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-91, спеціальності 121 – Інженерія програмного забезпечення

*Маховой Олександр Вікторович*

**Перевірив:**

к. т. н, старший викладач

*Хайдуров Владислав Володимирович*

Київ – 2021

**ЗМІСТ**

[**ВСТУП**](#_g3m3w97lczix) **3**

[1. Постановка задачі](#_buh4naa4v55s) **4**

[2. Короткі теоретичні відомості](#_xfn3ibhalz3h) **5**

[3. Програмна реалізація задачі](#_5vx8958s0i4y) **6**

[**ВИСНОВКИ**](#_d4hicg2cja8p) **38**

[**СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ**](#_q75rbm6vnfle) **39**

# **ВСТУП**

Для розробки майже будь-якого мобільного додатку необхідно добре розуміти як зробити приємний інтерфейс, а для цього необхідно розуміти як працюють анімації в мобільних додатках. В даній лабораторній роботі була проведена робота з анімаціями.

# **1. Постановка задачі**

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.

# 

# **2. Короткі теоретичні відомості**

Пакет android.graphics має всі необхідні бібліотеки для роботи з двовимірною графікою. Існує кілька підходів для малювання графіки.

Для малювання простої графіки, яка не буде динамічно змінюватися під час роботи програми, зазвичай використовують клас, що успадковує від View і задіють метод onDraw().

Клас Color відповідає за колір. Кольори можна описувати чотирма числами в форматі ARGB, по одному для кожного каналу (Alpha, Red, Green, Blue).

У графіці важливу роль відіграє колір, який представлений класом Color. Клас Color містить кілька констант і методів для конвертації і вилучення колірного компонента.

Кольори можна описувати чотирма числами в форматі ARGB, по одному для кожного каналу (Alpha, Red, Green, Blue). Кожен з каналів є восьмибітних цілим числом і може приймати значення від 0 до 256. Зазвичай колір упаковують в 32-бітове ціле число. Варто зазначити, що використовувати цілі числа для квітів ефективніше, ніж екземпляри класу Color.

# **3. Програмна реалізація задачі**

Файли програми lab5\_1:

|  |
| --- |
| Main Activity.java: |
| package com.example.lab5\_1;  import androidx.appcompat.app.AppCompatActivity;  import android.graphics.Canvas;  import android.os.Bundle;  import android.view.MotionEvent;  import android.view.View;  public class MainActivity extends AppCompatActivity {  private View view;  @Override  protected void onCreate(Bundle savedInstanceState) {  super.onCreate(savedInstanceState);  view = new MyView9(this);  setContentView(view);  }  } |

|  |
| --- |
| MyView.java: |
| package com.example.lab5\_1;  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;  public MyView(Context context) {  super(context);  }  @Override  protected void onDraw(Canvas canvas) {  super.onDraw(canvas);  Paint paint = new Paint();  paint.setColor(Color.RED);  paint.setStyle(Paint.Style.STROKE);  paint.setStrokeWidth(20);  x++;  canvas.drawRect(x,200,300,500,paint);  invalidate();  }  } |

|  |
| --- |
| MyView2.java: |
| package com.example.lab5\_1;  import android.content.Context;  import android.graphics.Canvas;  import android.graphics.Color;  import android.graphics.Paint;  import android.view.View;  public class MyView2 extends View {  int x = 0;  int widthRect = 300;  int vx = 10;  public MyView2(Context context) {  super(context);  }  @Override  protected void onDraw(Canvas canvas) {  super.onDraw(canvas);  Paint paint = new Paint();  paint.setColor(Color.RED);  paint.setStyle(Paint.Style.STROKE);  paint.setStrokeWidth(20);  x = x + vx;  if (x > canvas.getWidth() - widthRect) vx = vx \* -1;  if (x < 0) vx = vx \* -1;  canvas.drawRect(x,200,widthRect + x,500,paint);  invalidate();  }  } |

|  |
| --- |
| MyView3.java: |
| package com.example.lab5\_1;  import android.content.Context;  import android.graphics.Canvas;  import android.graphics.Color;  import android.graphics.Paint;  import android.view.View;  public class MyView3 extends View {  int x = 0;  int y = 0;  int widthRect = 300;  int heightRect = 200;  int strokeWidth = 20;  int vx = 10;  int vy = 10;  public MyView3(Context context) {  super(context);  }  @Override  protected void onDraw(Canvas canvas) {  super.onDraw(canvas);  Paint paint = new Paint();  paint.setColor(Color.RED);  paint.setStyle(Paint.Style.STROKE);  paint.setStrokeWidth(strokeWidth);  x = x + vx;  y = y + vy;  if (x > canvas.getWidth() - widthRect) {  vx = vx \* -1;  strokeWidth+=30;  }  if (x < 0) {  vx = vx \* -1;  strokeWidth+=30;  }  if (y > canvas.getHeight() - heightRect) {  vy = vy \* -1;  strokeWidth+=30;  }  if (y < 0) {  vy = vy \* -1;  strokeWidth+=30;  }  canvas.drawRect(x,y,widthRect + x,heightRect+y,paint);  invalidate();  }  } |

|  |
| --- |
| MyView4.java: |
| package com.example.lab5\_1;  import android.content.Context;  import android.graphics.Canvas;  import android.graphics.Color;  import android.graphics.Paint;  import android.view.View;  public class MyView4 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;  public MyView4(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);  }  @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);  invalidate();  }  } |

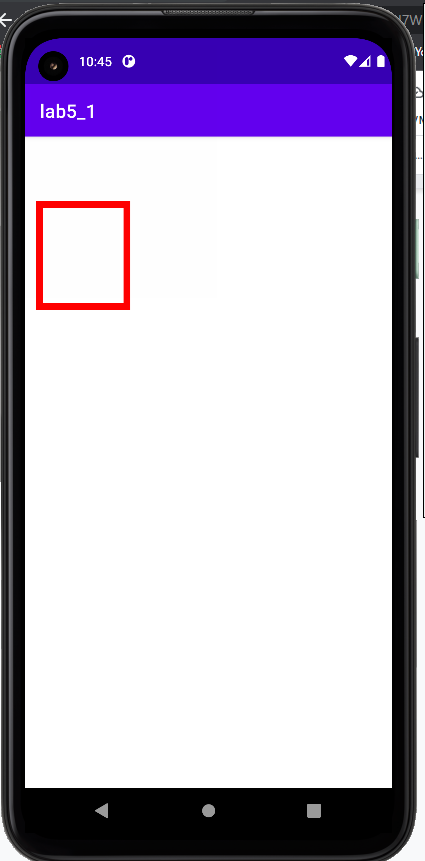
|  |
| --- |
| MyView5.java: |
| package com.example.lab5\_1;  import android.content.Context;  import android.graphics.Canvas;  import android.graphics.Color;  import android.graphics.Paint;  import android.view.View;  public class MyView5 extends View {  int x = 0;  int y = 0;  int widthRect = 300;  int heightRect = 200;  int vx = 10;  int vy = 10;  int x2 = 800;  int y2 = 500;  int widthRect2 = 300;  int heightRect2 = 200;  int vx2 = -10;  int vy2 = -10;  int strokeWidth = 20;  int blue = 0;  int green = 0;  int red = 0;  public MyView5(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);  }  @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);  x2 = x2 + vx2;  y2 = y2 + vy2;  if (x2 > canvas.getWidth() - widthRect) {  vx2 = vx2 \* -1;  modifyRect();  }  if (x2 < 0) {  vx2 = vx2 \* -1;  modifyRect();  }  if (y2 > canvas.getHeight() - heightRect) {  vy2 = vy2 \* -1;  modifyRect();  }  if (y2 < 0) {  vy2 = vy2 \* -1;  modifyRect();  }  canvas.drawRect(x2,y2,widthRect2 +  x2,heightRect2+y2,paint);  invalidate();  }  } |

|  |
| --- |
| MyView6.java: |
| package com.example.lab5\_1;  import android.content.Context;  import android.graphics.Canvas;  import android.graphics.Color;  import android.graphics.Paint;  import android.view.View;  public class MyView6 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 MyView6(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();  }  } |

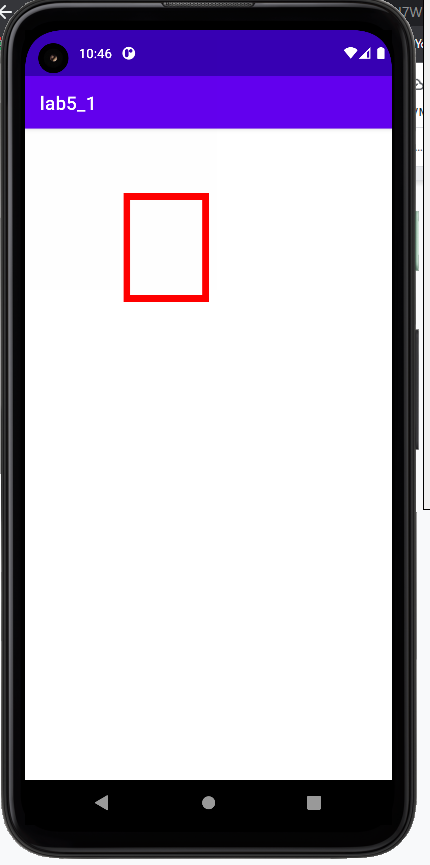
|  |
| --- |
| MyView7.java: |
| package com.example.lab5\_1;  import android.content.Context;  import android.graphics.Canvas;  import android.graphics.Color;  import android.graphics.Paint;  import android.view.GestureDetector;  import android.view.MotionEvent;  import android.view.View;  public class MyView7 extends View {  int x = 0;  int y = 0;  int side = 201;  int strokeWidth = 20;  int xEnd = 0;  int yEnd = 0;  int blue = 0;  int green = 255;  int red = 0;  int degrees = 0;  int step = 1;  int circleCenterX = 0;  int circleCenterY = 0;  int radius = 0;  public MyView7(Context context) {  super(context);  }  @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);  circleCenterX = canvas.getWidth()/2;  circleCenterY = canvas.getHeight()/2;  radius = (int) (canvas.getWidth() \* 0.4);  if (degrees == 360) {  degrees = 0;  }  degrees += step;  int positionX = (int) (circleCenterX + radius \* Math.cos(Math.toRadians(degrees)));  int positionY = (int) (circleCenterY + radius \* Math.sin(Math.toRadians(degrees)));  x = positionX - side/2;  y = positionY - side/2;  xEnd = positionX + side/2;  yEnd = positionY + side/2;  canvas.drawRect(x,y,xEnd,yEnd,paint);  invalidate();  }  public void increaseStep() {  if (step < 8) {  step++;  }  }  public void reduceStep() {  if (step > 0) {  step--;  }  }  } |

|  |
| --- |
| MyView8: |
| package com.example.lab5\_1;  import android.content.Context;  import android.graphics.Canvas;  import android.graphics.Color;  import android.graphics.Paint;  import android.view.GestureDetector;  import android.view.MotionEvent;  import android.view.View;  public class MyView8 extends View {  int p1x = 0;  int p1y = 0;  int p2x = 0;  int p2y = 0;  int p3x = 0;  int p3y = 0;  int red = 0;  int green = 255;  int blue = 0;  int strokeWidth = 10;  Paint paint;  double i = 1;  public MyView8(Context context) {  super(context);  }  @Override  protected void onDraw(Canvas canvas) {  super.onDraw(canvas);  paint = new Paint();  paint.setColor(Color.rgb(red, green, blue));  paint.setStyle(Paint.Style.STROKE);  paint.setStrokeWidth(strokeWidth);  p3x = canvas.getWidth();  int height = (int) (p3x \* Math.sin(Math.toRadians(60)));  p2y = (canvas.getHeight() - height)/2;  p1y = canvas.getHeight() - p2y;  p3y = p1y;  p2x = canvas.getWidth()/2;  canvas.drawLine(p1x,p1y,p2x,p2y,paint);  canvas.drawLine(p2x,p2y,p3x,p3y,paint);  canvas.drawLine(p1x,p1y,p3x,p3y,paint);  drawTriangle(canvas, p1x, p2x, p3x, p1y, p2y, p3y);  i+=0.1;  invalidate();  }  private void drawTriangle(Canvas canvas, int p1x, int p2x, int p3x, int p1y, int p2y, int p3y) {  int old\_p1x = p1x;  int old\_p2x = p2x;  int old\_p3x = p3x;  int old\_p1y = p1y;  int old\_p2y = p2y;  int old\_p3y = p3y;  p1x = (old\_p1x + old\_p2x) / 2;  p1y = (old\_p1y + old\_p2y) / 2;  p2x = (old\_p1x + old\_p3x) / 2;  p2y = (old\_p1y + old\_p3y) / 2;  p3x = (old\_p3x + old\_p2x) / 2;  p3y = (old\_p3y + old\_p2y) / 2;  canvas.drawLine(p1x,p1y,p2x,p2y,paint);  canvas.drawLine(p2x,p2y,p3x,p3y,paint);  canvas.drawLine(p1x,p1y,p3x,p3y,paint);  if (p2x - p1x > canvas.getWidth()/(2\*i)) {  drawTriangle(canvas, p1x, old\_p2x, p3x, p1y, old\_p2y, p3y);  drawTriangle(canvas, old\_p1x, p1x, p2x, old\_p1y, p1y, p2y);  drawTriangle(canvas, p2x, p3x, old\_p3x, p2y, p3y, old\_p3y);  }  }  } |

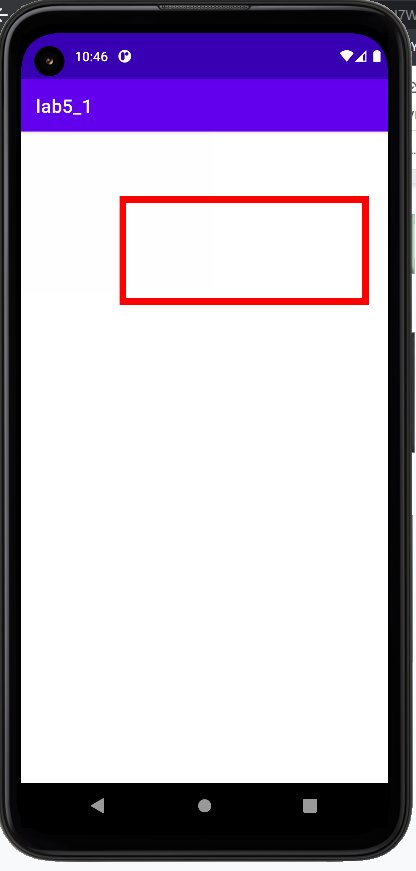
**4. Результати роботи програми**



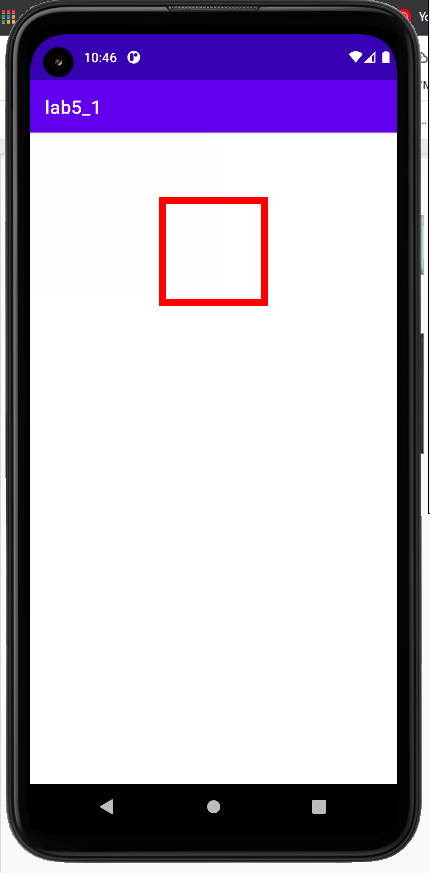
*Рис. 1.* Перша анімація, скріншот №1



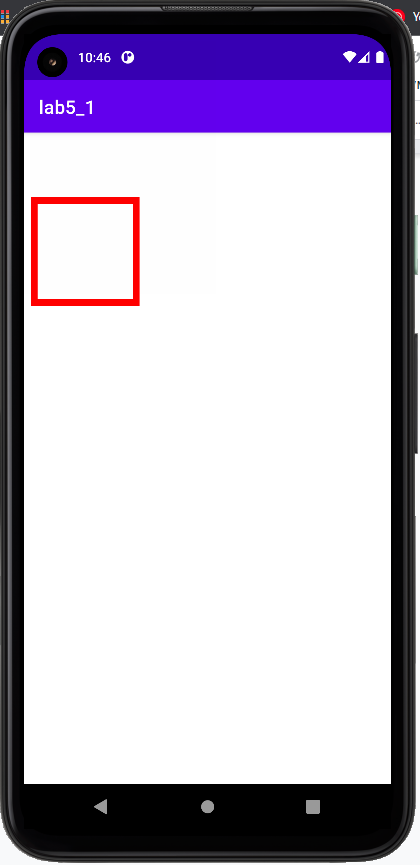
*Рис. 2.* Перша анімація, скріншот №2



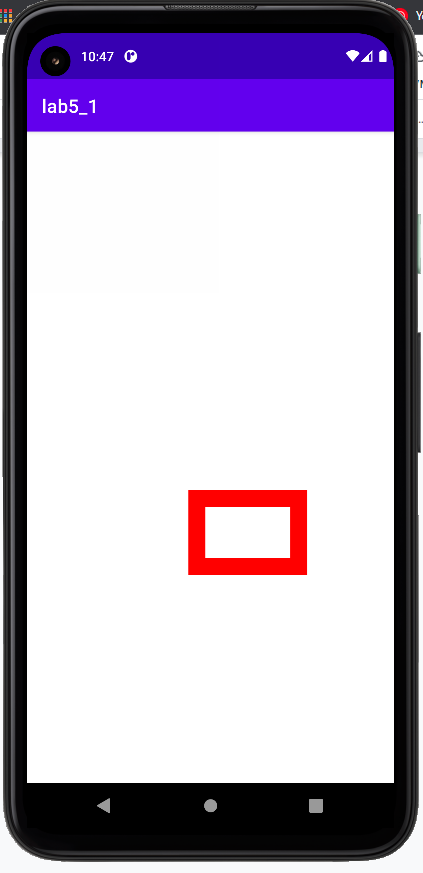
*Рис. 3.* Перша анімація, скріншот №3



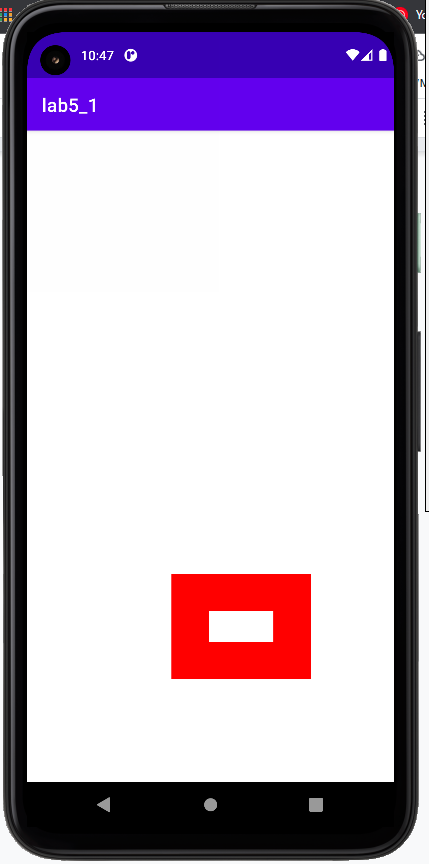
*Рис. 4.* Друга анімація, скріншот №1



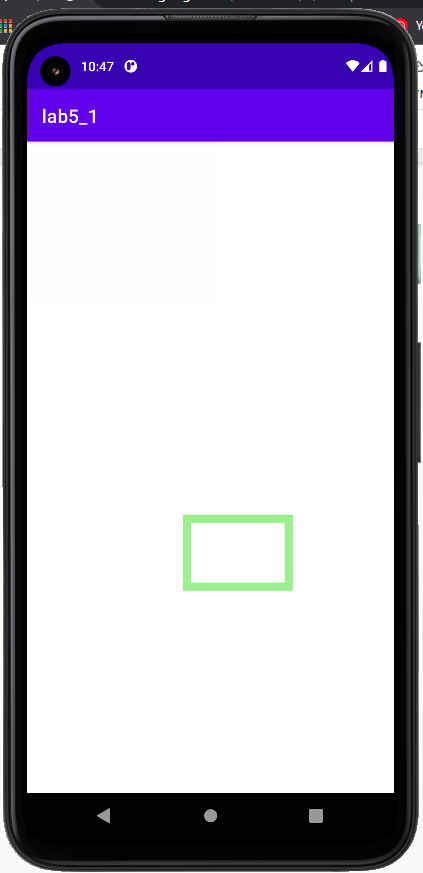
*Рис. 5.* Друга анімація, скріншот №2



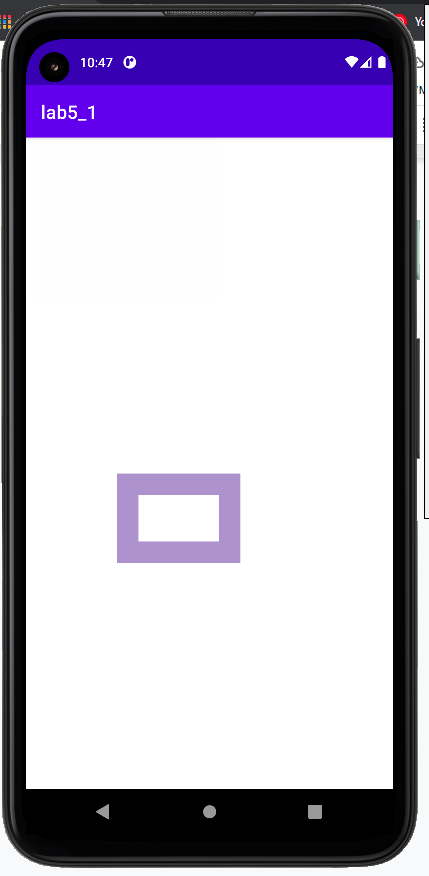
*Рис. 6.* Третя анімація, скріншот №1



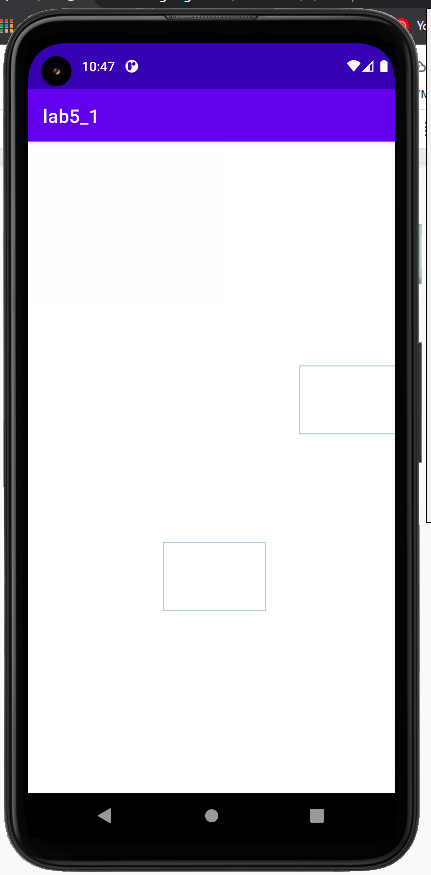
*Рис. 7.* Третя анімація, скріншот №2



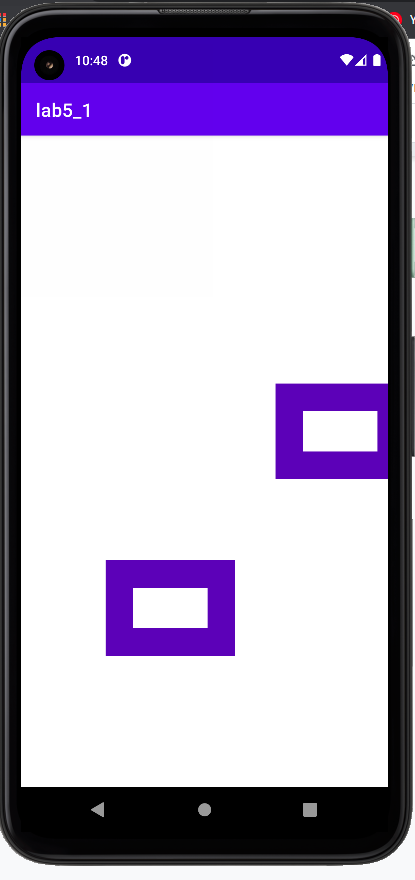
*Рис. 8.* Четверта анімація, скріншот №1



*Рис. 9.* Четверта анімація, скріншот №2



*Рис. 10.* П’ята анімація, скріншот №1



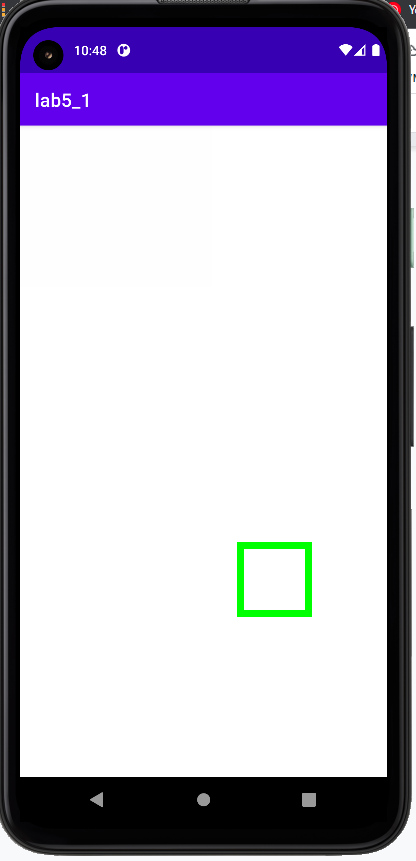
*Рис. 11.* П’ята анімація, скріншот №2



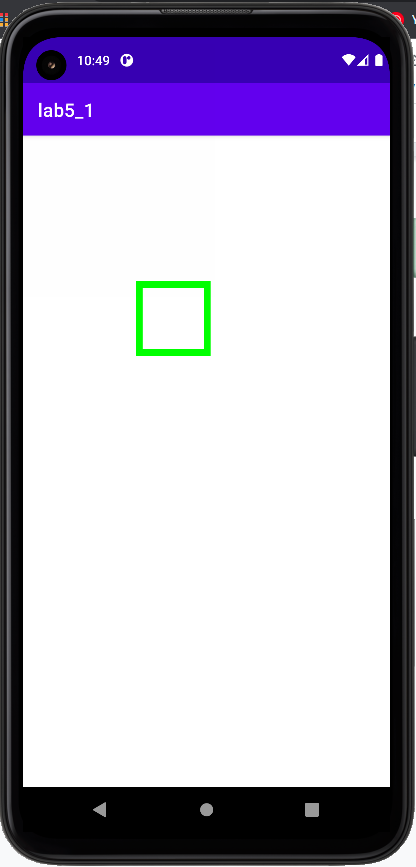
*Рис. 12.* Шоста анімація, скріншот №1



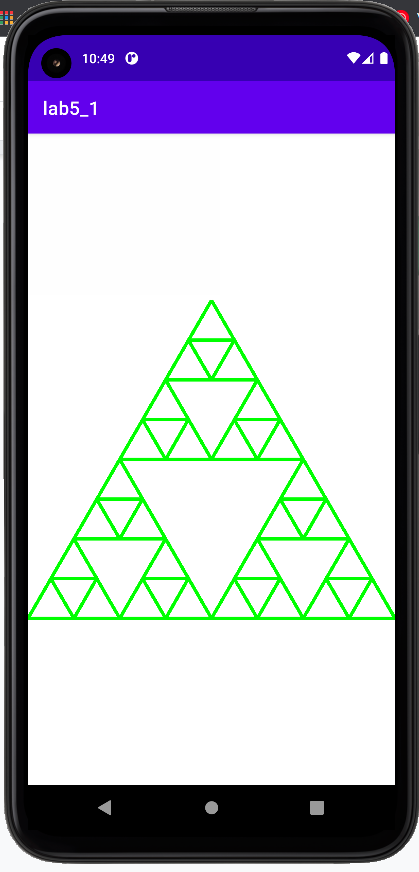
*Рис. 13.* Шоста анімація, скріншот №2



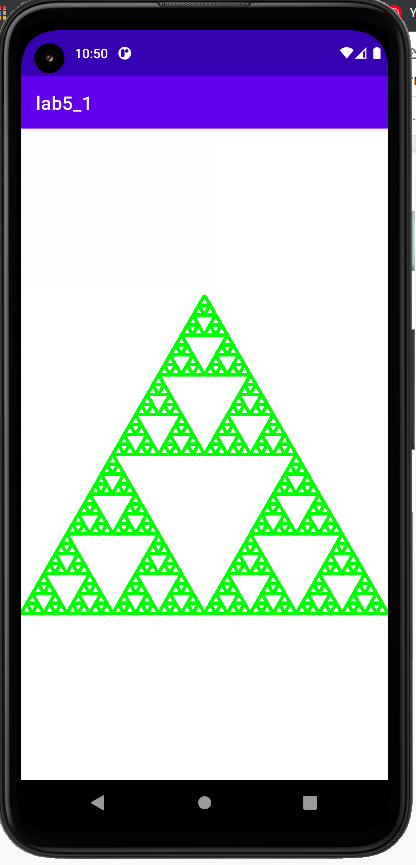
*Рис. 14.* Обертання квадрата по колу, скріншот №1



*Рис. 15.* Обертання квадрата по колу, скріншот №2



*Рис. 16.* Побудова килима Серпинського, скріншот №1



*Рис. 16.* Побудова килима Серпинського, скріншот №2

# **ВИСНОВКИ**

В ході виконання даної лабораторної роботи було створено додаток, який реалізує різні анімації.

# **СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ**

1. <http://hpc-education.ru/files/lectures/2011/ershov/ershov_2011_slides02.pdf>