Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**ИРКУТСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

Институт информационных технологий и анализа данных

наименование института

ОТЧЕТ  
к лабораторной работе по дисциплине

Объектно-ориентированное программирование

«Разработка приложения с оконным графическим   
пользовательским интерфейсом»

Наименование темы

Выполнил студент группы ИСМб 19–1 Малиновцев И.А.

шифр Фамилия И.О.

Проверила Маланова Т.В.

Фамилия И.О.

Содержание отчета на 18 стр.

Иркутск 2021 г.

Содержание

[1 Введение 3](#_Toc68340577)

[2 Проектирование графического интерфейса пользователя 4](#_Toc68340578)

[3 Проектирование классов 5](#_Toc68340579)

[3.1 Словесное описание классов 5](#_Toc68340580)

[3.2 Класс MainActivity 5](#_Toc68340581)

[3.3 Класс DrawingView 5](#_Toc68340582)

[4 Таблица тестов 6](#_Toc68340583)

[5 Результаты тестирования 7](#_Toc68340584)

[6 Исходный код 12](#_Toc68340585)

[6.1 Класс MainActivity 12](#_Toc68340586)

[6.2 Класс DrawingView 15](#_Toc68340587)

[Список литературы 18](#_Toc68340588)

1 Введение

**Цель работы**

Получение общего представления о создании Android-приложения на Java

Получение практического опыта работы с базовыми графическими библиотеками java-android.

**Варианты индивидуальных заданий**

В качестве индивидуального задания была выбрана реализация приложения для рисования

2 Проектирование графического интерфейса пользователя

Графический интерфейс состоит из одного окна. На нем находятся 16 кнопок и View, который был разработан для этого приложения

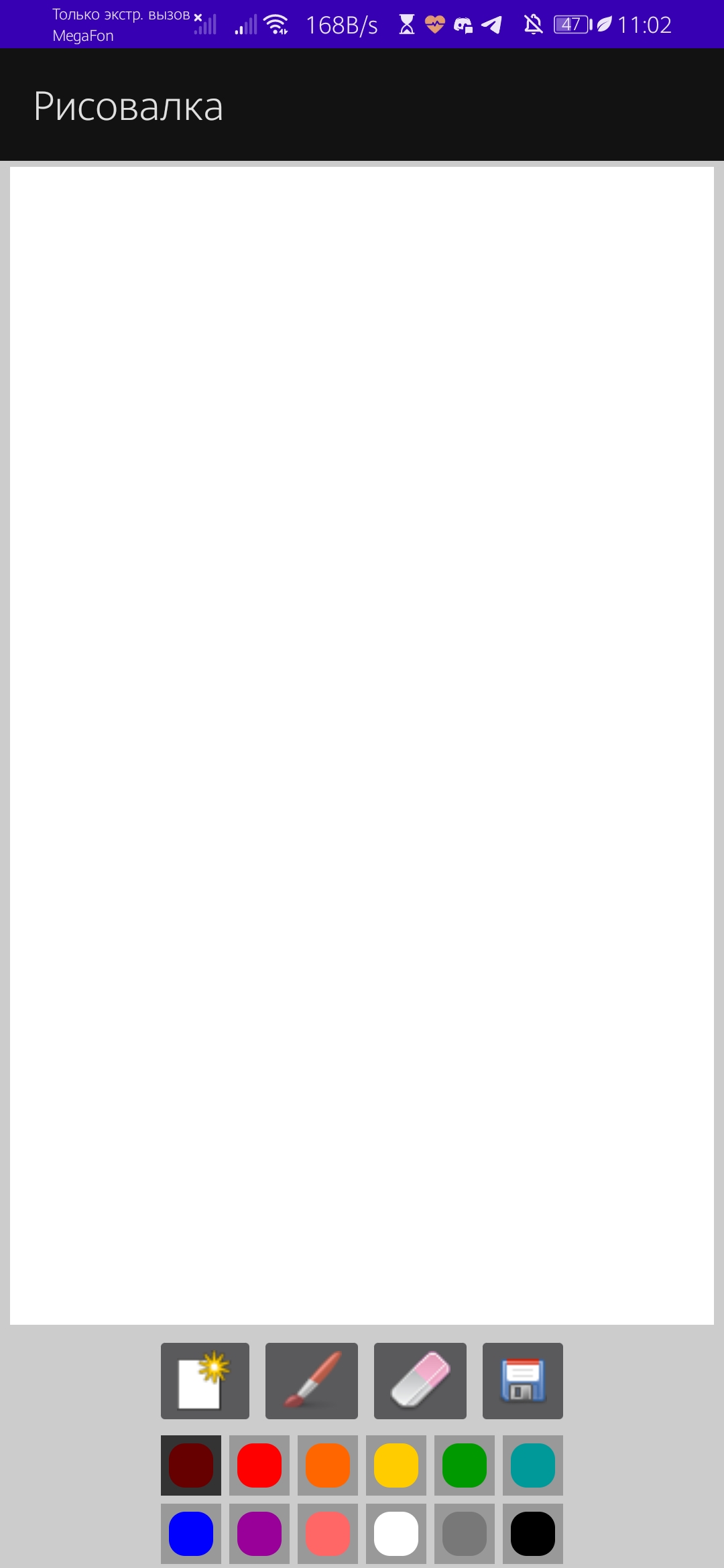


Рисунок 1 – Вид главного окна

3 Проектирование классов

3.1 Словесное описание классов

Программа состоит из 2 классов, один из которых MainActivity– точка входа в программу.

Описание классов:

1. MainActivity– точка входа в программу, запуск графического интерфейса.
2. DrawingView– кастомный View для рисования

3.2 Класс MainActivity

Таблица 3.1 – Таблица методов класса MainActivity

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Имя** | **Тип** | **Статический** | **Уровень доступа** | **Аргументы** | **Назначение** |
| onCreate | void | - | protected | Bundle savedInstanceState | Инициализация объектов в окне |
| paintClicked | void | - | public | View view | Метод для выбора цвета |
| onClick | void | - | public | View view | Метод для клике на кнопки |

3.3 Класс DrawingView

Таблица 3.2 – Таблица методов класса 2. DrawingView

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Имя** | **Тип** | **Статический** | **Уровень доступа** | **Аргументы** | **Назначение** |
| onDraw | void | - | protected | Canvas canvas | Рисование на View |
| setupDrawing | Void | - | Public | - | Установка значений полей для рисования |
| onSizeChanged | Void | - | protected | int w, int h, int oldw, int oldh | Метод для инициализации размеров View |
| onTouchEvent | boolean | - | public | MotionEvent event | Метод для отслеживания касания по форме |
| setColor | Void | - | Public | String newColor | Метод для изменения цвета |
| setBrushSize | Void | - | Public | float newSize | Метод для изменения размера кисти |
| setErase | Void | - | Public | boolean isErase | Метод для стирания |
| startNew | Void | - | Public | - | Метод для очистки окна |

4 Таблица тестов

Таблица 4.1 – таблица тестов

|  |  |  |
| --- | --- | --- |
| **№** | **Действие** | **Результат** |
| 1 | Нажатие на кнопку изменения размера и выбор нового | Изменение размера кисти |
| 2 | Нажатие на кнопку «Ластик» и стирание элементов | Стирание с холста |
| 3 | Нажатие на кнопку «Новый» | Очистка холста |
| 4 | Нажатие на кнопку «Сохранить» | Сохранение изображения в галерею |
| 5 | Выбор цвета из палитры | Изменение цвета кисти |

5 Результаты тестирования

Результаты тестирования представлены на рисунках 2 – 6.

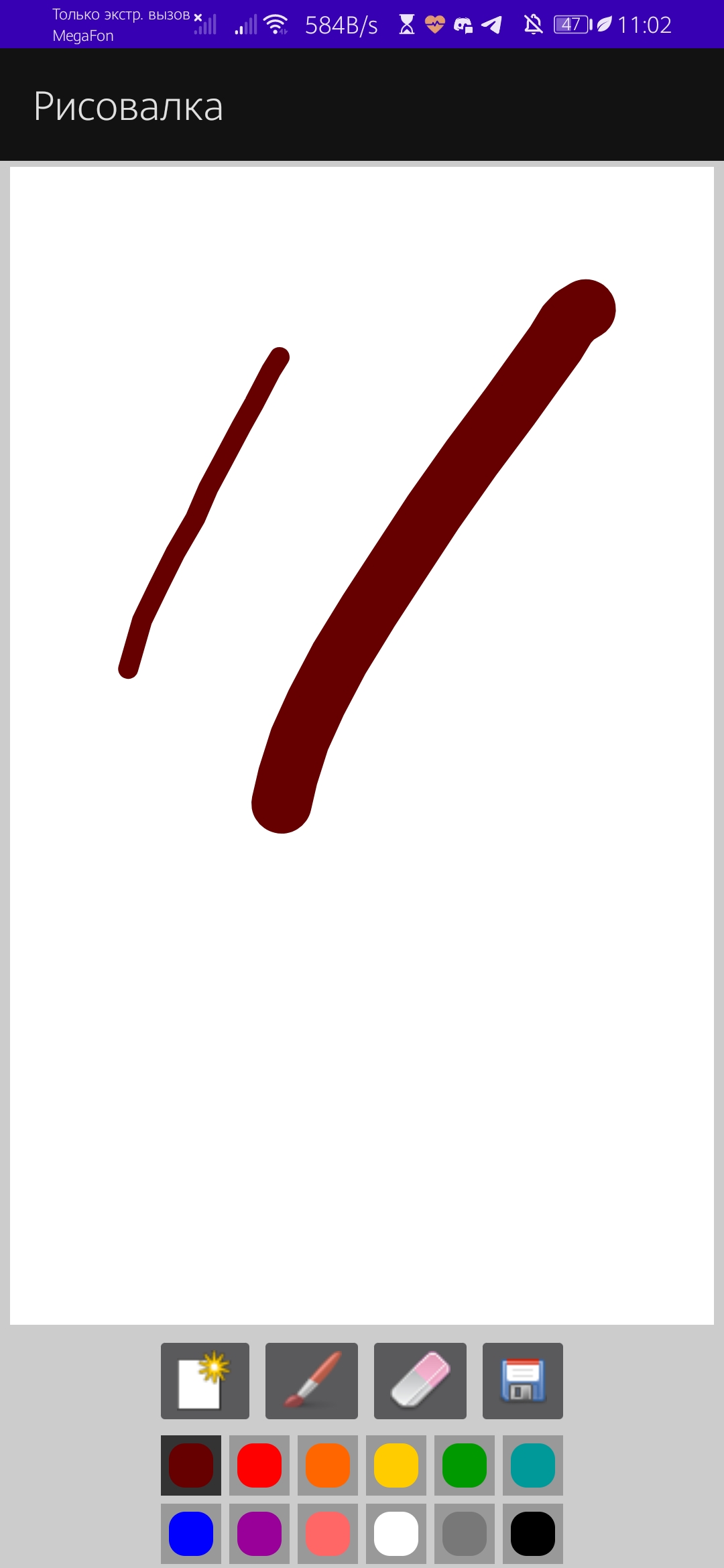


Рисунок 2 – Результат выполнения теста 1

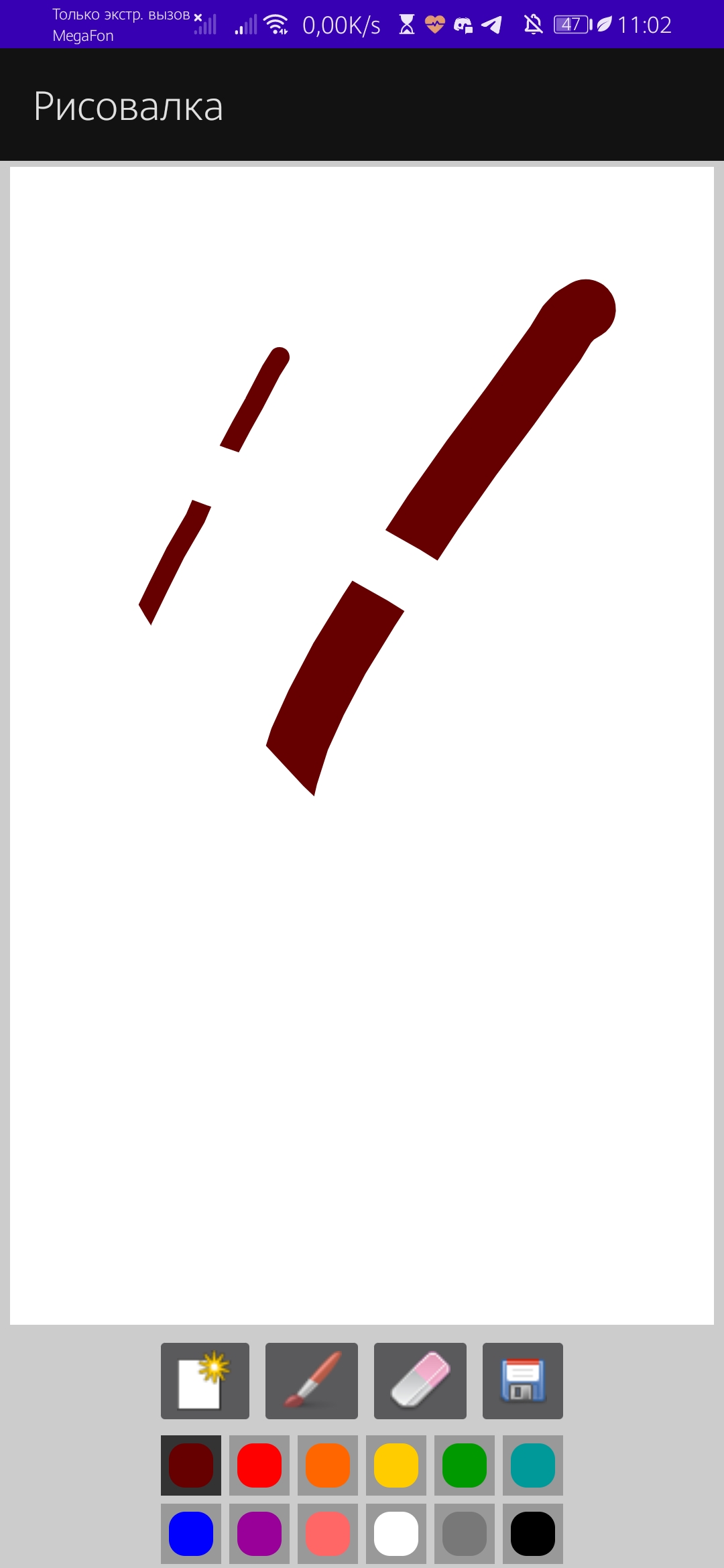


Рисунок 3 – Результат выполнения теста 2

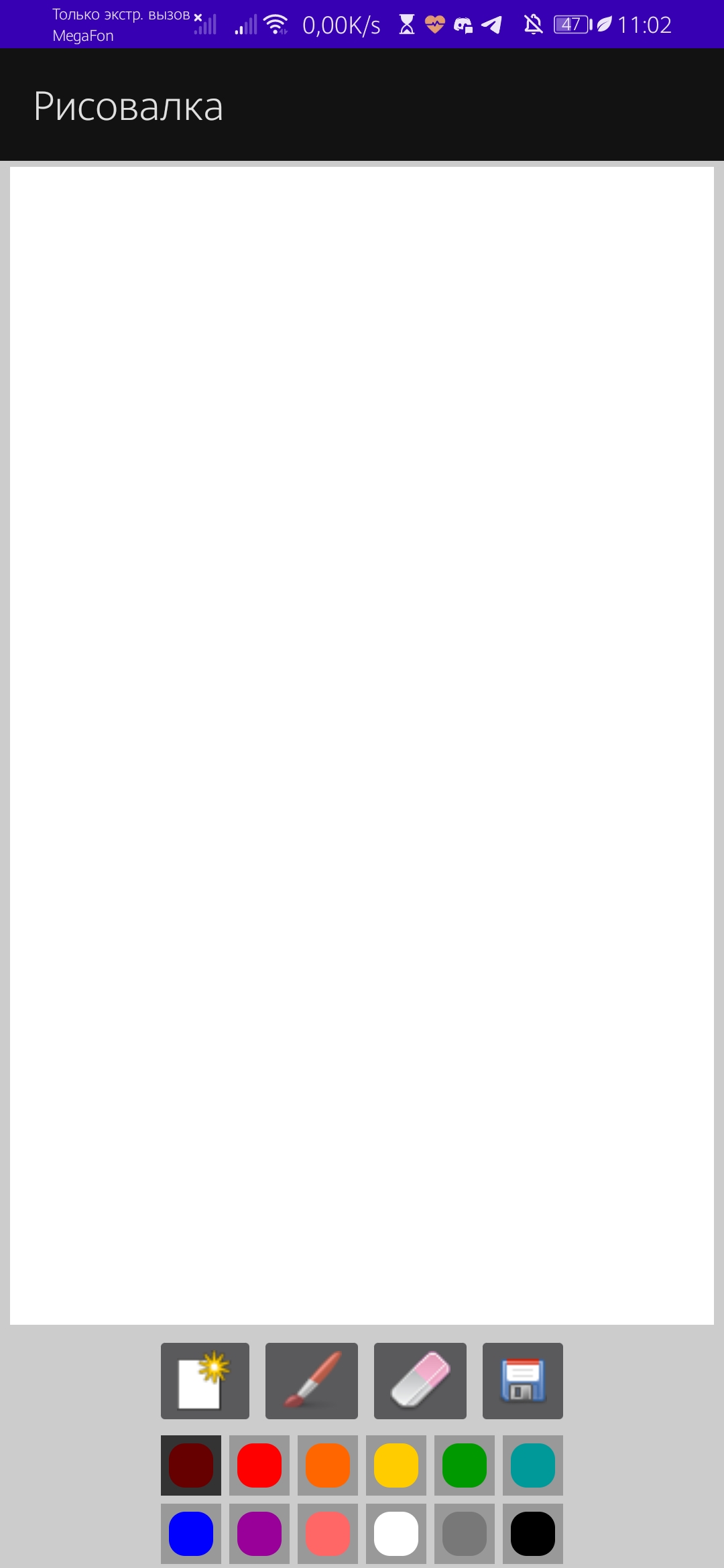


Рисунок 4 – Результат выполнения теста 3

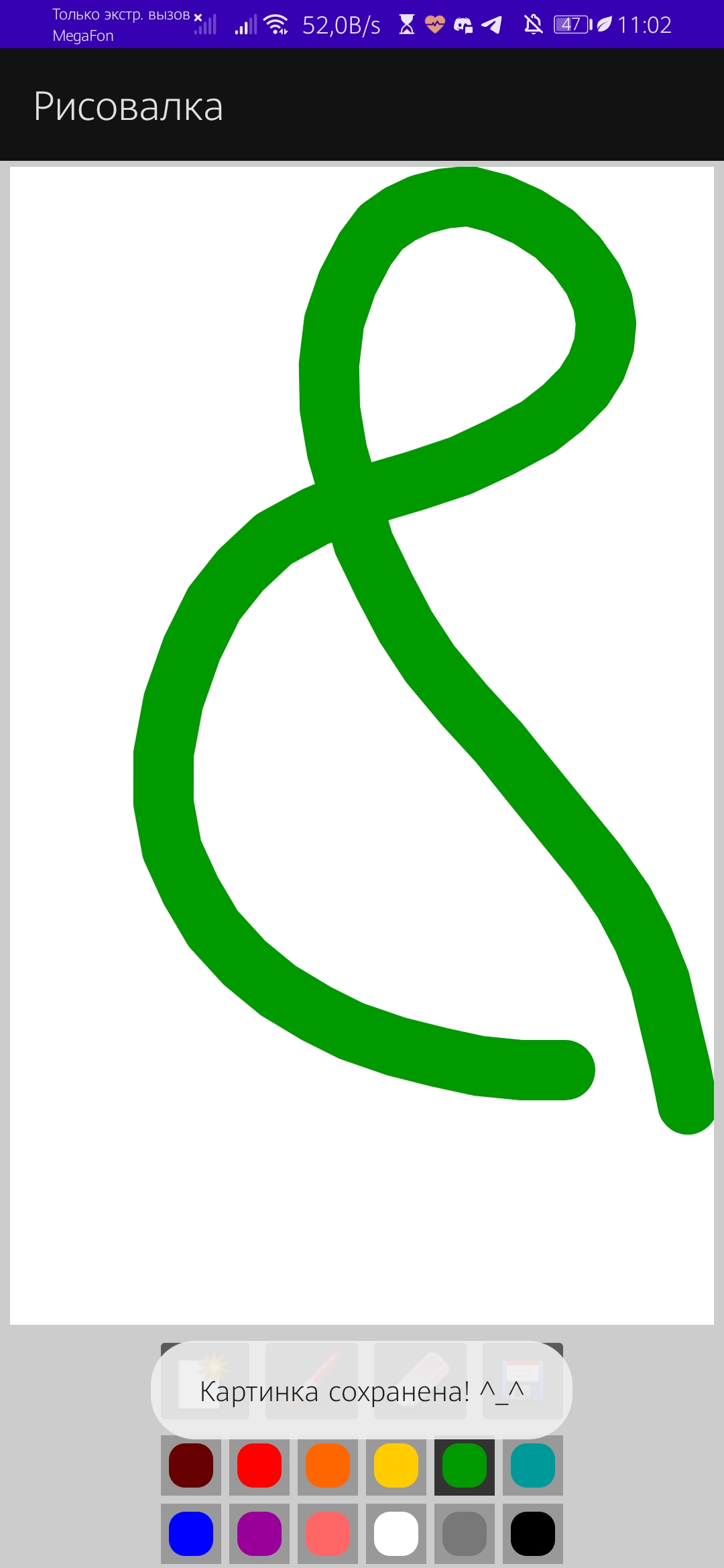


Рисунок 5 – Результат выполнения теста 4

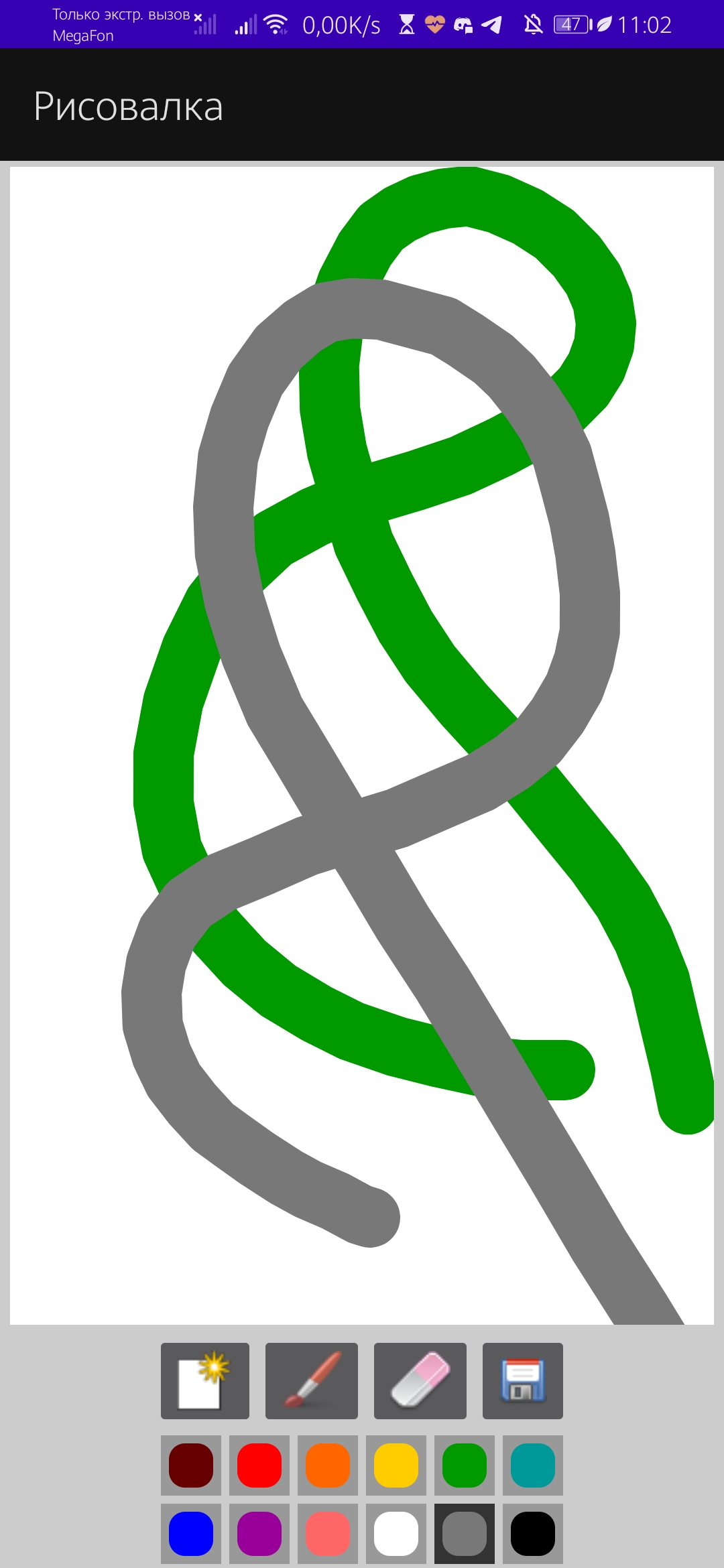


Рисунок 6 – Результат выполнения теста 5

6 Исходный код

6.1 Класс MainActivity

public class MainActivity extends AppCompatActivity implements OnClickListener {

//Создаем переменную класса

private DrawingView drawView;

//Создаем все кнопки : кнопка цвета, кнопка кисти, стерка, новый, сохранить

private ImageButton currPaint, drawBtn, eraseBtn, newBtn, saveBtn;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

//Передать id для рисования

drawView = (DrawingView)findViewById(R.id.drawing);

//Лейаут, отвещающий за кнопки цвета

LinearLayout paintLayout = (LinearLayout)findViewById(R.id.paint\_colors);

//Найти первую кнопку

currPaint = (ImageButton)paintLayout.getChildAt(0);

//Показать что выбрана текущая кнопка

currPaint.setImageDrawable(getResources().getDrawable(R.drawable.paint\_pressed));

//Установить значения из ресурсов

smallBrush = getResources().getInteger(R.integer.small\_size);

mediumBrush = getResources().getInteger(R.integer.medium\_size);

largeBrush = getResources().getInteger(R.integer.large\_size);

//Создать кнопку

drawBtn = (ImageButton)findViewById(R.id.draw\_btn);

//Добавить кнопку

drawBtn.setOnClickListener(this);

//Установить средний размер по умолчанию

drawView.setBrushSize(mediumBrush);

//Стерка

eraseBtn = (ImageButton)findViewById(R.id.erase\_btn);

eraseBtn.setOnClickListener(this);

//Новый

newBtn = (ImageButton)findViewById(R.id.new\_btn);

newBtn.setOnClickListener(this);

//Сохранить

saveBtn = (ImageButton)findViewById(R.id.save\_btn);

saveBtn.setOnClickListener(this);

}

//Размеры кнопок(кисти)

private float smallBrush, mediumBrush, largeBrush;

//Тут мы выбираем цвета

public void paintClicked(View view){

//Вытащить размер кисти до стирания

drawView.setBrushSize(drawView.getLastBrushSize());

drawView.setErase(false);

if(view!=currPaint){

//Найти тег кнопки

ImageButton imgView = (ImageButton)view;

String color = view.getTag().toString();

//Передать цвет кноки

drawView.setColor(color);

//Отобразить текущую кнопку

imgView.setImageDrawable(getResources().getDrawable(R.drawable.paint\_pressed));

currPaint.setImageDrawable(getResources().getDrawable(R.drawable.paint));

currPaint=(ImageButton)view;

}

}

@Override

public void onClick(View view) {

//Если выбрали рисовать

if(view.getId()==R.id.draw\_btn){

//Меню выбора

final Dialog brushDialog = new Dialog(this);

brushDialog.setContentView(R.layout.brush\_chooser);

//Создать кнопки и дбавить слушателей

ImageButton smallBtn = (ImageButton)brushDialog.findViewById(R.id.small\_brush);

smallBtn.setOnClickListener(v -> {

//Считать все размеры

drawView.setBrushSize(smallBrush);

drawView.setLastBrushSize(smallBrush);

//Показать что рисуем

drawView.setErase(false);

brushDialog.dismiss();

});

ImageButton mediumBtn = (ImageButton)brushDialog.findViewById(R.id.medium\_brush);

mediumBtn.setOnClickListener(v -> {

drawView.setBrushSize(mediumBrush);

drawView.setLastBrushSize(mediumBrush);

drawView.setErase(false);

brushDialog.dismiss();

});

ImageButton largeBtn = (ImageButton)brushDialog.findViewById(R.id.large\_brush);

largeBtn.setOnClickListener(v -> {

drawView.setBrushSize(largeBrush);

drawView.setLastBrushSize(largeBrush);

drawView.setErase(false);

brushDialog.dismiss();

});

//Отобразить выбор

brushDialog.show();

}

//Если выбрана стерка

else if(view.getId()==R.id.erase\_btn){

//Отобразить меню выбора

final Dialog brushDialog = new Dialog(this);

brushDialog.setContentView(R.layout.brush\_chooser);

//Тоже самое чтои выше, только для стерки

ImageButton smallBtn = (ImageButton)brushDialog.findViewById(R.id.small\_brush);

smallBtn.setOnClickListener(v -> {

//Передать что стираем

drawView.setErase(true);

//Передать что малым

drawView.setBrushSize(smallBrush);

brushDialog.dismiss();

});

ImageButton mediumBtn = (ImageButton)brushDialog.findViewById(R.id.medium\_brush);

mediumBtn.setOnClickListener(v -> {

drawView.setErase(true);

drawView.setBrushSize(mediumBrush);

brushDialog.dismiss();

});

ImageButton largeBtn = (ImageButton)brushDialog.findViewById(R.id.large\_brush);

largeBtn.setOnClickListener(v -> {

drawView.setErase(true);

drawView.setBrushSize(largeBrush);

brushDialog.dismiss();

});

brushDialog.show();

}

//Если новый

else if(view.getId()==R.id.new\_btn){

//Спросить хочет ли пользователь начать новый

AlertDialog.Builder newDialog = new AlertDialog.Builder(this);

newDialog.setTitle("Новый рисунок");

newDialog.setMessage("Хотите начать новый рисунок(текущий будет утерян)?");

//Если да

newDialog.setPositiveButton("Да", (dialog, which) -> {

drawView.startNew();

dialog.dismiss();

});

//Если нет

newDialog.setNegativeButton("Закрыть", (dialog, which) -> dialog.cancel());

newDialog.show();

}

//Сохранялся

else if(view.getId()==R.id.save\_btn){

AlertDialog.Builder saveDialog = new AlertDialog.Builder(this);

//Хотим ли сохранять?

saveDialog.setTitle("Сохранение рисунка");

saveDialog.setMessage("Хотите сохранить рисунок в галерее?");

//Если да

saveDialog.setPositiveButton("Да", (dialog, which) -> {

//Кеш рисунка

drawView.setDrawingCacheEnabled(true);

//Сохраняем(путь)

String imgSaved = MediaStore.Images.Media.insertImage(

getContentResolver(), drawView.getDrawingCache(),

UUID.randomUUID().toString()+".png", "drawing");

//Пост сообщение

if(imgSaved!=null){

Toast savedToast = Toast.makeText(getApplicationContext(),

"Картинка сохранена! ^\_^", Toast.LENGTH\_SHORT);

savedToast.show();

}

else{

Toast unsavedToast = Toast.makeText(getApplicationContext(),

"Что-то пошло не так Q\_Q", Toast.LENGTH\_SHORT);

unsavedToast.show();

}

//Уничтожить кеш рисунка

drawView.destroyDrawingCache();

});

//Если не хотим сохранять

saveDialog.setNegativeButton("Закрыть", (dialog, which) -> dialog.cancel());

saveDialog.show();

}

}

}

6.2 Класс DrawingView

public class DrawingView extends View {

//Стираем ли?

private boolean erase=false;

//Рисуен путь(линии рисования)

private Path drawPath;

//Рисуем и отображаем

private Paint drawPaint, canvasPaint;

//Устанавливаем цвет

private int paintColor = 0xFF660000;

//То, где рисуем

private Canvas drawCanvas;

//Что рисуем

private Bitmap canvasBitmap;

//Текущий размер, последний

private float brushSize, lastBrushSize;

//Перерисовка

@Override

protected void onDraw(Canvas canvas) {

super.onDraw(canvas);

//Холст

canvas.drawBitmap(canvasBitmap, 0, 0, canvasPaint);

//ПУти рисования

canvas.drawPath(drawPath, drawPaint);

}

//Конструктор для инициализации

public DrawingView(Context context) {

super(context);

}

public DrawingView(Context context, AttributeSet attrs){

super(context, attrs);

setupDrawing();

}

//Установка рисовальщина

public void setupDrawing(){

//Инициализация объектов

drawPath = new Path();

drawPaint = new Paint();

//Установили первоначальный цвет

drawPaint.setColor(paintColor);

//Все это отвечает за размеры кисти

drawPaint.setAntiAlias(true);

drawPaint.setStyle(Paint.Style.STROKE);

drawPaint.setStrokeJoin(Paint.Join.ROUND);

drawPaint.setStrokeCap(Paint.Cap.ROUND);

//Нарисовали

canvasPaint = new Paint(Paint.DITHER\_FLAG);

//Установить новый размер

drawPaint.setStrokeWidth(brushSize);

//Сохранить старый

lastBrushSize = brushSize;

}

//Вызывается, когда моему View присвается размер

@Override

protected void onSizeChanged(int w, int h, int oldw, int oldh) {

//Отправка рахмеров в супер-класс

super.onSizeChanged(w, h, oldw, oldh);

//Создать область рисования

canvasBitmap = Bitmap.createBitmap(w, h, Bitmap.Config.ARGB\_8888);

//Добавить эту область на канвас

drawCanvas = new Canvas(canvasBitmap);

}

//Вызывается при касании экрана

@Override

public boolean onTouchEvent(MotionEvent event) {

//Текущие ккординаты касания

float touchX = event.getX();

float touchY = event.getY();

switch (event.getAction()) {

//При нажатии на экран

case MotionEvent.ACTION\_DOWN:

//Начинаем рисовку

drawPath.moveTo(touchX, touchY);

break;

//При движении по жкрану

case MotionEvent.ACTION\_MOVE:

//Рисуем

drawPath.lineTo(touchX, touchY);

break;

//При прекращении касания

case MotionEvent.ACTION\_UP:

//Прорисовать путь до конца

drawCanvas.drawPath(drawPath, drawPaint);

//Очистить путь

drawPath.reset();

break;

//Если ничего вернуть фалс

default:

return false;

}

//Вызывает OnDraw

invalidate();

//Все хорошо

return true;

}

//Выбор цвета

public void setColor(String newColor){

//Вызвать onDraw

invalidate();

//Переписывает цвет

paintColor = Color.parseColor(newColor);

//Устанливаем цвет

drawPaint.setColor(paintColor);

}

//Установить размер кисти

public void setBrushSize(float newSize){

//Обновляем цвета

float pixelAmount = TypedValue.applyDimension(TypedValue.COMPLEX\_UNIT\_DIP,

newSize, getResources().getDisplayMetrics());

brushSize=pixelAmount;

drawPaint.setStrokeWidth(brushSize);

}

//Геттеры

public void setLastBrushSize(float lastSize){

lastBrushSize=lastSize;

}

public float getLastBrushSize(){

return lastBrushSize;

}

//Простой сетер

public void setErase(boolean isErase){

//Присвоить значение

erase=isErase;

//Переключаемся на стирание

if(erase) drawPaint.setXfermode(new PorterDuffXfermode(PorterDuff.Mode.CLEAR));

//Либо на рисование

else drawPaint.setXfermode(null);

}

//новый

public void startNew(){

//Очисть окно

drawCanvas.drawColor(0, PorterDuff.Mode.CLEAR);

//Вызвать onDraw

invalidate();

}

}

Список литературы

1. Java. Экспресс-курс [электронный ресурс] // Сайт Александра Климова [сайт], URL: http://developer.alexanderklimov.ru/android/java/java.php

2. API Specification for the Java 7 SE. [официальный сайт] URL: http://docs.oracle.com/javase/7/docs/api/

3. The Java Tutorials. SE [электронный ресурс], URL: http://docs.oracle.com/javase/tutorial/index.html

4. Гради Буч, Роберт А. Максимчук, Майкл У. Энгл, Бобби Дж. Янг, Джим Коналлен, Келли А. Хьюстон. Объектно-ориентированны анализ и проек-тирование с примерами приложений. Третье издание. М.: "Вильямс", 2010.

5. Хабибуллин И.Ш. Java 7: для программистов / И. Ш. Хабибуллин. – Санкт-Петербург : БХВ–Петербург, 2014.