**Министерство науки и высшего образования Российской Федерации**

Федеральное государственное автономное образовательное учреждение

высшего образования «Уральский федеральный университет

имени первого Президента России Б.Н. Ельцина»

Институт радиоэлектроники и информационных технологий - РТФ

Центр ускоренного обучения

**ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ 25**

по дисциплине «Разработка мобильных приложений»

**Тема: Unit 4 Настройки «Общие настройки»**

Студент группы РИЗ-300028у: А.С. Севостьянов

Преподаватель: С.И.Тимошенко,

доц., к.т.н.

**Екатеринбург 2022**

содержание

[1 Постановка задачи 3](#_Toc118041527)

[2 Результаты проделанной работы 4](#_Toc118041528)

[3 Анализ полученных результатов 5](#_Toc118041529)

## 1 Постановка задачи

Обновите приложение, чтобы оно могло сохранять, извлекать и сбрасывать общие настройки.

.

## Результаты проделанной работы

Программный код проекта, файл MainActivity.

package com.example.android.hellosharedprefs;  
  
import android.content.SharedPreferences;  
import android.graphics.drawable.ColorDrawable;  
import android.support.v4.content.ContextCompat;  
import android.support.v7.app.AppCompatActivity;  
import android.os.Bundle;  
import android.view.View;  
import android.widget.TextView;  
  
*/\*\*  
 \* \* Hello SharedPrefs - это адаптация приложения HelloToast из главы 1.  
 \* Он включает в себя:  
 \* - Кнопки для изменения цвета фона.  
 \* - Поддержание состояния экземпляра.  
 \* - Темы и стили.  
 \* - Считывание и запись общих настроек для текущего количества и цвета.  
 \*/*public class MainActivity extends AppCompatActivity {  
 // Текущее количество  
 private int mCount = 0;  
 // Текущий цвет фона  
 private int mColor;  
 // Текстовый вид для отображения как количества, так и цвета  
 private TextView mShowCountTextView;  
  
 // Клавиша для текущего подсчета  
 private final String COUNT\_KEY = "count";  
 // Клавиша для текущего цвета  
 private final String COLOR\_KEY = "color";  
  
 // Объект общих предпочтений  
 private SharedPreferences mPreferences;  
  
 // Имя файла общих настроек  
 private String sharedPrefFile =  
 "com.example.android.hellosharedprefs";  
  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
  
 // Инициализировать виды, цвет, предпочтения  
 mShowCountTextView = findViewById(R.id.*count\_textview*);  
 mColor = ContextCompat.*getColor*(this,  
 R.color.*default\_background*);  
  
 mPreferences = getSharedPreferences(sharedPrefFile, *MODE\_PRIVATE*);  
  
 // Восстановить настройки  
 mCount = mPreferences.getInt(COUNT\_KEY, 0);  
 mShowCountTextView.setText(String.*format*("%s", mCount));  
 mColor = mPreferences.getInt(COLOR\_KEY, mColor);  
 mShowCountTextView.setBackgroundColor(mColor);  
 }  
  
 */\*\*  
 \* Обрабатывает onClick для кнопок цвета фона. Получает предысторию  
 \* цвет кнопки, которая была нажата, и задает фон TextView  
 \* к этому цвету.  
 \*/* public void changeBackground(View view) {  
 int color = ((ColorDrawable) view.getBackground()).getColor();  
 mShowCountTextView.setBackgroundColor(color);  
 mColor = color;  
 }  
  
 */\*\*  
 \* Обрабатывает повторный щелчок для кнопки подсчета. Увеличивает значение глобального параметра  
 \* mCount и обновляет текстовое представление.  
 \*/* public void countUp(View view) {  
 mCount++;  
 mShowCountTextView.setText(String.*format*("%s", mCount));  
 }  
  
 */\*\*  
 \* Обрабатывает повторный щелчок кнопки сброса. Сбрасывает глобальное количество и  
 \* фоновые переменные на значения по умолчанию и сбрасывает представления на эти  
 \* значения по умолчанию.tes TextView.  
 \*/* public void reset(View view) {  
 // Сброс количества  
 mCount = 0;  
 mShowCountTextView.setText(String.*format*("%s", mCount));  
  
 // сброс цвета  
 mColor = ContextCompat.*getColor*(this,  
 R.color.*default\_background*);  
 mShowCountTextView.setBackgroundColor(mColor);  
  
 // Четкие предпочтения  
 SharedPreferences.Editor preferencesEditor = mPreferences.edit();  
 preferencesEditor.clear();  
 preferencesEditor.apply();  
 }  
  
 */\*\*  
 \* Обратный вызов для приостановки активности. Общие настройки сохраняются здесь.  
 \*/* @Override  
 protected void onPause() {  
 super.onPause();  
  
 SharedPreferences.Editor preferencesEditor = mPreferences.edit();  
 preferencesEditor.putInt(COUNT\_KEY, mCount);  
 preferencesEditor.putInt(COLOR\_KEY, mColor);  
 preferencesEditor.apply();  
 }  
}

На рисунке 1 показано первоначальное состояние программы.

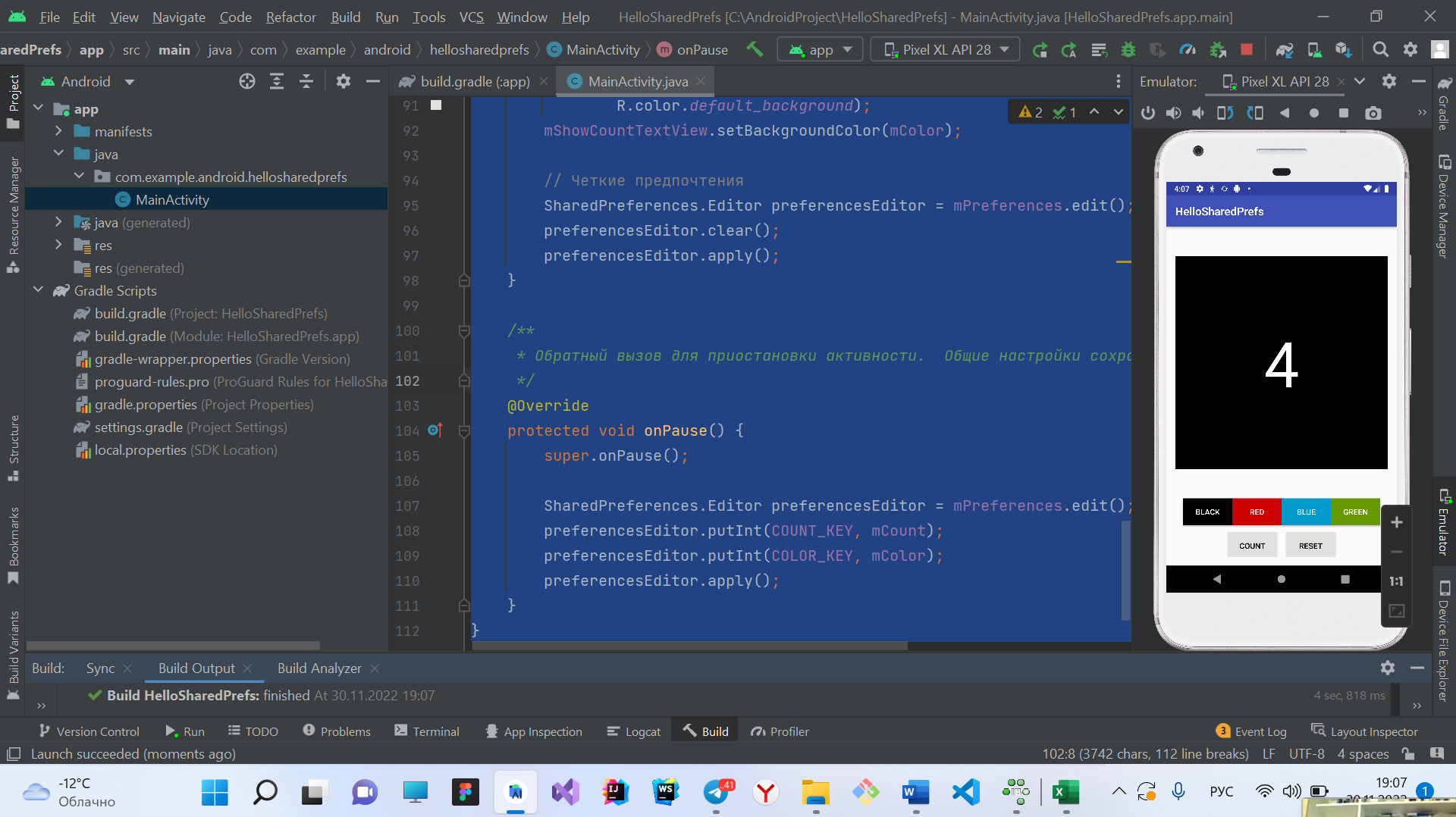


Рисунок 1 – Первоначальное состояние программы

На рисунке 2 показан результат работы программы.

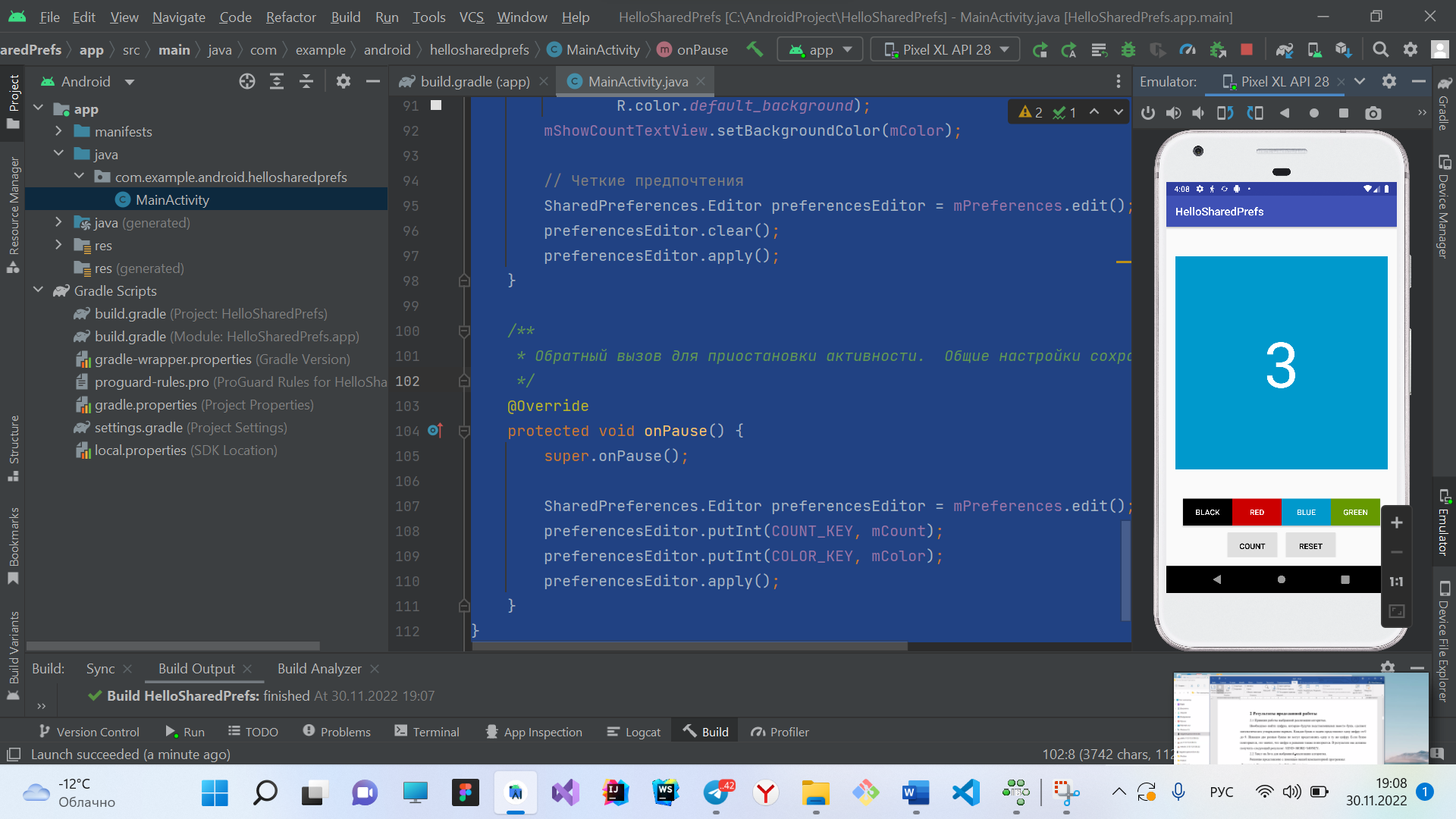


Рисунок 2 – Результат работы программы

## **Анализ полученных результатов**

В результате выполнения данной лабораторной работы мной был разработан проект, в котором я научился определять, что такое общие предпочтения. Как создать общий файл настроек для вашего приложения.

Как сохранить данные в общих настройках и снова прочитать эти настройки. Как очистить данные в общих настройках.

.