Министерство науки и высшего образования Российской Федерации

**Федеральное государственное бюджетное образовательное учреждение**

**высшего образования**

«Владимирский государственный университет

имени Александра Григорьевича и Николая Григорьевича Столетовых»

(ВлГУ)

Кафедра информационных систем

и программной инженерии

**Лабораторные работы № 6-7**

**по дисциплине**

**«Технологии разработки мобильных приложений»**

Выполнил:

ст. гр. ПРИ-120

К. А. Борисова

Принял:

А. И. Рощина

Владимир, 2023

**Работа с Canvas**

**Цель работы**

Разработать мобильное приложение, которое будет давать возможность рисовать простейшие геометрические фигуры.

**Ход работы**

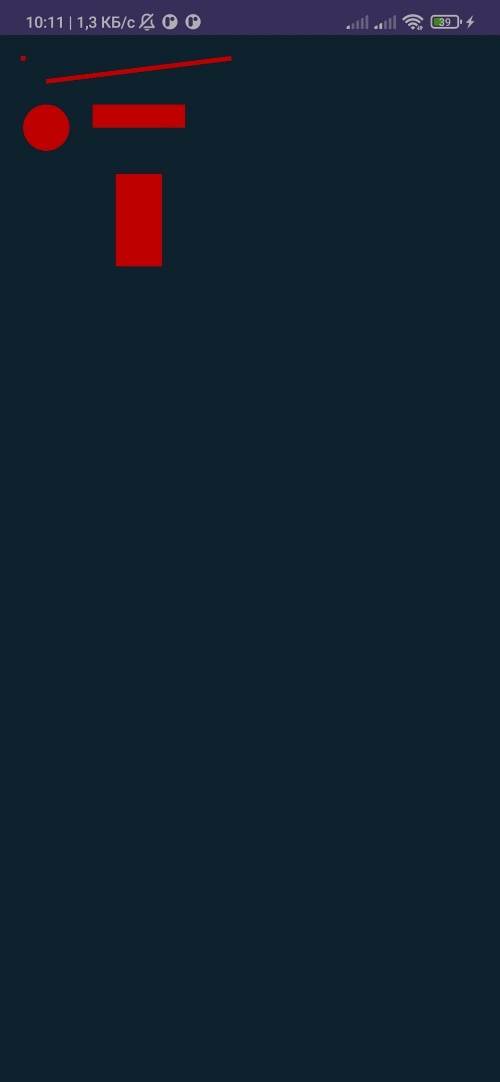
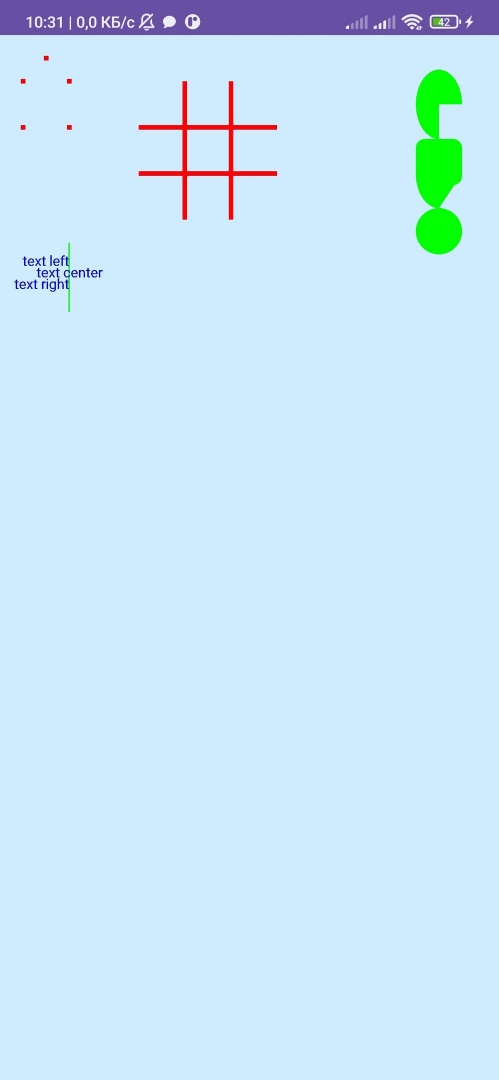
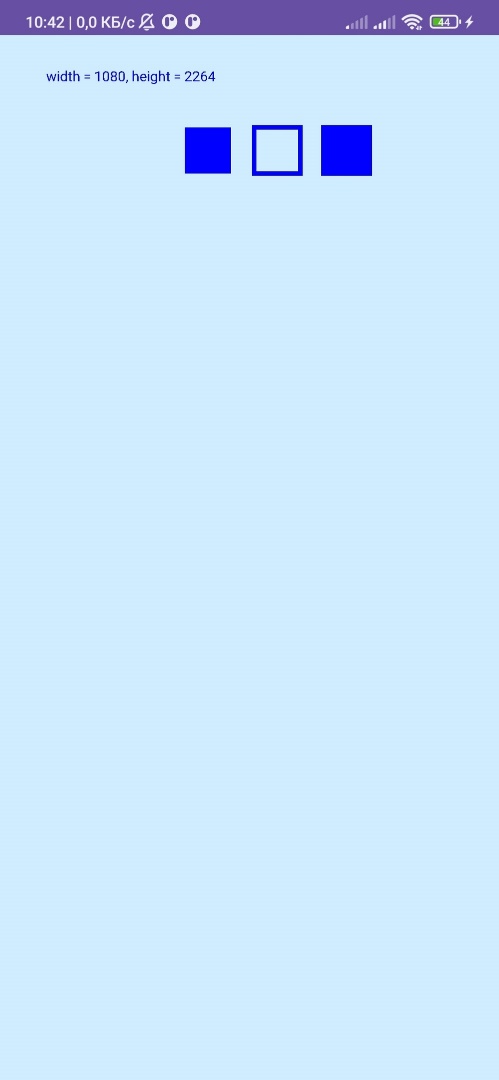
Повторим код из методических указаний и получим следующие результаты (рисунок 1) (последний вариант кода можно посмотреть по ссылке: [https://github.com/VV-I-S/trmp\_labs/tre e/main/Lab6\_2](https://github.com/VV-I-S/trmp_labs/tree/main/Lab6_2) ).

Рисунок . Упражнение из методических указаний

Далее с помощью инструментов canvas построим пятиугольник и звезду внутри него.

Листинг кода (<https://github.com/VV-I-S/trmp_labs/tree/main/Lab6_3> ):

public class MainActivity extends AppCompatActivity {  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(new DrawView(this));  
 }  
 class DrawView extends View {  
 Paint p;  
 Rect rect;  
 StringBuilder sb;  
 public DrawView(Context context) {  
 super(context);  
 p = new Paint();  
 Paint paint = new Paint();  
 rect = new Rect(100,200,200,300);  
 sb = new StringBuilder();  
 }  
  
 @Override  
 public void onDraw(Canvas canvas){  
 canvas.drawARGB(80, 102, 204, 255);  
 p.setColor(Color.*RED*);  
 p.setStrokeWidth(15);  
 p.setStyle(Paint.Style.*STROKE*);  
  
 // Определяем объект Path и закрываем пентаграмму  
 Path path1 = new Path();  
 path1.moveTo(550, 550);  
 path1.lineTo(950, 850);  
 path1.lineTo(800, 1350);  
 path1.lineTo(300, 1350);  
 path1.lineTo(150, 850);  
 path1.close();  
 // Рисуем по Пути, рисуем пятиконечную звезду  
 canvas.drawPath(path1, p);  
  
 p.setStyle(Paint.Style.*FILL*);  
 // Определяем объект Path и закрываем пентаграмму  
 Path path2 = new Path();  
 path2.moveTo(550, 550);  
 path2.lineTo(300, 1350);  
 path2.lineTo(950, 850);  
 path2.lineTo(150, 850);  
 path2.lineTo(800, 1350);  
 path2.close();  
 // Рисуем по Пути, рисуем пятиконечную звезду  
 canvas.drawPath(path2, p);  
 }  
 }  
}

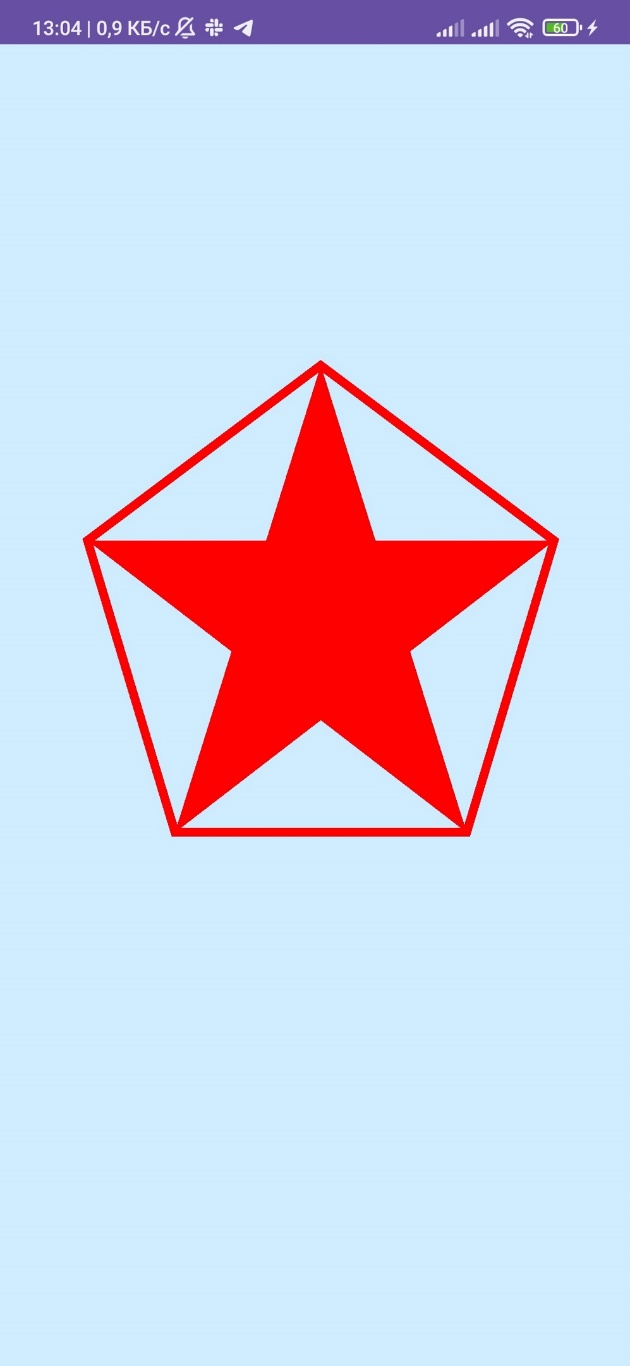
Результат работы в разных темах (рисунок 2):

Рисунок . Результат выполнения задания

**Обработка касаний экрана**

**Цель работы**

Разработать мобильное приложение, которое будет обрабатывать нажатия на экран.

**Ход работы**

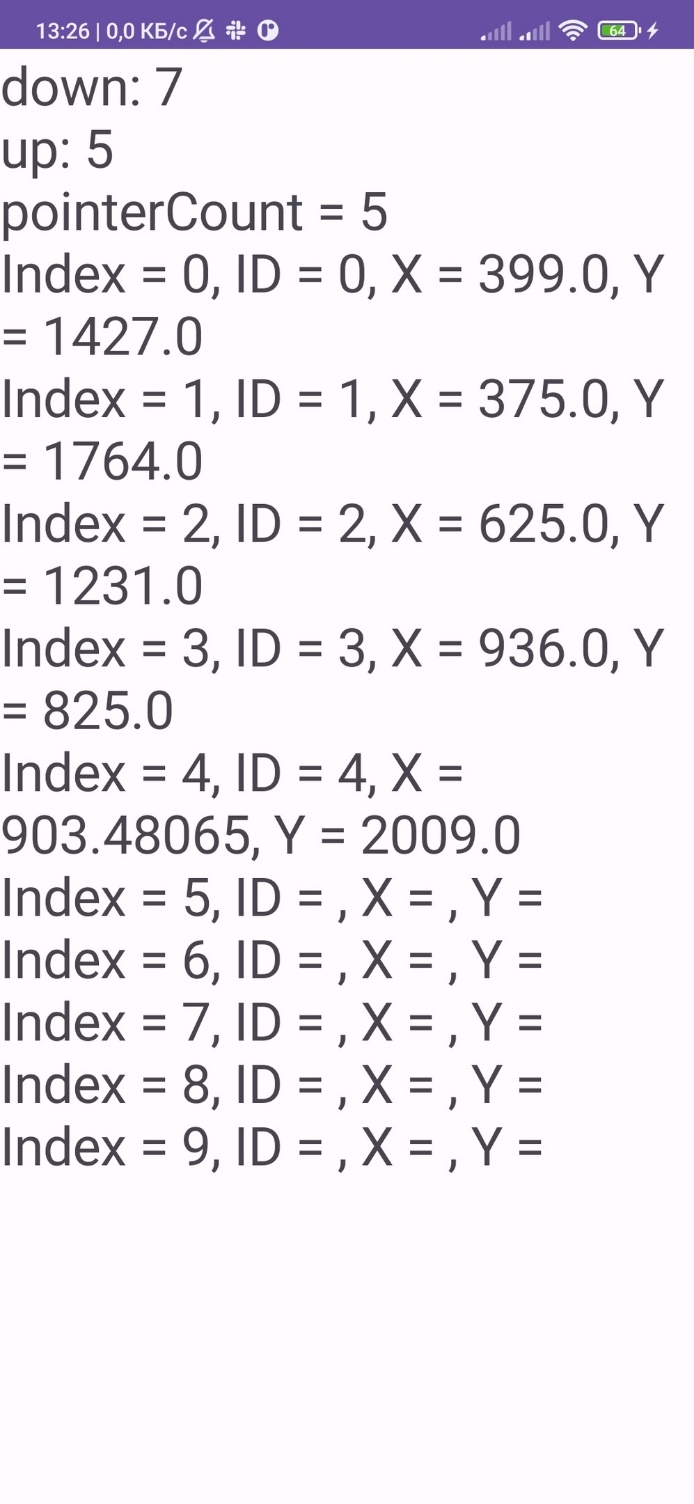
Повторим код из методических показаний (рисунок 3). Полный код можно посмотреть по ссылке: <https://github.com/VV-I-S/trmp_labs/tree/main/Lab7>

Рисунок . Результат работы программы

Выполним задание, реализуя рисование на экране. Было решено сделать применение рандомного цвета при каждом нажатии на экран.

Листинг кода:

public class MainActivity extends AppCompatActivity {  
  
 private ImageView imageView;  
 private float startX = -1, startY = -1, endX = -1, endY = -1;  
 private Bitmap bitmap;  
 private Canvas canvas;  
 private Paint p = new Paint();  
 Random r = new Random();  
  
 @SuppressLint("MissingInflatedId")  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
  
 imageView = findViewById(R.id.*imageView*);  
 }  
  
 private void onDrow(){  
 if (bitmap == null){  
 bitmap = Bitmap.*createBitmap*(imageView.getWidth(), imageView.getHeight(), Bitmap.Config.*ARGB\_8888*);  
 canvas = new Canvas(bitmap);  
 p.setAntiAlias(true);  
 p.setStyle(Paint.Style.*STROKE*);  
 p.setStrokeWidth(10);  
 }  
 canvas.drawLine(startX, startY -220, endX, endY -220, p);  
 imageView.setImageBitmap(bitmap);  
 }  
  
 @Override  
 public boolean onTouchEvent(MotionEvent event) {  
  
 if (event.getAction() == MotionEvent.*ACTION\_DOWN*){  
 startX = event.getX();  
 startY = event.getY();  
 p.setColor(Color.*rgb*((int)(Math.*random*()\*256), (int)(Math.*random*()\*256), (int)(Math.*random*()\*256)));  
 }  
  
 if (event.getAction() == MotionEvent.*ACTION\_MOVE*){  
 endX = event.getX();  
 endY = event.getY();  
 onDrow();  
 startX = event.getX();  
 startY = event.getY();  
 }  
 if (event.getAction() == MotionEvent.*ACTION\_UP*){  
 endX = event.getX();  
 endY = event.getY();  
 onDrow();  
 }  
 return super.onTouchEvent(event);  
 }  
}

Результат работы представлен на рисунке 4:

Рисунок . Результат выполнения программы

**Вывод**

В ходе данной работы были разработаны мобильные приложения, которые дают возможность рисовать простейшие геометрические фигуры и обрабатывают нажатия на экран.