

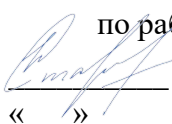
**ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ  
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ  
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»**

Факультет компьютерных наук  
Департамент программной инженерии

**СОГЛАСОВАНО**

Научный руководитель,  
Заместитель директора 1С

по работе с НИУ

 Н.Ю. Старичков  
«\_\_» \_\_\_\_\_ 2021 г.

**УТВЕРЖДАЮ**

Академический руководитель  
образовательной программы  
«Программная инженерия»

\_\_\_\_\_ В. В. Шилов  
«\_\_» \_\_\_\_\_ 2021 г.

**УМНЕЕ МЕНЯ – ЛИНИИ**  
**Текст программы**

**ЛИСТ УТВЕРЖДЕНИЯ**

**RU.17701729.05.03-01 12 01-1-ЛУ**

Исполнитель:

 студент группы БПИ193  
\_\_\_\_\_ /Дыдыпов Агван Валерьевич /  
«18» \_\_\_\_\_ мая 2021 г.

Подп. и дата

Инв. № дубл.

Взам. инв. №

Подп. и дата

Инв. № подл

RU.17701729.05.03-01  
12 01-1

**УТВЕРЖДЕНО**  
**RU.17701729.05.03-01 12 01–1-ЛЮ**

**УМНЕЕ МЕНЯ – ЛИНИИ**

**Текст программы**

**RU.17701729.05.03-01 12 01–1**

**Листов 30**

*Подп. и дата*

*Инв. № дубл.*

*Взам. инв. №*

*Подп. и дата*

*Инв. № подл*

RU.17701729.05.03-01  
12 01-1

## Аннотация

В данном документе приведен текст «компьютерной игра-головоломки “Мальчик-с-пальчик”».

Программа разработана на языке С#. Среда разработки - Microsoft Visual Studio Professional 2019, Unity3D 2019.

Функциональным назначением программы является генерация и сохранение алгеброгеометрических кодов, а также кодирование и декодирование некоторых сообщений с их помощью.

Настоящий документ разработан в соответствии с требованиями:

- 1) ГОСТ 19.101–77 Виды программ и программных документов [1];
- 2) ГОСТ 19.102–77 Стадии разработки [2];
- 3) ГОСТ 19.103–77 Обозначения программ и программных документов [3];
- 4) ГОСТ 19.104–78 Основные надписи [4];
- 5) ГОСТ 19.105–78 Общие требования к программным документам [5];
- 6) ГОСТ 19.106–78 Требования к программным документам, выполненным печатным способом [6];
- 7) ГОСТ 19.401–78 Текст программы. Требования к содержанию и оформлению [7].

Изменения к данному документу оформляются согласно ГОСТ 19.603–78 [8], ГОСТ 19.604–78 [9].

Перед прочтением данного документа рекомендуется ознакомиться с терминологией, приведенной в Приложении 1 и 2.

## СОДЕРЖАНИЕ

### Аннотация 2

Текст программы.....	3
1.1 Код класса AnswerLevel	3
1.2 Код класса MainActivity	8
1.3 Код класса MainComputer	10

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.03-01 12				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

1.4	Код класса MouseDemo	20
1.5	Код класса Node	22
1.6	Код класса Pair	23
1.7	Код класса PhotoLevel	24
ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ.....		29

## Текст программы

### 1.1 Код класса AnswerLevel

```
package kursuch.project.smarterthanmelines;  
  
import android.app.AlertDialog;  
import android.content.Context;
```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.03-01 12				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

import android.content.DialogInterface;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.view.Window;
import android.view.WindowManager;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.Toast;
import android.os.Bundle;
import android.app.Activity;
import android.app.AlertDialog;
import android.content.Context;
import android.content.DialogInterface;
import android.content.DialogInterface.OnCancelListener;
import android.content.DialogInterface.OnClickListener;
import android.view.Menu;
import android.view.View;
import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;

public class AnswerLevel extends AppCompatActivity {

    public static ImageView answerImage;
    Button back_btn;
    Button answer_btn;

    AlertDialog.Builder ad;
    Context context;

    public static boolean helpTrue = false;
    public static boolean helpFalse = false;

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

        answerImage = findViewById(R.id.image_answer);
        answerImage.setImageDrawable(PhotoLevel.mImageView.getDrawable());

        back_btn = findViewById(R.id.back_btn);
        back_btn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                try {
                    Intent intent = new Intent(AnswerLevel.this,
PhotoLevel.class);
                    startActivity(intent);
                    finish();
                    System.out.println("back");
                } catch (Exception e) {
                }
            }
        });
    }
}

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.03-01 12				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

answer_btn = findViewById(R.id.solution_btn);
answer_btn.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        MainComputer.startProgram();
    }
});

AlertDialog.Builder builder = new
AlertDialog.Builder(AnswerLevel.this);
builder.setTitle("Обратите внимание!")
    .setMessage("Сначала выберите начальную и конечную точку,
между которыми надо найти путь. " +
        "Затем для отображения решения по частям повторно
нажимайте в любое место на картинке.")
    .setNegativeButton("OK",
        new DialogInterface.OnClickListener() {
            public void onClick(DialogInterface dialog, int
id) {
                dialog.cancel();
                helpTrue = true;
            }
        });
AlertDialog alert = builder.create();

AlertDialog.Builder builderFalse = new
AlertDialog.Builder(AnswerLevel.this);
builderFalse.setTitle("Обратите внимание!")
    .setMessage("Сначала выберите начальную и конечную точку,
между которыми надо найти путь. " +
        "Затем для отображения полного решения повторно
нажмите в любое место на картинке.")
    .setNegativeButton("OK",
        new DialogInterface.OnClickListener() {
            public void onClick(DialogInterface dialog, int
id) {
                dialog.cancel();
                helpTrue = false;
            }
        });
AlertDialog alertFalse = builder.create();

context = AnswerLevel.this;
String title = "Подсказки";
String message = "Включить подсказки по решению лабиринта? Подсказки
- отображение лабиринта по частям";
String button1String = "Да";
String button2String = "Нет";

ad = new AlertDialog.Builder(context);
ad.setTitle(title); // заголовок
ad.setMessage(message); // сообщение
ad.setPositiveButton(button1String, new OnClickListener() {
    public void onClick(DialogInterface dialog, int arg1) {
        MainComputer.helpSolMaze = true;
        Toast.makeText(context, "Подсказки включены",
            Toast.LENGTH_LONG).show();
        System.out.println(MainComputer.helpSolMaze);
    }
});

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.03-01 12				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

        if(MainComputer.helpSolMaze == true && helpTrue == false){
            alert.show();
        }
    });
    ad.setNegativeButton(button2String, new OnClickListener() {
        public void onClick(DialogInterface dialog, int arg1) {
            MainComputer.helpSolMaze = false;
            Toast.makeText(context, "Подсказки отключены",
                Toast.LENGTH_LONG)
                .show();
            if(MainComputer.helpSolMaze == false && helpFalse == false){
                alertFalse.show();
            }
        }
    });
    ad.setCancelable(true);
    ad.setOnCancelListener(new OnCancelListener() {
        public void onCancel(DialogInterface dialog) {
            MainComputer.helpSolMaze = false;
            Toast.makeText(context, "Подсказки отключены",
                Toast.LENGTH_LONG).show();
        }
    });

    ad.show();

    Window w = getWindow();
    w.setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN,
        WindowManager.LayoutParams.FLAG_FULLSCREEN);
}

public void errorPath()
{
    AlertDialog.Builder builder = new AlertDialog.Builder(this);
    builder.setTitle("Не удалось построить путь!")
        .setMessage("Попробуйте еще раз.")
        .setNegativeButton("OK",
            new DialogInterface.OnClickListener() {
                public void onClick(DialogInterface dialog, int
id) {
                    dialog.cancel();
                }
            });
    AlertDialog alert = builder.create();
    alert.show();
}
}
XML

```

```

<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.03-01 12				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

<ImageView
    android:id="@+id/photo_background"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:scaleType="centerCrop"
    app:srcCompat="@drawable/black" />

<LinearLayout
    android:id="@+id/container1"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:gravity="center"
    android:orientation="vertical">

    <ImageView
        android:id="@+id/image_answer"
        android:layout_width="match_parent"
        android:layout_height="450dp"
        android:scaleType="centerCrop"/>

    <LinearLayout
        android:layout_width="268dp"
        android:layout_height="158dp"
        android:layout_marginTop="0dp"
        android:orientation="horizontal">

        <Button
            android:id="@+id/back_btn"
            android:layout_width="0dp"
            android:layout_height="wrap_content"
            android:layout_marginTop="35dp"
            android:layout_marginRight="10dp"
            android:layout_weight="1"
            android:background="@drawable/btn_stroke_white30_press_white"
            android:paddingLeft="10dp"
            android:paddingRight="10dp"
            android:text="@string/back"
            android:textColor="@color/black"
            android:textSize="24sp" />

        <Button
            android:id="@+id/solution_btn"
            android:layout_width="7dp"
            android:layout_height="wrap_content"
            android:layout_marginTop="35dp"
            android:layout_weight="1"
            android:background="@drawable/btn_stroke_white30_press_white"
            android:paddingLeft="10dp"
            android:paddingRight="10dp"
            android:text="@string/answer"
            android:textColor="@color/black"
            android:textSize="24sp" />

    </LinearLayout>
</LinearLayout>
</RelativeLayout>

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.03-01 12				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата



```
package kursuch.project.smarterthanmelines;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.view.Window;
import android.view.WindowManager;
import android.widget.Button;
import android.app.AlertDialog;
import android.content.DialogInterface;
import org.opencv.android.OpenCVLoader;

public class MainActivity extends AppCompatActivity {

    public static boolean isOpened = false;
    static {
        OpenCVLoader.initDebug();
        if(!OpenCVLoader.initDebug())
        {
            Log.i("OpenCV", "Not working");
        }else
        {
            Log.i("OpenCV", "successfully working");
        }
    }

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

        AlertDialog.Builder builder = new
        AlertDialog.Builder(MainActivity.this);
        builder.setTitle("Привет!")
            .setMessage("Это приложение решает лабиринт по фотографии.")
            .setNegativeButton("ОК",
                new DialogInterface.OnClickListener() {
                    public void onClick(DialogInterface dialog, int
id) {

                        dialog.cancel();
                        isOpened = true;
                    }
                })
            .setPositiveButton("Начать",
                new DialogInterface.OnClickListener() {
                    public void onClick(DialogInterface dialog, int
id) {
                        startActivity(new Intent(MainActivity.this,
                        MainActivity.class));
                    }
                });
        AlertDialog alert = builder.create();
        if(isOpened == false)
            alert.show();

        Button btn_start = (Button)findViewById(R.id.btn_start);
        btn_start.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                startActivity(new Intent(MainActivity.this,
                MainActivity.class));
            }
        });
    }
}
```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.03-01 12				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

        try {
            Intent intent = new Intent(MainActivity.this,
PhotoLevel.class);
            startActivity(intent);
            finish();
        } catch (Exception e) {

        }

    }

});

Window w = getWindow();
w.setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN,
WindowManager.LayoutParams.FLAG_FULLSCREEN);
    }
}

```

### XML

```

<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <ImageView
        android:id="@+id/background"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:scaleType="centerCrop"
        app:srcCompat="@drawable/menu_background" />

    <LinearLayout
        android:id="@+id/startLayout"
        android:layout_width="wrap_content"
        android:layout_height="100dp"
        android:layout_alignParentBottom="true"
        android:layout_centerHorizontal="true"
        android:padding="10dp"
        android:orientation="horizontal">

        <Button
            android:id="@+id/btn_start"
            android:layout_width="0dp"
            android:layout_height="wrap_content"
            android:background="@drawable/btn_stroke_white30_press_white"
            android:layout_weight="1"
            android:paddingLeft="10dp"
            android:paddingRight="10dp"
            android:textColor="@color/black"
            android:textSize="24sp"
            android:text="@string/start" />

    </LinearLayout>
</RelativeLayout>

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.03-01 12				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

### 1.3 Код класса MainComputer

```
package kursuch.project.smarterthanmelines;

import android.graphics.Bitmap;
import android.graphics.drawable.BitmapDrawable;
import android.graphics.drawable.Drawable;
import android.widget.ImageView;

import org.opencv.android.Utils;
import org.opencv.core.*;
import org.opencv.core.Point;
import org.opencv.imgcodecs.Imgcodecs;
import org.opencv.imgproc.Imgproc;
import java.io.IOException;
import java.util.ArrayList;
import java.util.List;
import org.opencv.core.Mat;

public class MainComputer {

    static int p = 0;
    static Point start;
    static Point end;
    public static ArrayList<Pair> path;
    public static boolean helpSolMaze = false;

    public static Mat to_draw;

    public static int find_width_end() {
        int width = 1;
        while (true) {
            if (to_draw.get((int)end.y - width, (int)end.x)[0] == 0
                && to_draw.get((int)end.y - width, (int)end.x)[1] == 255
                && to_draw.get((int)end.y - width, (int)end.x)[2] == 0) {
                return width;
            }
            if (to_draw.get((int)end.y + width, (int)end.x)[0] == 0
                && to_draw.get((int)end.y + width, (int)end.x)[1] == 255
                && to_draw.get((int)end.y + width, (int)end.x)[2] == 0) {
                return width;
            }
            if (to_draw.get((int)end.y, (int)end.x + width)[0] == 0
                && to_draw.get((int)end.y, (int)end.x + width)[1] == 255
                && to_draw.get((int)end.y, (int)end.x + width)[2] == 0) {
                return width;
            }
            if (to_draw.get((int)end.y, (int)end.x - width)[0] == 0
                && to_draw.get((int)end.y, (int)end.x - width)[1] == 255
                && to_draw.get((int)end.y, (int)end.x - width)[2] == 0) {
                return width;
            }
            if (to_draw.get((int)end.y - width, (int)end.x - width)[0] == 0
                && to_draw.get((int)end.y - width, (int)end.x - width)[1]
                == 255
                && to_draw.get((int)end.y - width, (int)end.x - width)[2]
```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.03-01 12				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```
== 0) {
    return width;
}
if (to_draw.get((int)end.y + width, (int)end.x - width)[0] == 0
    && to_draw.get((int)end.y + width, (int)end.x - width)[1]
== 255
    && to_draw.get((int)end.y + width, (int)end.x - width)[2]
== 0) {
    return width;
}
if (to_draw.get((int)end.y - width, (int)end.x + width)[0] == 0
    && to_draw.get((int)end.y - width, (int)end.x + width)[1]
== 255
    && to_draw.get((int)end.y - width, (int)end.x + width)[2]
== 0) {
    return width;
}
if (to_draw.get((int)end.y + width, (int)end.x + width)[0] == 0
    && to_draw.get((int)end.y + width, (int)end.x + width)[1]
== 255
    && to_draw.get((int)end.y + width, (int)end.x + width)[2]
== 0) {
    return width;
}
++width;
}

public static int find_width_start() {
    int width = 1;
    while (true) {
        if (to_draw.get((int)start.y - width, (int)start.x)[0] == 0
            && to_draw.get((int)start.y - width, (int)start.x)[1] ==
255
            && to_draw.get((int)start.y - width, (int)start.x)[2] ==
0) {
            return width;
        }
        if (to_draw.get((int)start.y + width, (int)start.x)[0] == 0
            && to_draw.get((int)start.y + width, (int)start.x)[1] ==
255
            && to_draw.get((int)start.y + width, (int)start.x)[2] ==
0) {
            return width;
        }
        if (to_draw.get((int)start.y, (int)start.x + width)[0] == 0
            && to_draw.get((int)start.y, (int)start.x + width)[1] ==
255
            && to_draw.get((int)start.y, (int)start.x + width)[2] ==
0) {
            return width;
        }
        if (to_draw.get((int)start.y, (int)start.x - width)[0] == 0
            && to_draw.get((int)start.y, (int)start.x - width)[1] ==
255
            && to_draw.get((int)start.y, (int)start.x - width)[2] ==
0) {
            return width;
        }
    }
}
```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.03-01 12				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```
    }
    if (to_draw.get((int)start.y - width, (int)start.x - width)[0] ==
0
        width)[1] == 255
        && to_draw.get((int)start.y - width, (int)start.x -
        width)[2] == 0) {
            return width;
        }
    if (to_draw.get((int)start.y + width, (int)start.x - width)[0] ==
0
        width)[1] == 255
        && to_draw.get((int)start.y + width, (int)start.x -
        width)[2] == 0) {
            return width;
        }
    if (to_draw.get((int)start.y - width, (int)start.x + width)[0] ==
0
        width)[1] == 255
        && to_draw.get((int)start.y - width, (int)start.x +
        width)[2] == 0) {
            return width;
        }
    if (to_draw.get((int)start.y + width, (int)start.x + width)[0] ==
0
        width)[1] == 255
        && to_draw.get((int)start.y + width, (int)start.x +
        width)[2] == 0) {
            return width;
        }
    ++width;
}

public static int find_width()
{
    return Math.min(find_width_end(), find_width_start());
}

public static boolean is_way(int curr_i, int curr_j) {
    return (to_draw.get(curr_i, curr_j)[0] == 255 && to_draw.get(curr_i,
curr_j)[1] == 255
        && to_draw.get(curr_i, curr_j)[2] == 255);
}

public static boolean is_wall(int curr_i, int curr_j) {
    return (to_draw.get(curr_i, curr_j)[0] == 0 && to_draw.get(curr_i,
curr_j)[1] == 255
        && to_draw.get(curr_i, curr_j)[2] == 0) ||
        (to_draw.get(curr_i, curr_j)[0] == 0 && to_draw.get(curr_i, curr_j)[1] == 0
        && to_draw.get(curr_i, curr_j)[2] == 255);
}

public static void drawFirst()
{

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.03-01 12				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

    for (int i = 0; i < path.size() / 4 ; i++) {
        Scalar colour = new Scalar(255, 255, 255);
        Point p = new Point();
        p.x = path.get(i).second;
        p.y = path.get(i).first;
        Imgproc.circle(to_draw, p, 2, colour, 2);
    }
}

public static void drawSecond()
{
    for (int i = path.size() / 4; i < path.size() * 2 / 4; i++) {
        Scalar colour = new Scalar(255, 255, 255);
        Point p = new Point();
        p.x = path.get(i).second;
        p.y = path.get(i).first;
        Imgproc.circle(to_draw, p, 2, colour, 2);
    }
}

public static void drawThird()
{
    for (int i = path.size() * 2 / 4; i < path.size() * 3 / 4; i++) {
        Scalar colour = new Scalar(255, 255, 255);
        Point p = new Point();
        p.x = path.get(i).second;
        p.y = path.get(i).first;
        Imgproc.circle(to_draw, p, 2, colour, 2);
    }
}

public static void drawFourth()
{
    for (int i = path.size() * 3 / 4; i < path.size(); i++) {
        Scalar colour = new Scalar(255, 255, 255);
        Point p = new Point();
        p.x = path.get(i).second;
        p.y = path.get(i).first;
        Imgproc.circle(to_draw, p, 2, colour, 2);
    }
}

public static void drawAll()
{
    for (int i = 0; i < path.size(); i++) {
        Scalar colour = new Scalar(255, 255, 255);
        Point p = new Point();
        p.x = path.get(i).second;
        p.y = path.get(i).first;
        Imgproc.circle(to_draw, p, 2, colour, 2);
    }
}

public static void BFS() throws IOException {
    int rows = to_draw.rows();
    int cols = to_draw.cols();
    int curr_i = (int)start.y;
    int curr_j = (int)start.x;
    int start_i = curr_i;
    int start_j = curr_j;
    int end_j = (int)end.x;
    int end_i = (int)end.y;

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.03-01 12				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

int width = find_width() - 3;
int[][] visited = new int[rows][cols];
path = new ArrayList<Pair>();
Node[][] matrix = new Node[rows][cols];
for (int i = 0; i < rows; ++i) {
    for (int j = 0; j < cols; ++j) {
        matrix[i][j] = new Node(0, 0, false);
        boolean check = is_wall(i, j);
        matrix[i][j].setIs_wall(check);
        matrix[i][j].blue = (Math.abs(end_j - j) + Math.abs(end_i -
i)) * 10;
        visited[i][j] = 0;
    }
}
for (int i = 0; i < rows; ++i) {
    for (int j = 0; j < cols; ++j) {
        if (matrix[i][j].isIs_wall()) {
            for (int k = 0; k < width; ++k) {
                if (i > k) {
                    matrix[i - k - 1][j].close_wall = true;
                }
                if (j > k) {
                    matrix[i][j - k - 1].close_wall = true;
                }
                if (i < rows - k - 1) {
                    matrix[i + k + 1][j].close_wall = true;
                }
                if (j < cols - k - 1) {
                    matrix[i][j + k + 1].close_wall = true;
                }
            }
        }
    }
}

Pair[][] parent = new Pair[rows][cols];
ArrayList<Pair> queue = new ArrayList<Pair>();
ArrayList<Pair> dir4 = new ArrayList<Pair>();
dir4.add(new Pair(-1, 0));
dir4.add(new Pair(1, 0));
dir4.add(new Pair(0, 1));
dir4.add(new Pair(0, -1));
dir4.add(new Pair(-1, 1));
dir4.add(new Pair(1, 1));
dir4.add(new Pair(-1, -1));
dir4.add(new Pair(1, -1));
Pair pnt = new Pair(start_i, start_j);
queue.add(pnt);
visited[start_i][start_j] = 1;
Pair par = null;
boolean found = false;
while (queue.size() > 0) {
    par = queue.get(0);
    queue.remove(0);
    for (Pair p : dir4) {
        curr_i = p.first + par.first;
        curr_j = p.second + par.second;
        if (curr_i >= 0 && curr_i < rows && curr_j >= 0 && curr_j <

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.03-01 12				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```
cols && !is_wall(curr_i, curr_j)
    && visited[curr_i][curr_j] == 0 &&
!matrix[curr_i][curr_j].close_wall) {
    queue.add(new Pair(curr_i, curr_j));
    visited[curr_i][curr_j] = visited[par.first][par.second]
+ 1;

    parent[curr_i][curr_j] = par;
    if (curr_i == end_i && curr_j == end_j) {
        found = true;
        break;
    }
}
}
}
if (found) {
    par.first = end_i;
    par.second = end_j;
    while (!(par.first == start_i && par.second == start_j)) {
        path.add(par);
        par = parent[par.first][par.second];
    }
    path.add(par);
} else {
    //AnswerLevel.errorPath();
    System.out.println("Path not found");
}

System.out.println("ready BFS");

//MatOfByte matOfByte = new MatOfByte();
//Imgcodecs.imencode(".jpg", to_draw, matOfByte);           //перевод
изображения в байтовую матрицу
//byte[] byteArray = matOfByte.toArray();                   //перевод
байтовой матрицы в массив
//InputStream in = new ByteArrayInputStream(byteArray);      //перевод в
буферизованное изображени
//BufferedImage bufImage = ImageIO.read(in);
//JFrame frame = new JFrame();                               //создаем
окно
//frame.getContentPane().add(new JLabel(new ImageIcon(bufImage)));
//показываем изображение
//frame.pack();
//frame.setVisible(true);
}

public static void mouse_event(int pX, int pY) throws IOException {
    if (p == 0) {
        start = new Point();
        start.x = pX; // + (int)AnswerLevel.answerImage.getX(); // -7;
        start.y = pY; // + (int)AnswerLevel.answerImage.getY(); // - 28;
        System.out.println(start.x);
        System.out.print(" " + start.y);
        System.out.println();
        Scalar colour = new Scalar(255, 255, 255);
        Imgproc.circle(to_draw, start, 3, colour, 3);
        p = 1;

        Drawable drawable = convertMatToDrawable(to_draw);
    }
}
```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.03-01 12				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата



```
AnswerLevel.answerImage.setImageDrawable(drawable);

System.out.println("ready 1");

//MatOfByte matOfByte = new MatOfByte();
//Imgcodecs.imencode(".jpg", to_draw, matOfByte);
//перевод изображения в байтовую матрицу
//byte[] byteArray = matOfByte.toArray();
//перевод байтовой матрицы в массив
//InputStream in = new ByteArrayInputStream(byteArray);
//перевод в буферизованное изображение
//BufferedImage bufImage = ImageIO.read(in);
//JFrame frame = new JFrame();
//создаем окно
//frame.getContentPane().add(new JLabel(new
ImageIcon(bufImage))); //показываем изображение
//frame.pack();
//frame.setVisible(true);
} else if (p == 1) {
    end = new Point();
    end.x = pX; // - 7;
    end.y = pY; // - 28;
    System.out.println(end.x);
    System.out.print(" " + end.y);
    System.out.println();
    Scalar colour = new Scalar(255, 255, 255);
    Imgproc.circle(to_draw, end, 3, colour, 3);
    p = 2;

    Drawable drawable = convertMatToDrawable(to_draw);
    AnswerLevel.answerImage.setImageDrawable(drawable);

    System.out.println("ready 2");
    //MatOfByte matOfByte = new MatOfByte();
    //Imgcodecs.imencode(".jpg", to_draw, matOfByte);
    //перевод изображения в байтовую матрицу
    //byte[] byteArray = matOfByte.toArray();
    //перевод байтовой матрицы в массив
    //InputStream in = new ByteArrayInputStream(byteArray);
    //перевод в буферизованное изображение
    //BufferedImage bufImage = ImageIO.read(in);
    //JFrame frame = new JFrame();
    //создаем окно
    //frame.getContentPane().add(new JLabel(new
    ImageIcon(bufImage))); //показываем изображение
    //frame.pack();
    //frame.setVisible(true);
}
else if (p == 2) {
    BFS();
    p = 3;
    if (helpSolMaze == false)
        p = 7;
}
else if (p == 3 && helpSolMaze == true)
{
    drawFourth();
    Drawable drawable = convertMatToDrawable(to_draw);
```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.03-01 12				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```
        AnswerLevel.answerImage.setImageDrawable(drawable);
        p = 4;
    }
    else if (p == 4 && helpSolMaze == true)
    {
        drawThird();
        Drawable drawable = convertMatToDrawable(to_draw);
        AnswerLevel.answerImage.setImageDrawable(drawable);
        p = 5;
    }
    else if (p == 5 && helpSolMaze == true)
    {
        drawSecond();
        Drawable drawable = convertMatToDrawable(to_draw);
        AnswerLevel.answerImage.setImageDrawable(drawable);
        p = 6;
    }
    else if (p == 6 && helpSolMaze == true)
    {
        drawFirst();
        Drawable drawable = convertMatToDrawable(to_draw);
        AnswerLevel.answerImage.setImageDrawable(drawable);
        p = 0;
    }
    else if (p == 7 && helpSolMaze == false)
    {
        drawAll();
        Drawable drawable = convertMatToDrawable(to_draw);
        AnswerLevel.answerImage.setImageDrawable(drawable);
        p = 0;
    }
}

public static Mat convertImageViewMat(ImageView v){
    Bitmap bm=((BitmapDrawable)v.getDrawable()).getBitmap();
    Mat mat = new Mat();
    Bitmap bmp32 = bm.copy(Bitmap.Config.ARGB_8888, true);
    Utils.bitmapToMat(bmp32, mat);
    return mat;
}

public static Drawable convertMatToDrawable(Mat tmp) {
    Bitmap bmp = null;
    bmp = Bitmap.createBitmap(tmp.cols(), tmp.rows(),
    Bitmap.Config.ARGB_8888);
    Utils.matToBitmap(tmp, bmp);
    Drawable d = new BitmapDrawable(bmp);
    return d;
}

public static void startProgram() {
    //System.loadLibrary(Core.NATIVE_LIBRARY_NAME);    //Загружаем
    библиотека OpenCV

    Imgcodecs imageCodecs = new Imgcodecs();        //Создаем экземпляр
    класса imagecodecs

    Mat src_gray = new Mat();                        //бинарное
```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.03-01 12				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```
изображение
    Mat src; //исходное
изображение
    int thresh = 100; //пороговое значение

    //String file = "C://Users//79852//Desktop//Screens//maze.jpg";
//путь до изображения
    //src = Imgcodecs.imread(file); //загружаем
изображение из файла
    // image to byte array
    src = convertImageViewMat(AnswerLevel.answerImage);

    if (src.empty()) { //обработка ошибки
загрузки
        System.out.println("Could not open or find the image!\n");
        // предложить загрузить заново
    }
    Imgproc.cvtColor(src, src, Imgproc.COLOR_BGR2RGB); //конвертирование
изображения
    Imgproc.cvtColor(src, src, Imgproc.COLOR_RGB2GRAY);
    Imgproc.GaussianBlur(src, src, new Size(15, 15), 0);
//Imgproc.threshold(src, src, 0, 255, Imgproc.THRESH_OTSU);
    Imgproc.adaptiveThreshold(src, src, 255,
    Imgproc.ADAPTIVE_THRESH_GAUSSIAN_C, Imgproc.THRESH_BINARY_INV, 7, 2);
    Imgproc.cvtColor(src, src_gray, Imgproc.COLOR_GRAY2BGR);

    Mat canny_output = new Mat();
    Imgproc.Canny(src_gray, canny_output, thresh, thresh * 2);
//поиск границ
    List<MatOfPoint> contours = new ArrayList<MatOfPoint>();
//массив контуров
    Mat hierarchy = new Mat();
//иерархия контуров
    Imgproc.findContours(canny_output, contours, hierarchy, 0, 1);
//поиск контуров
    Mat res = Mat.zeros(canny_output.size(), CvType.CV_8UC3);
//итоговое изображение с контурами

    double[] green = {0, 255, 0};

    List<Point> all_contours = new ArrayList<Point>(); //массив
всех контуров
    for (MatOfPoint current_contour : contours) { //заполнение
массива
        all_contours.addAll(current_contour.toList());
    }

    for (Point p : all_contours) {
        res.put((int) p.y, (int) p.x, green);
        res.put((int) p.y - 1, (int) p.x, green);
        res.put((int) p.y - 1, (int) p.x - 1, green);
        res.put((int) p.y - 1, (int) p.x + 1, green);
        res.put((int) p.y + 1, (int) p.x, green);
        res.put((int) p.y + 1, (int) p.x - 1, green);
        res.put((int) p.y + 1, (int) p.x + 1, green);
        res.put((int) p.y, (int) p.x + 1, green);
        res.put((int) p.y, (int) p.x - 1, green); //прорисовка контуров
    }
}
```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.03-01 12				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```
MatOfPoint all_cont_mat = new MatOfPoint();
all_cont_mat.fromList(all_contours);

Rect boundingRect = Imgproc.boundingRect(all_cont_mat);
//прямоугольник вокруг контура
//Imgproc.rectangle(res, boundingRect, new Scalar(255, 0, 0));

List<MatOfPoint> hull_list = new ArrayList<>();
//выпуклая оболочка
MatOfPoint contour = all_cont_mat;
MatOfInt hull = new MatOfInt();
Imgproc.convexHull(contour, hull);
//поиск выпуклой оболочки

Point[] points_of_hull = new Point[hull.rows()];
Point[] array_of_contour = contour.toArray();
List<Integer> hull_ind = hull.toList();
for (int i = 0; i < hull_ind.size(); i++) {
    points_of_hull[i] = array_of_contour[hull_ind.get(i)];
}
hull_list.add(new MatOfPoint(points_of_hull));

for (int i = 0; i < hull_list.size(); i++) {
    Scalar color = new Scalar(0, 0, 255);
    Imgproc.drawContours(res, hull_list, i, color);
}
//прорисовка оболочки
to_draw = res;

Drawable drawable = convertMatToDrawable(to_draw);
AnswerLevel.answerImage.setImageDrawable(drawable);

System.out.println("ready Main");

//MatOfByte matOfByte = new MatOfByte();
//Imgcodecs.imencode(".jpg", res, matOfByte); //перевод
изображения в байтовую матрицу
//byte[] byteArray = matOfByte.toArray(); //перевод
байтовой матрицы в массив
//InputStream in = new ByteArrayInputStream(byteArray); //перевод в
буферизованное изображение
//BufferedImage bufImage = ImageIO.read(in);
//JFrame frame = new JFrame(); //создаем
окно
//frame.getContentPane().add(new JLabel(new ImageIcon(bufImage)));
//показываем изображение
MouseDemo md = new MouseDemo(AnswerLevel.answerImage, 400, 400);
//frame.pack();
//frame.setVisible(true);
}
};
```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.03-01 12				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

## 1.4 Код класса MouseDemo

```

2  package kursuch.project.smarterthanmelines;

import android.app.Activity;
import android.graphics.Matrix;
import android.view.MotionEvent;
import android.view.View;

import org.opencv.core.Mat;

import java.io.IOException;
//import javax.swing.*;
//import java.awt.*;

public class MouseDemo extends Activity implements View.OnTouchListener {
    float touchX;
    float touchY;
    boolean isTouchDown;
    boolean isTouchUp;
    int sceneWidth;
    int sceneHeight;

    public MouseDemo(View view, int sceneWidth, int sceneHeight) {
        view.setOnTouchListener(this);
        this.sceneWidth = sceneWidth;
        this.sceneHeight = sceneHeight;
    }

    @Override
    public boolean onTouch(View view, MotionEvent event)
    {
        synchronized (this){
            isTouchDown = false;
            isTouchUp = false;

            switch (event.getAction())
            {
                case MotionEvent.ACTION_DOWN:
                    //touchX = event.getX(); // * (1080 /
MainComputer.to_draw.rows()); // *
(AnswerLevel.answerImage.getDrawable().getIntrinsicWidth() /
PhotoLevel.oldWidth); // / view.getWidth() ;
                    //touchY = event.getY(); // * (1920 /
MainComputer.to_draw.cols()); // *
(AnswerLevel.answerImage.getDrawable().getIntrinsicHeight() /
PhotoLevel.oldHeight); // / view.getHeight();
                    ////touchX = view.getLeft();
                    ////touchY = view.getTop();
                    ////int[] posXY = new int[2];
                    ////view.getLocationOnScreen(posXY);
                    ////touchX = (int) event.getX();
                    ////touchY = (int) event.getY();
                    ////touchX = touchX - posXY[0]; // posXY[0] is the X
coordinate
                    ////touchY = touchY - posXY[1]; // posXY[1] is the y

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.03-01 12				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

coordinate
//isTouchDown = true;
//isTouchUp = false;
float x = event.getX();
float y = event.getY();
// The coordinates of the target point
float dst[] = new float[2];
// Get the matrix of ImageView
Matrix imageMatrix =
AnswerLevel.answerImage.getImageMatrix();
// Create an inverse matrix
Matrix inverseMatrix = new Matrix();
// Inverse, the inverse matrix is assigned
imageMatrix.invert(inverseMatrix);
// Get the value of the target point dst through the
inverse matrix mapping
inverseMatrix.mapPoints(dst, new float[]{x, y});
float dstX = dst[0];
float dstY = dst[1];
touchX = (int) (dstX);
touchY = (int) (dstY);
// Determine the position of dstX, dstY on the Bitmap

try {
    MainComputer.mouse_event((int) (touchX * 2.65),
(int) (touchY * 2.65));
} catch (IOException e) {
    e.printStackTrace();
}
break;
//case MotionEvent.ACTION_UP:
//    touchX = (int)event.getX() * sceneWidth;
//    touchY = event.getY() * sceneHeight;
//    isTouchDown = false;
//    isTouchUp = true;
//    break;
}
}
return true;
}

// public boolean getTouchUp(int x, int y, int touchWidth, int
touchHeight) {
//    if (isTouchUp) {
//        if (touchX >= x && touchX <= x + touchWidth - 1 && touchY <=
y && touchY >= y - (touchHeight - 1)) {
//            isTouchUp = false;
//            return true;
//        }
//    }
//    return false;
// }
}

@Override
//public void mouseClicked(MouseEvent e) {
//    x = e.getX();
//    y = e.getY();

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.03-01 12				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```
//      try {  
//          Main.mouse_event(x, y);  
//      } catch (IOException ioException) {  
//      }  
//}  
  
// MouseDemo(JFrame frame1) {  
//     //label = new JLabel(new ImageIcon("maze.png"));  
//     //label.setBounds(0,0,600,600);  
//     //label = new JLabel();  
//     //frame = new JFrame();  
//     frame = frame1;  
//     //frame.setSize(840, 840);  
//     //frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
//     //frame.setLayout(null);  
//     frame.setVisible(true);  
//     frame.addMouseListener(this);  
//     //frame.add(label);  
// }
```

## 2.1 Код класса Node

```
package kursuch.project.smarterthanmelines;  
  
public class Node {  
  
    int green = 0;  
    int blue = 0;  
    private boolean is_wall = false;  
    private boolean is_start = false;  
    private boolean is_end = false;  
    boolean close_wall = false;  
  
    public boolean isIs_way() {  
        return is_way;  
    }  
  
    public void setIs_way(boolean is_way) {  
        this.is_way = is_way;  
    }  
  
    public boolean is_way = false;  
  
    public Node(int green, int blue, boolean is_wall) {  
        this.green = green;  
        this.blue = blue;  
        this.is_wall = is_wall;  
    }  
  
    public boolean isIs_wall() {  
        return is_wall;  
    }  
  
    public void setIs_wall(boolean is_wall) {  
        this.is_wall = is_wall;  
    }  
}
```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.03-01 12				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```
}

public boolean isIs_start() {
    return is_start;
}

public void setIs_start(boolean is_start) {
    this.is_start = is_start;
}

public boolean isIs_end() {
    return is_end;
}

public void setIs_end(boolean is_end) {
    this.is_end = is_end;
}
}
```

## 2.2 Код класса Pair

```
package kursuch.project.smarterthanmelines;

public class Pair {
    int first;
    int second;

    public Pair(int first, int second) {
        this.first = first;
        this.second = second;
    }

    public void setFirst(int first) {
        this.first = first;
    }

    public void setSecond(int second) {
        this.second = second;
    }

    public int getFirst() {
        return first;
    }

    public int getSecond() {
        return second;
    }
}
```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.03-01 12				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата



### 2.3 Код класса PhotoLevel

```

package kursuch.project.smarterthanmelines;

import android.Manifest;
import android.app.AlertDialog;
import android.content.DialogInterface;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.graphics.Bitmap;
import android.os.Build;
import android.os.Bundle;
import android.provider.MediaStore;
import android.view.View;
import android.view.Window;
import android.view.WindowManager;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.Toast;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;

public class PhotoLevel extends AppCompatActivity {

    public static ImageView mImageView;
    Button mChooseBtn;
    Button back_btn;
    Button photo_btn;
    Button solution_btn;

    public static int oldWidth;
    public static int oldHeight;

    private static final int IMAGE_PICK_CODE = 1000;
    private static final int PERMISSION_CODE = 1001;
    private static final int CAMERA_REQUEST = 0;

    public static boolean isOpened = false;
    public static boolean photo = false;

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

        AlertDialog.Builder builder = new
        AlertDialog.Builder(PhotoLevel.this);
        builder.setTitle("Обратите внимание!")
            .setMessage("Перед тем как получить решение лабиринта, вам
следует выбрать лабиринт из галереи или сделать его фото " +
                "(кнопки 'Галерея'/'Камера'). Далее нажмите кнопку
'Ответ'." +
                " Кнопка 'Ответ' будет не активна пока вы не выберете
фото!")
            .setNegativeButton("ОК",

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.03-01 12				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

        new DialogInterface.OnClickListener() {
            public void onClick(DialogInterface dialog, int
id) {
                dialog.cancel();
                isOpened = true;
            }
        });
        AlertDialog alert = builder.create();

        AlertDialog.Builder builderPhoto = new
AlertDialog.Builder(PhotoLevel.this);
        builderPhoto.setTitle("Обратите внимание!")
            .setMessage("Чтобы добиться лучшего результата, фотографию
необходимо сделать в хорошо освещенном помещении или со вспышкой, " +
"также избегайте теней на фото и резких перепадов
света.")
            .setNegativeButton("OK",
        new DialogInterface.OnClickListener() {
            public void onClick(DialogInterface dialog, int
id) {
                dialog.cancel();
                photo = true;
            }
        });
        AlertDialog alertPhoto = builderPhoto.create();

        if(photo == false)
            alertPhoto.show();

        if(isOpened == false)
            alert.show();

        System.out.println("photolevel loaded");

        mImageView = findViewById(R.id.image_view);
        mChooseBtn = findViewById(R.id.galery_btn);

        mChooseBtn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.M) {
                    if
(checkSelfPermission(Manifest.permission.READ_EXTERNAL_STORAGE) ==
PackageManager.PERMISSION_DENIED) {
                        String[] permissions =
{Manifest.permission.READ_EXTERNAL_STORAGE};
                        requestPermissions(permissions, PERMISSION_CODE);
                    } else {
                        pickImageFromGallery();
                    }
                } else {
                    pickImageFromGallery();
                }
            }
        });

        Window w = getWindow();
        w.setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN,

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.03-01 12				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

WindowManager.LayoutParams.FLAG_FULLSCREEN);

solution_btn = findViewById(R.id.solution_btn);
solution_btn.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        try {
            if (mImageView.getDrawable() != null) {
                oldHeight =
mImageView.getDrawable().getIntrinsicHeight();
                oldWidth =
mImageView.getDrawable().getIntrinsicWidth();
                Intent intent = new Intent(PhotoLevel.this,
AnswerLevel.class);

                startActivity(intent);
                finish();
                System.out.println("go to answer level");
            } else {
                alert.show();
                System.out.println("answer level error");
            }
        } catch (Exception e) {
            System.out.println("answer level error");
        }
    }
});

back_btn = findViewById(R.id.back_btn);
back_btn.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        try {
            Intent intent = new Intent(PhotoLevel.this,
MainActivity.class);
            startActivity(intent);
            finish();
            System.out.println("back");
        } catch (Exception e) {

        }
    }
});

photo_btn = findViewById(R.id.photo_btn);
photo_btn.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        takePicture();
        System.out.println("photoTaked");
    }
});

private void takePicture() {
    Intent cameraIntent = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
    startActivityForResult(cameraIntent, CAMERA_REQUEST);
    System.out.println("photoTaked228");
}

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.03-01 12				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

@Override
public void onBackPressed() {
    try {
        Intent intent = new Intent(PhotoLevel.this, MainActivity.class);
        startActivity(intent);
        finish();
        System.out.println("back");
    } catch (Exception e) {
    }
}

private void pickImageFromGallery() {
    Intent intent = new Intent(Intent.ACTION_PICK);
    intent.setType("image/*");
    startActivityForResult(intent, IMAGE_PICK_CODE);
    System.out.println("selected");
}

@Override
public void onRequestPermissionsResult(int requestCode, @NonNull String[]
permissions, @NonNull int[] grantResults) {
    switch (requestCode) {
        case PERMISSION_CODE: {
            if (grantResults.length > 0 && grantResults[0] ==
PackageManager.PERMISSION_GRANTED) {
                pickImageFromGallery();
            } else {
                Toast.makeText(this, "Permission denied. . . !",
Toast.LENGTH_SHORT).show();
            }
        }
    }
}

@Override
protected void onActivityResult(int requestCode, int resultCode, Intent
data) {
    super.onActivityResult(requestCode, resultCode, data);
    if (resultCode == RESULT_OK && requestCode == IMAGE_PICK_CODE) {
        mImageView.setImageURI(data.getData());
        System.out.println("Photo in imageView");
    }
    if (requestCode == CAMERA_REQUEST && resultCode == RESULT_OK) {
        // Фотка сделана, извлекаем картинку
        Bitmap thumbnailBitmap = (Bitmap) data.getExtras().get("data");
        mImageView.setImageBitmap(thumbnailBitmap);
    }
}
}

```

XML

```

<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.03-01 12				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```

android:layout_height="match_parent"
android:orientation="vertical">

<ImageView
    android:id="@+id/photo_background"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:scaleType="centerCrop"
    app:srcCompat="@drawable/menu_background" />

<LinearLayout
    android:id="@+id/container1"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:gravity="center"
    android:orientation="vertical">

    <LinearLayout
        android:id="@+id/back_container"
        android:layout_width="wrap_content"
        android:layout_height="123dp"
        android:layout_marginStart="-135dp"
        android:orientation="horizontal">

        <Button
            android:id="@+id/back_btn"
            android:layout_width="0dp"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:background="@drawable/btn_stroke_white30_press_white"
            android:paddingLeft="10dp"
            android:paddingRight="10dp"
            android:text="@string/back"
            android:textAllCaps="false"
            android:textColor="@color/black"
            android:textSize="24sp" />

    </LinearLayout>

    <ImageView
        android:id="@+id/image_view"
        android:layout_width="400dp"
        android:layout_height="400dp" />

    <LinearLayout
        android:id="@+id/threebar_btn"
        android:layout_width="394dp"
        android:layout_height="84dp"
        android:layout_marginTop="20dp"
        android:orientation="horizontal">

        <Button
            android:id="@+id/galery_btn"
            android:layout_width="0dp"
            android:layout_height="wrap_content"
            android:layout_marginTop="35dp"
            android:layout_marginRight="10dp"
            android:layout_weight="1"
            android:background="@drawable/btn_stroke_white30_press_white"

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.03-01 12				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```
        android:paddingLeft="10dp"
        android:paddingRight="10dp"
        android:text="@string/gallery"
        android:textColor="@color/black"
        android:textSize="24sp" />

<Button
    android:id="@+id/photo_btn"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginTop="35dp"
    android:layout_marginRight="10dp"
    android:layout_weight="1"
    android:background="@drawable/btn_stroke_white30_press_white"
    android:paddingLeft="10dp"
    android:paddingRight="10dp"
    android:text="@string/camera"
    android:textColor="@color/black"
    android:textSize="24sp" />

<Button
    android:id="@+id/solution_btn"
    android:layout_width="7dp"
    android:layout_height="wrap_content"
    android:layout_marginTop="35dp"
    android:layout_weight="1"
    android:background="@drawable/btn_stroke_white30_press_white"
    android:paddingLeft="10dp"
    android:paddingRight="10dp"
    android:text="@string/answer"
    android:textColor="@color/black"
    android:textSize="24sp" />
</LinearLayout>
</LinearLayout>
</RelativeLayout>
```

## ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ

Лист регистрации изменений

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.03-01 12				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

[illegible]

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.03-01 12				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата