Министерство образования и науки Российской Федерации

Калужский филиал федерального государственного бюджетного образовательного учреждения высшего образования

«Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)»

(КФ МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ "Фундаментальные науки"

КАФЕДРА "Программное обеспечение ЭВМ, информационные технологии и прикладная математика"

ОТЧЕТ

Лабораторная работа №3

ДИСЦИПЛИНА: " Мобильные операционные системы "

ТЕМА: " РАЗРАБОТКА ANDROID-ПРИЛОЖЕНИЙ С НЕСКОЛЬКИМИ ACTIVITY. РАБОТА С INTENT-ОБЪЕКТАМИ "

Выполнил: студент гр. ИТД-81 Турченко С.А._________________

Проверил: Гришунов С.С. _________________

Дата сдачи (защиты) лабораторной работы:

Результаты сдачи (защиты):

Количество рейтинговых баллов

Оценка

Калуга, 2018 г.

Цель:

1. Изучить механизм создания многоэкранных приложений на платформе Android
2. Научиться разрабатывать приложения с использованием явного и неявного вызова классов
3. Понять особенности реализации Android-приложений с использованием Intent-объектов

Задача:

Задание состоит из двух проектов. Все проекты используют Intent объекты. В первом проекте все требуемые параметры устанавливаются в главном Activity и результат отображается на второстепенном Activity. Во втором проекте результат отображается на главном Activity, а для задания соответствующих параметров используется несколько второстепенных Activity. Один из проектов реализовать с использованием неявного вызова (action), второй проект реализовать с явным вызовом класса(-ов).

Вариант 9

С использованием RadioButton выбрать фигуру, с помощью EditText задать цвет в пространстве RGB для выбранной фигуры, с использованием Button определить место вывода фигуры (9 областей). Отобразить фигуру с заданными параметрами.

Код программы

App1

MainActivity.java

```
public class MainActivity extends AppCompatActivity {

    DrawOption option;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        this.option = new DrawOption();
    }

    public void onPositionPick(View view) {
        int id = view.getId();
        switch (id) {
            case R.id.activity_main_position_top_left:
                this.option.PositionX = DrawOption.ePosition.LEFT;
                this.option.PositionY = DrawOption.ePosition.TOP;
                break;
            case R.id.activity_main_position_top_center:
                this.option.PositionX = DrawOption.ePosition.CENTER;
                this.option.PositionY = DrawOption.ePosition.TOP;
                break;
            case R.id.activity_main_position_top_right:
                this.option.PositionX = DrawOption.ePosition.RIGHT;
                this.option.PositionY = DrawOption.ePosition.TOP;
                break;
            case R.id.activity_main_position_center_left:
                this.option.PositionX = DrawOption.ePosition.LEFT;
                this.option.PositionY = DrawOption.ePosition.CENTER;
                break;
            case R.id.activity_main_position_center_center:
                this.option.PositionX = DrawOption.ePosition.CENTER;
                this.option.PositionY = DrawOption.ePosition.CENTER;
                break;
            case R.id.activity_main_position_center_right:
                this.option.PositionX = DrawOption.ePosition.RIGHT;
                this.option.PositionY = DrawOption.ePosition.CENTER;
                break;
            case R.id.activity_main_position_bottom_left:
                this.option.PositionX = DrawOption.ePosition.LEFT;
                this.option.PositionY = DrawOption.ePosition.BOTTOM;
                break;
            case R.id.activity_main_position_bottom_center:
                this.option.PositionX = DrawOption.ePosition.CENTER;
                this.option.PositionY = DrawOption.ePosition.BOTTOM;
                break;
            case R.id.activity_main_position_bottom_right:
                this.option.PositionX = DrawOption.ePosition.RIGHT;
                this.option.PositionY = DrawOption.ePosition.BOTTOM;
                break;
        }
    }

    public void parseColor() {
        EditText colorEditText = (EditText) findViewById(R.id.activity_main_color);
        String colorText = colorEditText.getText().toString();
        try {
            this.option.Color = Color.parseColor(colorText);
        }
        catch (Exception e) {
            return;
        }
    }

    public void onClick(View view) {
        this.parseColor();

        Intent intent = new Intent();
        intent.setAction("edu.bmstu.stas.lab3.VIEW_TEXT");
        intent = this.option.addToIntent(intent);
        startActivity(intent);
    }

    public void onFigurePick(View view) {
        int id = view.getId();
        switch (id) {
            case R.id.activity_main_figure_circle:
                this.option.Figure = DrawOption.eFigure.CIRCLE;
                break;
            case R.id.activity_main_figure_square:
                this.option.Figure = DrawOption.eFigure.SQUARE;
                break;
            case R.id.activity_main_figure_triangle:
                this.option.Figure = DrawOption.eFigure.TRIANGLE;
                break;
        }
    }
}
```

DrawView.java

```
class DrawView extends View {

    Paint paint;
    Canvas canvas;
    DrawOption option;


    public DrawView(Context context, DrawOption option) {
        super(context);

        this.paint = new Paint();
        this.paint.setColor(option.Color);
        this.option = option;
    }


    class Position {
        public int X;
        public int Y;
    }

    private Position calculatePosition(int width, int height) {
        Position result = new Position();

        int shiftX = 100;
        int shiftY = 100;
        int padding = 50;

        result.X = 0;
        switch (this.option.PositionX) {
            case LEFT:
                result.X = padding;
                break;
            case CENTER:
                result.X = width / 2 - shiftX;
                break;
            case RIGHT:
                result.X = width - shiftX * 2 - padding;
                break;
        }

        result.Y = 0;
        switch (this.option.PositionY) {
            case TOP:
                result.Y = padding;
                break;
            case CENTER:
                result.Y = height / 2 - shiftY;
                break;
            case BOTTOM:
                result.Y = height - shiftY * 2 - padding;
                break;
        }

        Log.d("figure", "calculated position as X:" + Integer.toString(result.X) +
                " and Y:" + Integer.toString(result.Y));
        return result;
    }

    private void drawTriangle(Canvas canvas, Position position) {
        // https://stackoverflow.com/questions/20544668/how-to-draw-filled-
triangle-on-android-canvas/22690364

        Point p1 = new Point(position.X, position.Y);
        Point p2 = new Point(position.X + 200, position.Y + 100);
        Point p3 = new Point(position.X, position.Y + 200);

        Path path = new Path();
        path.setFillType(Path.FillType.EVEN_ODD);
        path.moveTo(p1.x, p1.y);
        //path.lineTo(p1.x, p1.y);
        path.lineTo(p2.x, p2.y);
        path.lineTo(p3.x, p3.y);
        path.close();

        canvas.drawPath(path, paint);
    }

    @Override
    protected void onDraw(Canvas canvas) {
        this.canvas = canvas;

        // clear screen
        this.canvas.drawARGB(255, 255, 255, 255);
        Log.d("canvas", "canvas cleared");

        Position position = calculatePosition(canvas.getWidth(),
canvas.getHeight());

        Log.d("figure", "calculating figure and output it");
        switch (this.option.Figure) {
            case CIRCLE:
                Log.d("canvas", "draw circle");
                canvas.drawCircle(position.X + 100, position.Y + 100, 100,
this.paint);
```

```
                break;
            case SQUARE:
                Log.d("canvas", "draw rectangle");
                canvas.drawRect(position.X, position.Y, position.X + 200,
position.Y + 200, this.paint);
                break;
            case TRIANGLE:
                Log.d("canvas", "draw triangle");
                this.drawTriangle(canvas, position);
                break;
        }
    }

}
```

DrawOption.java

```
public class DrawOption {

    enum eFigure {
        CIRCLE,
        SQUARE,
        TRIANGLE
    }

    public enum ePosition {
        CENTER,
        LEFT,
        RIGHT,
        TOP,
        BOTTOM
    }

    public int Color;
    public eFigure Figure;
    public ePosition PositionX;
    public ePosition PositionY;

    public DrawOption() {
        this.Figure = eFigure.CIRCLE;
        this.PositionX = ePosition.CENTER;
        this.PositionY = ePosition.CENTER;
        this.Color = android.graphics.Color.BLUE;
    }

    public DrawOption(Bundle extras) {
        this.Figure = (eFigure) extras.getSerializable("figure");
        this.PositionX = (ePosition) extras.getSerializable("position_x");
        this.PositionY = (ePosition) extras.getSerializable("position_y");
        this.Color = extras.getInt("color");
    }

    public Intent addToIntent(Intent intent) {
        intent.putExtra("figure", this.Figure);
        intent.putExtra("position_x", this.PositionX);
        intent.putExtra("position_y", this.PositionY);
        intent.putExtra("color", this.Color);
        return intent;
    }

}
```

AboutActivity.java

```
public class AboutActivity extends Activity {

    DrawOption option;
    DrawView view;

    @Override
    protected void onCreate(@Nullable Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_about);

        Intent intent = getIntent();

        this.option = new DrawOption(intent.getExtras());

        this.draw();
    }

    private void draw() {
        view = new DrawView(this, option);
        setContentView(view);
    }


}
```

App2

MainActivity.java

```
public class MainActivity extends AppCompatActivity {

    DrawView canvas;
    DrawOption option;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        canvas = (DrawView) findViewById(R.id.canvas);
        option = new DrawOption();
    }

    public void onColor(View view) {
        Intent intent = new Intent(MainActivity.this, ColorActivity.class);
        startActivityForResult(intent, 0);
    }

    public void onFigure(View view) {
        Intent intent = new Intent(MainActivity.this, FigureActivity.class);
        startActivityForResult(intent, 1);

    }

    public void onPosition(View view) {
        Intent intent = new Intent(MainActivity.this, PositionActivity.class);
        startActivityForResult(intent, 2);
    }

    public void onActivityResult(int requestCode, int resultCode, Intent data) {
        if (data == null)
```

```
            return;

        switch (requestCode) {
            case 0:
                this.option.Color = data.getExtras().getInt("color");
                break;
            case 1:
                this.option.Figure = (DrawOption.eFigure)
data.getExtras().getSerializable("figure");
                break;
            case 2:
                this.option.PositionX = (DrawOption.ePosition)
data.getExtras().getSerializable("position_x");
                this.option.PositionY = (DrawOption.ePosition)
data.getExtras().getSerializable("position_y");
                break;
        }
        this.canvas.update(this.option);
    }
}
```

PositionActivity.java

```
public class PositionActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_position);
    }

    public void onPositionPick(View view) {
        int id = view.getId();

        DrawOption.ePosition positionX = DrawOption.ePosition.CENTER;
        DrawOption.ePosition positionY = DrawOption.ePosition.CENTER;

        switch (id) {
            case R.id.activity_main_position_top_left:
                positionX = DrawOption.ePosition.LEFT;
                positionY = DrawOption.ePosition.TOP;
                break;
            case R.id.activity_main_position_top_center:
                positionX = DrawOption.ePosition.CENTER;
                positionY = DrawOption.ePosition.TOP;
                break;
            case R.id.activity_main_position_top_right:
                positionX = DrawOption.ePosition.RIGHT;
                positionY = DrawOption.ePosition.TOP;
                break;
            case R.id.activity_main_position_center_left:
                positionX = DrawOption.ePosition.LEFT;
                positionY = DrawOption.ePosition.CENTER;
                break;
            case R.id.activity_main_position_center_center:
                positionX = DrawOption.ePosition.CENTER;
                positionY = DrawOption.ePosition.CENTER;
                break;
            case R.id.activity_main_position_center_right:
                positionX = DrawOption.ePosition.RIGHT;
                positionY = DrawOption.ePosition.CENTER;
                break;
            case R.id.activity_main_position_bottom_left:
                positionX = DrawOption.ePosition.LEFT;
                positionY = DrawOption.ePosition.BOTTOM;
                break;
            case R.id.activity_main_position_bottom_center:
                positionX = DrawOption.ePosition.CENTER;
                positionY = DrawOption.ePosition.BOTTOM;
                break;
            case R.id.activity_main_position_bottom_right:
                positionX = DrawOption.ePosition.RIGHT;
                positionY = DrawOption.ePosition.BOTTOM;
                break;
        }

        Intent intent = getIntent();
        intent.putExtra("position_x", positionX);
        intent.putExtra("position_y", positionY);
        setResult(Activity.RESULT_OK, intent);
        this.finish();
    }
}
```

FigureActivity.java

```
public class FigureActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_figure);
    }

    public void onChoice(View view) {
        int id = view.getId();

        DrawOption.eFigure figure = DrawOption.eFigure.CIRCLE;

        switch (id) {
            case R.id.activity_figure_circle:
                figure = DrawOption.eFigure.CIRCLE;
                break;
            case R.id.activity_figure_square:
                figure = DrawOption.eFigure.SQUARE;
                break;
            case R.id.activity_figure_triangle:
                figure = DrawOption.eFigure.TRIANGLE;
                break;
        }

        Intent intent = getIntent();
        intent.putExtra("figure", figure);
        setResult(Activity.RESULT_OK, intent);
        this.finish();
    }
}
```

DrawView.java

```
public class DrawView extends View {

    Paint paint;
    Canvas canvas;
    DrawOption option;

    // https://stackoverflow.com/questions/12872848/android-view-inflateexception-
binary-xml-file-line-when-using-custom-view
    public DrawView(Context context, AttributeSet st) {
        super(context, st);

        this.paint = new Paint();
        this.option = new DrawOption();
        this.applyColor();
    }

    class Position {
        public int X;
        public int Y;
    }

    private void applyColor() {
        this.paint.setColor(option.Color);
    }

    private Position calculatePosition(int width, int height) {
        Position result = new Position();

        int shiftX = 100;
        int shiftY = 100;
        int padding = 50;

        result.X = 0;
        switch (this.option.PositionX) {
            case LEFT:
                result.X = padding;
                break;
            case CENTER:
                result.X = width / 2 - shiftX;
                break;
            case RIGHT:
                result.X = width - shiftX * 2 - padding;
                break;
        }

        result.Y = 0;
        switch (this.option.PositionY) {
            case TOP:
                result.Y = padding;
                break;
            case CENTER:
                result.Y = height / 2 - shiftY;
                break;
            case BOTTOM:
                result.Y = height - shiftY * 2 - padding;
                break;
        }

        Log.d("figure", "calculated position as X:" + Integer.toString(result.X) +
                " and Y:" + Integer.toString(result.Y));
        return result;
    }
    private void drawTriangle(Canvas canvas, Position position) {
        // https://stackoverflow.com/questions/20544668/how-to-draw-filled-
triangle-on-android-canvas/22690364

        Point p1 = new Point(position.X, position.Y);
        Point p2 = new Point(position.X + 200, position.Y + 100);
        Point p3 = new Point(position.X, position.Y + 200);

        Path path = new Path();
        path.setFillType(Path.FillType.EVEN_ODD);
        path.moveTo(p1.x, p1.y);
        //path.lineTo(p1.x, p1.y);
        path.lineTo(p2.x, p2.y);
        path.lineTo(p3.x, p3.y);
        path.close();

        canvas.drawPath(path, paint);


    }

    @Override
    protected void onDraw(Canvas canvas) {
        super.onDraw(canvas);
        this.canvas = canvas;

        // clear screen
        this.canvas.drawARGB(255, 255, 255, 255);
        Log.d("canvas", "canvas cleared");

        Position position = calculatePosition(canvas.getWidth(),
canvas.getHeight());

        Log.d("figure", "calculating figure and output it");
        switch (this.option.Figure) {
            case CIRCLE:
                Log.d("canvas", "draw circle");
                canvas.drawCircle(position.X + 100, position.Y + 100, 100,
this.paint);
                break;
            case SQUARE:
                Log.d("canvas", "draw rectangle");
                canvas.drawRect(position.X, position.Y, position.X + 200,
position.Y + 200, this.paint);
                break;
            case TRIANGLE:
                Log.d("canvas", "draw triangle");
                this.drawTriangle(canvas, position);
                break;
        }
    }

    // https://stackoverflow.com/questions/25314707/how-to-combine-canvas-drawing-
with-android-activity-layout-which-includes-button
    public void update(DrawOption option) {
        this.option = option;
        this.applyColor();
        invalidate();
    }

}
```

DrawOption.java

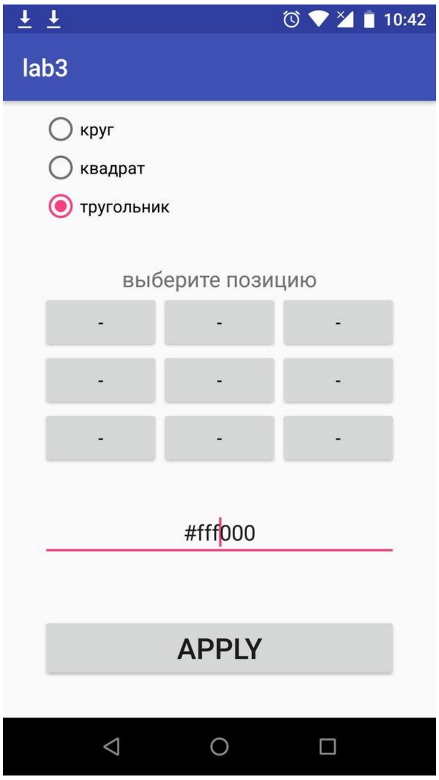
```
public class DrawOption {

    enum eFigure {
        CIRCLE,
        SQUARE,
        TRIANGLE
    }

    public enum ePosition {
        CENTER,
        LEFT,
        RIGHT,
        TOP,
        BOTTOM
    }

    public int Color;
    public eFigure Figure;
    public ePosition PositionX;
    public ePosition PositionY;

    public DrawOption() {
        this.Figure = eFigure.CIRCLE;
        this.PositionX = ePosition.CENTER;
        this.PositionY = ePosition.CENTER;
        this.Color = android.graphics.Color.BLUE;
    }

    public DrawOption(Bundle extras) {
        this.Figure = (eFigure) extras.getSerializable("figure");
        this.PositionX = (ePosition) extras.getSerializable("position_x");
        this.PositionY = (ePosition) extras.getSerializable("position_y");
        this.Color = extras.getInt("color");
    }

    public Intent addToIntent(Intent intent) {
        intent.putExtra("figure", this.Figure);
        intent.putExtra("position_x", this.PositionX);
        intent.putExtra("position_y", this.PositionY);
        intent.putExtra("color", this.Color);
        return intent;
    }

}
```

ColorActivity.java

```
public class ColorActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_color);
    }

    public void onApply(View view) {
        EditText colorEditText = (EditText)
findViewById(R.id.activity_color_field);
        String colorText = colorEditText.getText().toString();

        int color;

        try {
            color = Color.parseColor(colorText);
        } catch (Exception e) {
            color = Color.BLACK;
        }

        Intent intent = getIntent();
        intent.putExtra("color", color);
        setResult(Activity.RESULT_OK, intent);
        this.finish();
    }
}
```

Результат

App1

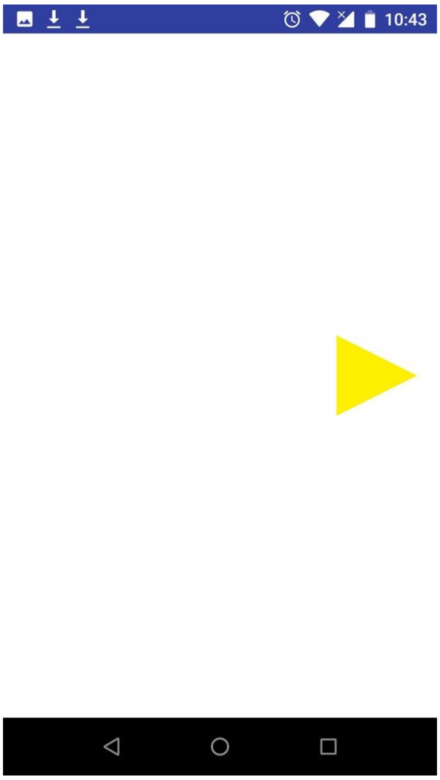


○ круг
○ квадрат
● тругольник

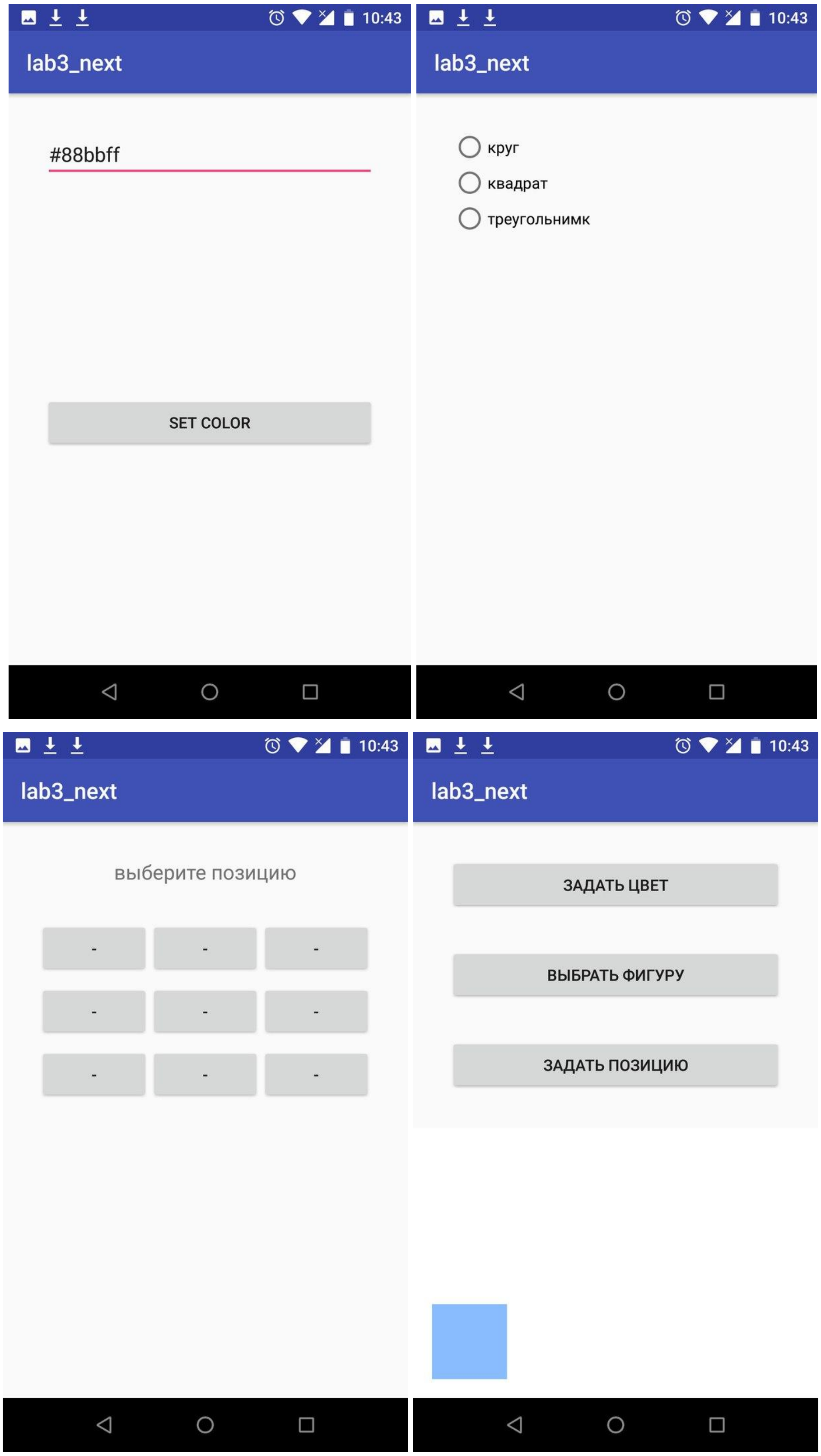
выберите позицию

-	-	-
-	-	-
-	-	-

#fff000



App2



#88bbff

Вывод

Были приобретены практические навыки разработки приложений с несколькими Activity и Intent-объектами.