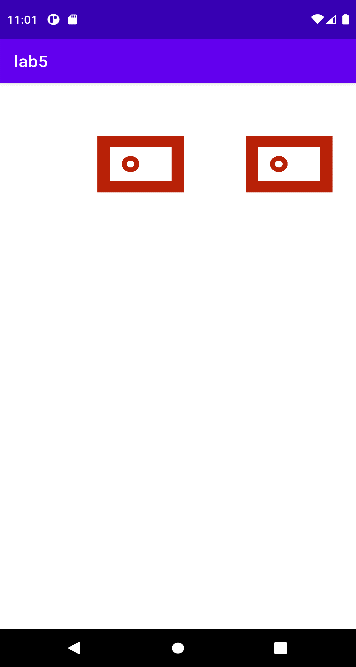
**Мета роботи**

Ознайомитись із основними принципами створення

анімацій у програмних застосунках під ОС Android та навчитись створювати

анімації власних графічних елементів дизайну.

**Перше завдання**

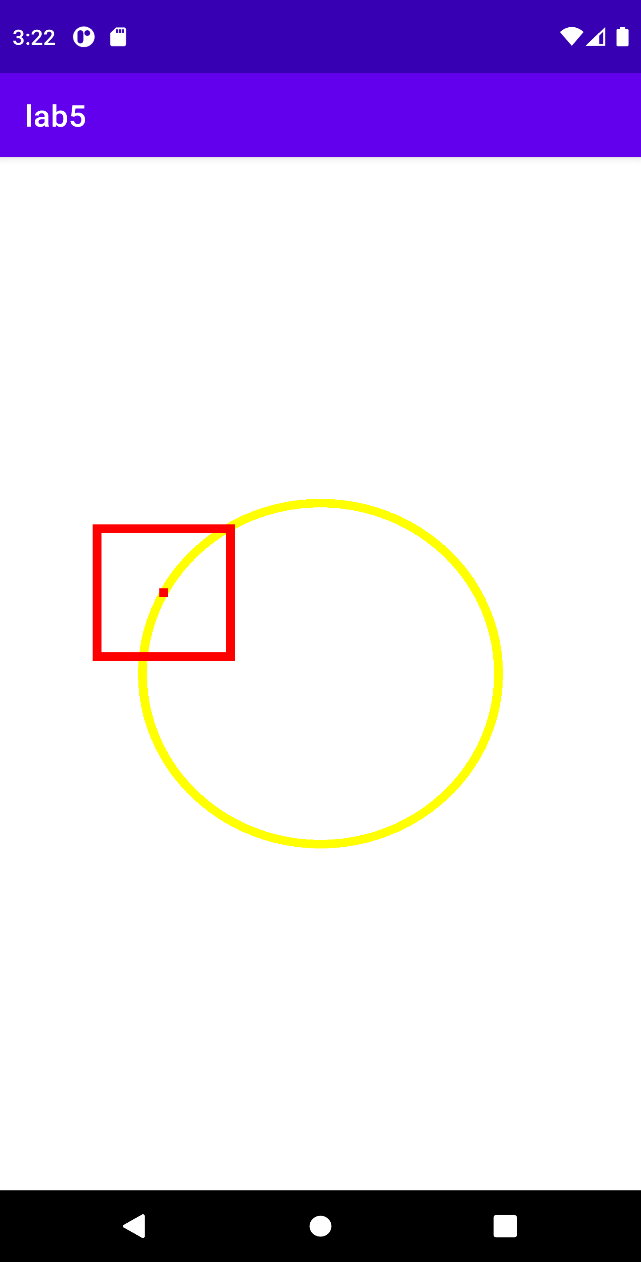
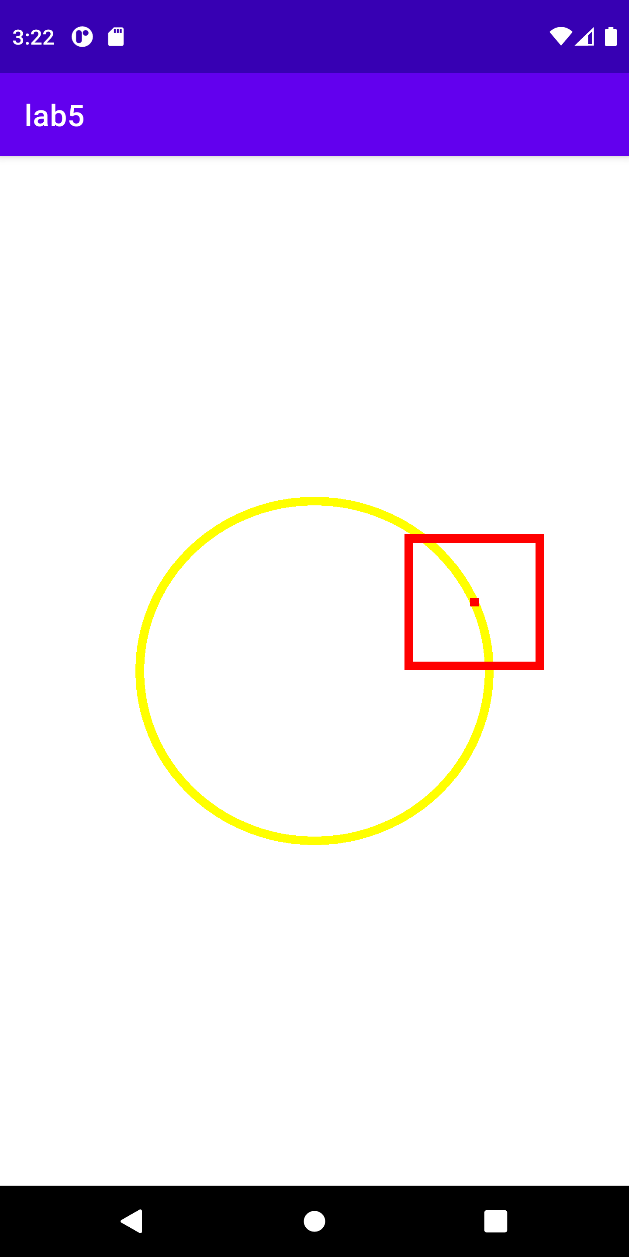
1.  
2. 
3. 
4. 
5.  
6.  
7.  
8.  

|  |
| --- |
| MyView.java |
| package com.example.lab5;  import android.content.Context;  import android.graphics.Canvas;  import android.graphics.Color;  import android.graphics.Paint;  import android.view.View;  public class MyView extends View {  int x = 0;  int y = 0;  int widthRect = 300;  int heightRect = 200;  int strokeWidth = 20;  int vx = 10;  int vy = 10;  int blue = 0;  int green = 0;  int red = 0;  int madX = 0;  int madY = 0;  public MyView(Context context) {  super(context);  }  void modifyRect() {  strokeWidth = (int)(Math.random()\*100);  blue = (int)(Math.random()\*255);  green = (int)(Math.random()\*255);  red = (int)(Math.random()\*255);  madX = (int)(Math.random()\*100 - 50);  madY = (int)(Math.random()\*100 - 50);  }  @Override  protected void onDraw(Canvas canvas) {  super.onDraw(canvas);  Paint paint = new Paint();  paint.setColor(Color.rgb(red, green, blue));  paint.setStyle(Paint.Style.STROKE);  paint.setStrokeWidth(strokeWidth);  x = x + vx;  y = y + vy;  if (x > canvas.getWidth() - widthRect) {  vx = vx \* -1;  modifyRect();  }  if (x < 0) {  vx = vx \* -1;  modifyRect();  }  if (y > canvas.getHeight() - heightRect) {  vy = vy \* -1;  modifyRect();  }  if (y < 0) {  vy = vy \* -1;  modifyRect();  }  canvas.drawRect(x,y,widthRect + x,heightRect+y,paint);  int xSecond = x + 600;  canvas.drawRect(xSecond,y,widthRect +  xSecond,heightRect+y,paint);  canvas.drawCircle(xSecond + widthRect/2 +  madX,y+heightRect/2 + madY,10,paint);  canvas.drawCircle(x + widthRect/2 + madX,y+heightRect/2 +  madY,10,paint);  invalidate();  }  } |

**Друге завдання**

|  |
| --- |
| MyView.java |
| package com.example.lab5;  import android.content.Context;  import android.graphics.Canvas;  import android.graphics.Color;  import android.graphics.Paint;  import android.view.View;  import java.util.Random;  public class MyView extends View {  private float circleRadius = 400;  private float step = -400;  private float vstep = 1;  private float[] speeds = new float[]{1, 2, 4, 5, 8};  private boolean leftSide = true;  public MyView(Context context) {  super(context);  }  private float calculateX2Coordinate(float x1, float y1, float y2, boolean left){  // (x1 - x2)^2 + (y1 - y2)^2 = length^2  double tmp = Math.pow(circleRadius, 2) - Math.pow((y1 - y2), 2);  double x2 = 0;  if(left){  x2 = x1 - Math.sqrt(tmp);  }  else{  x2 = x1 + Math.sqrt(tmp);  }  return (float) x2;  }  @Override  protected void onDraw(Canvas canvas) {  super.onDraw(canvas);  float circleMiddleX = canvas.getWidth() / 2;  float circleMiddleY = canvas.getHeight() / 2;  Paint paintCircle = new Paint();  paintCircle.setColor(Color.YELLOW);  paintCircle.setStyle(Paint.Style.STROKE);  paintCircle.setStrokeWidth(20);  canvas.drawCircle(circleMiddleX, circleMiddleY, circleRadius, paintCircle);  float squareMiddleY = circleMiddleY + step;  float squareMiddleX = calculateX2Coordinate(circleMiddleX, circleMiddleY, squareMiddleY, leftSide);  if(Math.abs(squareMiddleY - circleMiddleY) == circleRadius){  leftSide = !leftSide;  int rnd = new Random().nextInt(speeds.length);  vstep = speeds[rnd];  }  if(leftSide){  step -= vstep;  }  else {  step += vstep;  }  Paint paintMiddleSquarePoint = new Paint();  paintMiddleSquarePoint.setColor(Color.RED);  paintMiddleSquarePoint.setStyle(Paint.Style.STROKE);  paintMiddleSquarePoint.setStrokeWidth(20);  canvas.drawPoint(squareMiddleX, squareMiddleY, paintMiddleSquarePoint);  Paint paint = new Paint();  paint.setColor(Color.RED);  paint.setStyle(Paint.Style.STROKE);  paint.setStrokeWidth(20);  canvas.drawRect(squareMiddleX - 150,squareMiddleY - 150,  squareMiddleX + 150,squareMiddleY + 150, paint);  invalidate();  }  } |

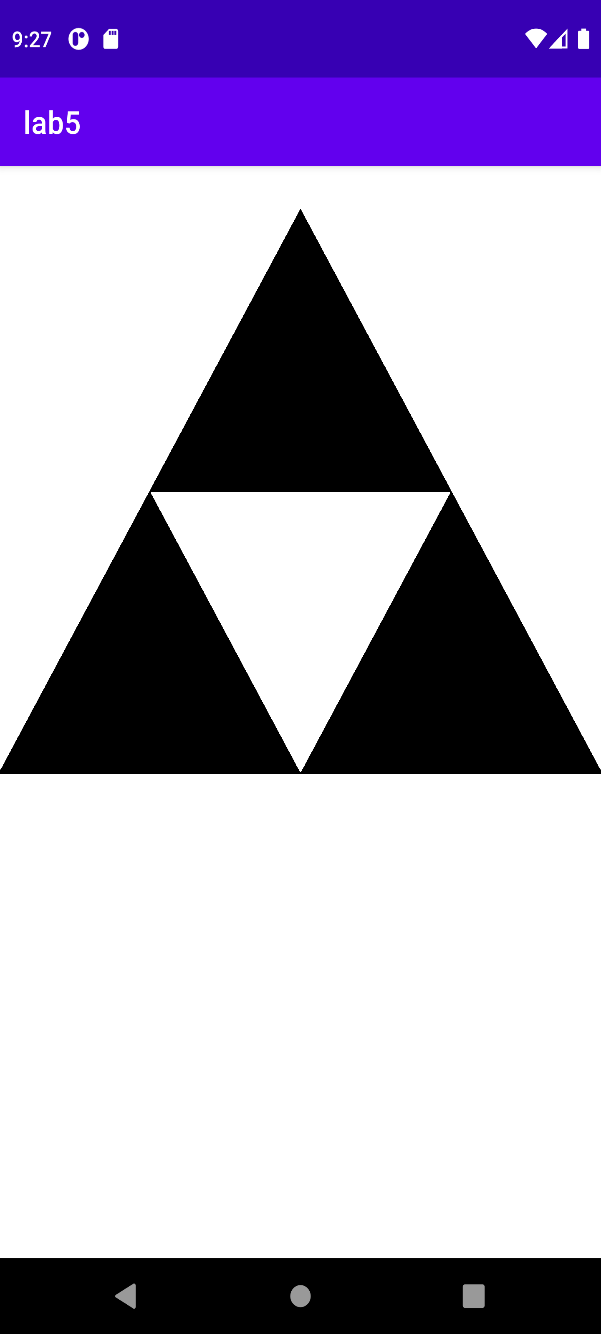
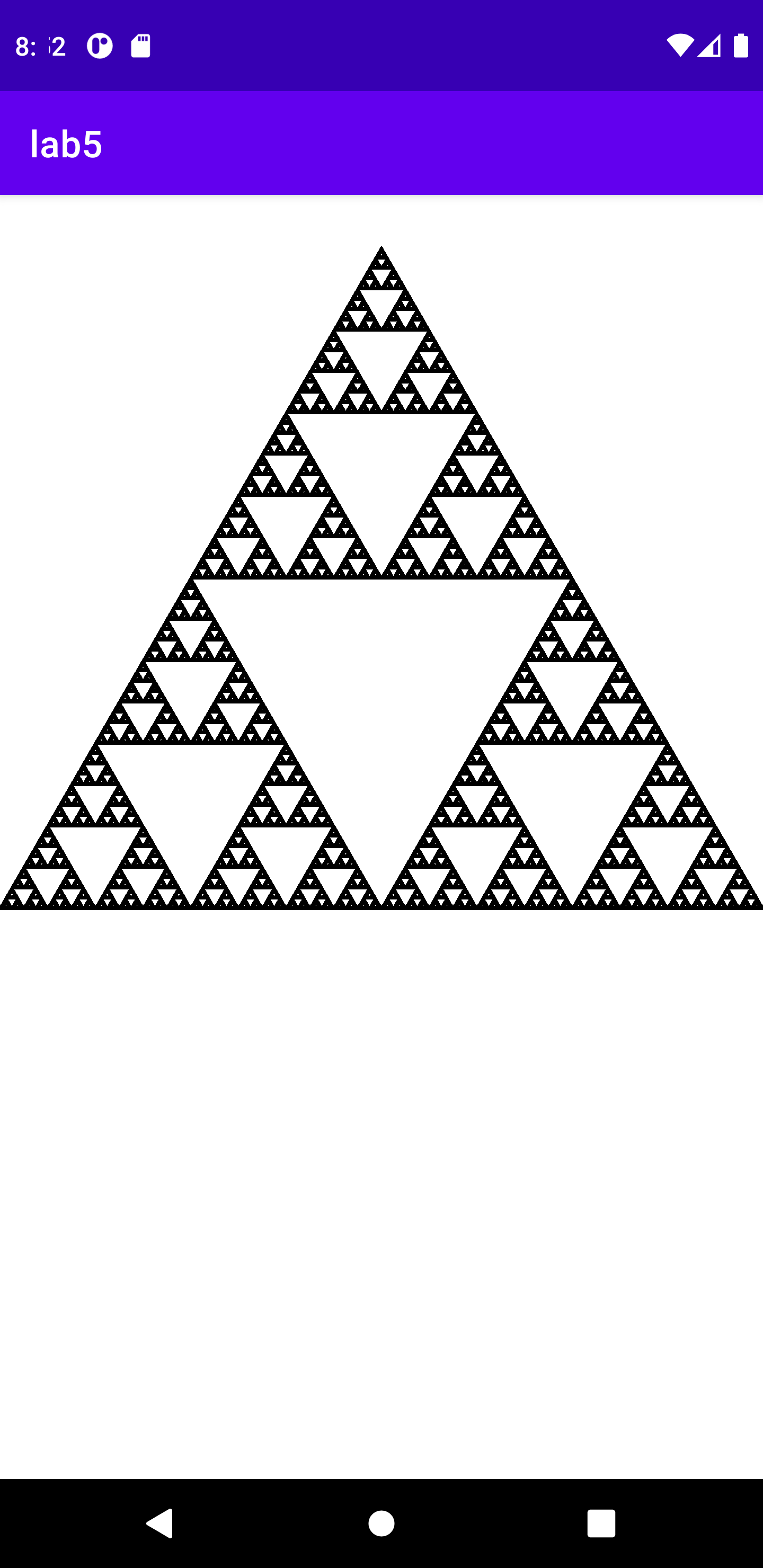
**Зображення роботи програми:**

** **

**Третє завдання**

|  |
| --- |
| MyView.java |
| package com.example.lab5;  import android.content.Context;  import android.graphics.Canvas;  import android.graphics.Color;  import android.graphics.Paint;  import android.graphics.Path;  import android.view.View;  class Point{  public float X;  public float Y;  public Point(float x, float y){  X = x;  Y = y;  }  }  public class MyView extends View {  private int iteration = 0;  public MyView(Context context) {  super(context);  }  private void SierpinskiFunc(Point[] points, Canvas canvas, int depth){  Paint paint = new Paint();  paint.setColor(Color.BLACK);  boolean finish = false;  if(depth >= iteration) {  paint.setStyle(Paint.Style.FILL);  finish = true;  }  else{  paint.setStyle(Paint.Style.STROKE);  paint.setStrokeWidth(5);  depth++;  }  Path path = new Path();  path.moveTo(points[0].X, points[0].Y);  path.lineTo(points[1].X, points[1].Y);  path.lineTo(points[2].X, points[2].Y);  path.lineTo(points[0].X, points[0].Y);  path.close();  canvas.drawPath(path, paint);  if(finish) return;  Point middleLeft = new Point((points[0].X + points[1].X) / 2,  (points[0].Y + points[1].Y) /2);  Point middleRight = new Point((points[0].X + points[2].X) / 2,  (points[0].Y + points[2].Y) /2);  Point middleBottom = new Point((points[1].X + points[2].X) / 2,  (points[1].Y + points[2].Y) /2);  SierpinskiFunc(new Point[]{ points[0], middleLeft, middleRight }, canvas, depth);  SierpinskiFunc(new Point[]{ middleLeft, points[1], middleBottom }, canvas, depth);  SierpinskiFunc(new Point[]{ middleRight, middleBottom, points[2] }, canvas, depth);  }  @Override  protected void onDraw(Canvas canvas) {  super.onDraw(canvas);  float mainX = canvas.getWidth() / 2;  float mainY = (float) (mainX \* Math.sqrt(3)) + 100;  Point[] points = new Point[]{  new Point(mainX, 100),  new Point(0, mainY),  new Point(mainX \* 2, mainY),  };  SierpinskiFunc(points, canvas, 0);  iteration++;  try {  Thread.sleep(2000);  } catch (InterruptedException e) {  e.printStackTrace();  }  if(iteration < 10){  invalidate();  }  }  } |

**Зображення роботи програми:**

** **

**Висновок**

У процесі виконання даної лабораторної нам вдалося:

1. Створити проекти з усіма розглянутими у лабораторній роботі анімаціями,

продемонструвати їх роботу. Виконати відповідні скріншоти та додати їх

до звіту лабораторної роботи.

2. Створити мобільний застосунок, у якому квадрат рухається (заданий

довжиною сторони у пікселях у програмі) по колу (параметри кола

задаються розробником у програмі), причому центр квадрата лежить на

самому колі і квадрат не обертається відносно свого центра. Кольори

обрати на власний розсуд. У додатку передбачити зміну швидкості (кроку

пересування) квадрата по колу.

3. Створити програмний додаток, відтворює «килим» Серпинського:

http://hpc-education.ru/files/lectures/2011/ershov/ershov\_2011\_slides02.pdf.

Відтворення «килиму» можна виконувати як з анімацією, так і без неї (за

бажанням розробника). За додаткові бали можна реалізувати й інші (по 1

балу за кожен) клітинні автомати з даного джерела.