**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 26**

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

Создать приложение, которое включает **настройки** в меню параметров. Добавить переключатель **параметров** настройки. Добавить код, чтобы установить значение по умолчанию для параметра, и получите доступ к значению параметра после его изменения. Настроить шаблон действия настроек Android Studio.

.

## Результаты проделанной работы

Программный код файл, AppCompatPreferenceActivity.

package com.example.android.droidcafeinput;  
  
import android.content.res.Configuration;  
import android.os.Bundle;  
import android.preference.PreferenceActivity;  
import android.support.annotation.LayoutRes;  
import android.support.annotation.Nullable;  
import android.support.v7.app.ActionBar;  
import android.support.v7.app.AppCompatDelegate;  
import android.support.v7.widget.Toolbar;  
import android.view.MenuInflater;  
import android.view.View;  
import android.view.ViewGroup;  
  
  
public abstract class AppCompatPreferenceActivity extends PreferenceActivity {  
//и выполняет необходимые вызовы  
//\* для использования с AppCompat  
 private AppCompatDelegate mDelegate;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 getDelegate().installViewFactory();  
 getDelegate().onCreate(savedInstanceState);  
 super.onCreate(savedInstanceState);  
 }  
  
 @Override  
 protected void onPostCreate(Bundle savedInstanceState) {  
 super.onPostCreate(savedInstanceState);  
 getDelegate().onPostCreate(savedInstanceState);  
 }  
  
 public ActionBar getSupportActionBar() {  
 return getDelegate().getSupportActionBar();  
 }  
  
 public void setSupportActionBar(@Nullable Toolbar toolbar) {  
 getDelegate().setSupportActionBar(toolbar);  
 }  
  
 @Override  
 public MenuInflater getMenuInflater() {  
 return getDelegate().getMenuInflater();  
 }  
  
 @Override  
 public void setContentView(@LayoutRes int layoutResID) {  
 getDelegate().setContentView(layoutResID);  
 }  
  
 @Override  
 public void setContentView(View view) {  
 getDelegate().setContentView(view);  
 }  
  
 @Override  
 public void setContentView(View view, ViewGroup.LayoutParams params) {  
 getDelegate().setContentView(view, params);  
 }  
  
 @Override  
 public void addContentView(View view, ViewGroup.LayoutParams params) {  
 getDelegate().addContentView(view, params);  
 }  
  
 @Override  
 protected void onPostResume() {  
 super.onPostResume();  
 getDelegate().onPostResume();  
 }  
  
 @Override  
 protected void onTitleChanged(CharSequence title, int color) {  
 super.onTitleChanged(title, color);  
 getDelegate().setTitle(title);  
 }  
  
 @Override  
 public void onConfigurationChanged(Configuration newConfig) {  
 super.onConfigurationChanged(newConfig);  
 getDelegate().onConfigurationChanged(newConfig);  
 }  
  
 @Override  
 protected void onStop() {  
 super.onStop();  
 getDelegate().onStop();  
 }  
  
 @Override  
 protected void onDestroy() {  
 super.onDestroy();  
 getDelegate().onDestroy();  
 }  
  
 public void invalidateOptionsMenu() {  
 getDelegate().invalidateOptionsMenu();  
 }  
  
 private AppCompatDelegate getDelegate() {  
 if (mDelegate == null) {  
 mDelegate = AppCompatDelegate.*create*(this, null);  
 }  
 return mDelegate;  
 }  
}

Программный код файл, MainActivity.

package com.example.android.droidcafeinput;  
  
import android.content.Intent;  
import android.content.SharedPreferences;  
import android.os.Bundle;  
import android.preference.PreferenceManager;  
import android.support.design.widget.FloatingActionButton;  
import android.support.v7.app.AppCompatActivity;  
import android.support.v7.widget.Toolbar;  
import android.view.Menu;  
import android.view.MenuItem;  
import android.view.View;  
import android.widget.Toast;  
  
*/\*\*  
 \* Это приложение демонстрирует изображения, используемые в качестве кнопок, и плавающую кнопку действия для  
 \* намерение начать второе мероприятие. Приложение позволяет пользователю нажать на изображение, чтобы  
 \* сделать выбор. Приложение отображает тост, показывающий выбор пользователя, и отправляет  
 \* выбор в качестве данных с намерением запустить второе действие. Эта версия  
 \* включает опции в меню опций, в котором некоторые опции отображаются в  
 виде значков \* на панели приложений, а также включает кнопку Вверх.  
 \*  
 \* Эта версия также включает в себя опцию настроек. Пользователь нажимает кнопку Настройки  
 \* чтобы показать активность настроек и изменить настройки рынка.  
 \* При запуске приложение показывает тост, показывающий значение рыночной настройки.  
 \*/*public class MainActivity extends AppCompatActivity {  
  
 // Тег для дополнительного намерения.  
 public static final String *EXTRA\_MESSAGE* =  
 "com.example.android.droidcafeinput.extra.MESSAGE";  
  
 // Сообщение о заказе, отображаемое в тосте и отправляемое в новое действие.  
 private String mOrderMessage;  
  
 */\*\*  
 \* Создает представление содержимого, панель инструментов и плавающую кнопку действия.  
 \*  
 \* Этот метод предусмотрен в базовом шаблоне действия.  
 \*/* @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
 Toolbar toolbar = findViewById(R.id.*toolbar*);  
 setSupportActionBar(toolbar);  
  
 FloatingActionButton fab = findViewById(R.id.*fab*);  
 fab.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View view) {  
 Intent intent = new Intent(MainActivity.this,  
 OrderActivity.class);  
 intent.putExtra(*EXTRA\_MESSAGE*, mOrderMessage);  
 startActivity(intent);  
 }  
 });  
  
// Устанавливает значения по умолчанию только один раз, при первом вызове. Третий  
 // аргумент — это логическое значение, указывающее, являются ли значения по умолчанию  
 // должно быть установлено более одного раза. При значении false система устанавливает значение по умолчанию  
 // значения только при первом вызове.  
 PreferenceManager.*setDefaultValues*(this,  
 R.xml.*pref\_general*, false);  
 PreferenceManager.*setDefaultValues*(this,  
 R.xml.*pref\_notification*, false);  
 PreferenceManager.*setDefaultValues*(this,  
 R.xml.*pref\_account*, false);  
  
// Прочитайте настройки из общих настроек и отобразите тост.  
 SharedPreferences sharedPref = PreferenceManager  
 .*getDefaultSharedPreferences*(this);  
 String marketPref = sharedPref  
 .getString("sync\_frequency", "-1");  
 displayToast(marketPref);  
 }  
  
 */\*\*  
 \* Расширяет меню и добавляет элементы на панель действий, если она присутствует.  
 \*/* @Override  
 public boolean onCreateOptionsMenu(Menu menu) {  
 // Inflate the menu; this adds items to the action bar if it is present.  
 getMenuInflater().inflate(R.menu.*menu\_main*, menu);  
 return true;  
 }  
  
 */\*\*  
 \* Обрабатывает щелчки по элементам панели приложений.  
 \*/* @Override  
 public boolean onOptionsItemSelected(MenuItem item) {  
 switch (item.getItemId()) {  
 case R.id.*action\_order*:  
 Intent intent = new Intent(MainActivity.this,  
 OrderActivity.class);  
 intent.putExtra(*EXTRA\_MESSAGE*, mOrderMessage);  
 startActivity(intent);  
 return true;  
 case R.id.*action\_settings*:  
 Intent settingsIntent = new Intent(this,  
 SettingsActivity.class);  
 startActivity(settingsIntent);  
 return true;  
 case R.id.*action\_status*:  
 displayToast(getString(R.string.*action\_status\_message*));  
 return true;  
 case R.id.*action\_favorites*:  
 displayToast(getString(R.string.*action\_favorites\_message*));  
 return true;  
 case R.id.*action\_contact*:  
 displayToast(getString(R.string.*action\_contact\_message*));  
 return true;  
 default:  
 // Ничего не делать  
 }  
  
 return super.onOptionsItemSelected(item);  
 }  
  
 */\*\*  
 \* Отображает тост с сообщением.  
 \*/* public void displayToast(String message) {  
 Toast.*makeText*(getApplicationContext(), message,  
 Toast.*LENGTH\_SHORT*).show();  
 }  
  
 */\*\*  
 \* показывает сообщение о том, что изображение пончика было нажато.  
 \*/* public void showDonutOrder(View view) {  
 mOrderMessage = getString(R.string.*donut\_order\_message*);  
 displayToast(mOrderMessage);  
 }  
  
 */\*\*  
 \* Shows a message that the ice cream sandwich image was clicked.  
 \*/* public void showIceCreamOrder(View view) {  
 mOrderMessage = getString(R.string.*ice\_cream\_order\_message*);  
 displayToast(mOrderMessage);  
 }  
  
 */\*\*  
 \* Показывает сообщение о том, что изображение froyo было нажато.  
 \*/* public void showFroyoOrder(View view) {  
 mOrderMessage = getString(R.string.*froyo\_order\_message*);  
 displayToast(mOrderMessage);  
 }  
  
}

Программный код файл, OrderActivity.

package com.example.android.droidcafeinput;  
  
import android.content.Intent;  
import android.os.Bundle;  
import android.support.v7.app.AppCompatActivity;  
import android.view.View;  
import android.widget.RadioButton;  
import android.widget.TextView;  
import android.widget.Toast;  
  
*/\*\*  
 \* Это действие обрабатывает переключатели для выбора  
 \* способ доставки заказа и элементы управления вводом EditText.  
 \*/*public class OrderActivity extends AppCompatActivity {  
  
 */\*\*  
 \* Устанавливает для представления содержимого значение activity\_order и получает намерение и его  
 \* данные.  
 \*/* @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_order*);  
  
// Получить намерение и его данные.  
 Intent intent = getIntent();  
 String message = intent.getStringExtra(MainActivity.*EXTRA\_MESSAGE*);  
 TextView textView = findViewById(R.id.*order\_textview*);  
 textView.setText(message);  
 }  
  
 */\*\*  
 \* Проверяет, какой переключатель был нажат, и отображает всплывающее сообщение, чтобы  
 \* показать выбор.  
 \*/* public void onRadioButtonClicked(View view) {  
// Кнопка теперь проверена?  
 boolean checked = ((RadioButton) view).isChecked();  
// Проверьте, какой переключатель был нажат.  
 switch (view.getId()) {  
 case R.id.*sameday*:  
 if (checked)  
 // Обслуживание в тот же день  
 displayToast(  
 getString(R.string.*same\_day\_messenger\_service*));  
 break;  
 case R.id.*nextday*:  
 if (checked)  
 // Доставка на следующий день  
 displayToast(getString(R.string.*next\_day\_ground\_delivery*));  
 break;  
 case R.id.*pickup*:  
 if (checked)  
 // Забрать  
 displayToast(getString(R.string.*pick\_up*));  
 break;  
 default:  
 // Ничего не делай.  
 break;  
 }  
 }  
  
 */\*\*  
 \* Отображает фактическое сообщение в виде всплывающего сообщения.  
 \*/* public void displayToast(String message) {  
 Toast.*makeText*(getApplicationContext(), message,  
 Toast.*LENGTH\_SHORT*).show();  
 }  
}

Програмный код файл, SettingsActivity.

package com.example.android.droidcafeinput;  
  
import android.annotation.TargetApi;  
import android.content.Context;  
import android.content.Intent;  
import android.content.res.Configuration;  
import android.media.Ringtone;  
import android.media.RingtoneManager;  
import android.net.Uri;  
import android.os.Build;  
import android.os.Bundle;  
import android.preference.ListPreference;  
import android.preference.Preference;  
import android.preference.PreferenceFragment;  
import android.preference.PreferenceManager;  
import android.preference.RingtonePreference;  
import android.support.v4.app.NavUtils;  
import android.support.v7.app.ActionBar;  
import android.text.TextUtils;  
import android.view.MenuItem;  
  
import java.util.List;  
  
*/\*\*  
 \* PreferenceActivity, который представляет набор настроек приложения. На телефонной трубке  
 \* устройства, настройки представлены в виде единого списка. На планшетах настройки следующие  
 \* разделено по категориям, заголовки категорий отображаются слева от списка  
 настроек \*.  
 \*/*public class SettingsActivity extends AppCompatPreferenceActivity {  
  
 */\*\*  
 \* \* Прослушиватель изменения значения предпочтения, который обновляет сводку предпочтения  
 \* чтобы отразить его новую ценность.  
 \*/* private static Preference.OnPreferenceChangeListener  
 *sBindPreferenceSummaryToValueListener* =  
 new Preference.OnPreferenceChangeListener() {  
 @Override  
 public boolean onPreferenceChange(Preference preference,  
 Object value) {  
 String stringValue = value.toString();  
  
 if (preference instanceof ListPreference) {  
// Для получения предпочтений списка найдите правильное отображаемое значение в  
 // // список записей предпочтения.  
 ListPreference listPreference = (ListPreference) preference;  
 int index = listPreference.findIndexOfValue(stringValue);  
  
 // Установите сводку так, чтобы она отражала новое значение.  
 preference.setSummary(  
 index >= 0  
 ? listPreference.getEntries()[index]  
 : null);  
  
 } else if (preference instanceof RingtonePreference) {  
 // Для настройки мелодии звонка найдите правильное отображаемое значение  
 // с помощью RingtoneManager.  
 if (TextUtils.*isEmpty*(stringValue)) {  
 // Пустые значения соответствуют "беззвучному" (без мелодии звонка).  
 preference.setSummary(R.string.*pref\_ringtone\_silent*);  
  
 } else {  
 Ringtone ringtone = RingtoneManager.*getRingtone*(  
 preference.getContext(), Uri.*parse*(stringValue));  
  
 if (ringtone == null) {  
 // Очистите сводку, если произошла ошибка поиска.  
 preference.setSummary(null);  
 } else {  
 // Установите сводку так, чтобы она отражала отображение новой мелодии звонка  
 // имя.  
 String name = ringtone.getTitle(preference.getContext());  
 preference.setSummary(name);  
 }  
 }  
  
 } else {  
 // Для всех других настроек установите для сводки значение  
 // простое строковое представление.  
 preference.setSummary(stringValue);  
 }  
 return true;  
 }  
 };  
  
 */\*\*  
 \* Определяет, имеет ли устройство очень большой экран. Например, 10"  
 \* таблетки очень большие.  
 \*/* private static boolean isXLargeTablet(Context context) {  
 return (context.getResources().getConfiguration().screenLayout  
 & Configuration.*SCREENLAYOUT\_SIZE\_MASK*)  
 >= Configuration.*SCREENLAYOUT\_SIZE\_XLARGE*;  
 }  
  
 */\*\*  
 \*\* Привязывает сводку предпочтения к его значению. Более конкретно, когда  
 \* значение предпочтения \* изменяется, его краткое содержание (строка текста под заголовком  
 \* предпочтения \*) обновляется, чтобы отразить значение. Сводка также  
 \* \* немедленно обновляется при вызове этого метода. Точный формат отображения  
 \* \* зависит от типа предпочтения.  
 \*/* private static void bindPreferenceSummaryToValue(Preference preference) {  
 // Установите прослушиватель так, чтобы он отслеживал изменения значений.  
 preference.setOnPreferenceChangeListener  
 (*sBindPreferenceSummaryToValueListener*);  
  
 // // Немедленно запустите прослушиватель с помощью предпочтения  
 // текущее значение.  
 *sBindPreferenceSummaryToValueListener* .onPreferenceChange(preference, PreferenceManager  
 .*getDefaultSharedPreferences*(preference.getContext())  
 .getString(preference.getKey(), ""));  
 }  
  
 */\*\*  
 \* Настраивает панель действий для действия.  
 \*/* @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setupActionBar();  
 }  
  
 */\*\*  
 \* Настройте панель действий android.app.ActionBar, если доступен API.  
 \*/* private void setupActionBar() {  
 ActionBar actionBar = getSupportActionBar();  
 if (actionBar != null) {  
 // Показать кнопку "Вверх" на панели действий.  
 actionBar.setDisplayHomeAsUpEnabled(true);  
 }  
 }  
  
 @Override  
 public boolean onMenuItemSelected(int featureId, MenuItem item) {  
 int id = item.getItemId();  
 if (id == android.R.id.*home*) {  
 if (!super.onMenuItemSelected(featureId, item)) {  
 NavUtils.*navigateUpFromSameTask*(this);  
 }  
 return true;  
 }  
 return super.onMenuItemSelected(featureId, item);  
 }  
  
 @Override  
 public boolean onIsMultiPane() {  
 return *isXLargeTablet*(this);  
 }  
  
 @Override  
 @TargetApi(Build.VERSION\_CODES.*HONEYCOMB*)  
 public void onBuildHeaders(List<Header> target) {  
 loadHeadersFromResource(R.xml.*pref\_headers*, target);  
 }  
  
 */\*\*  
 \* Этот метод останавливает внедрение фрагментов в вредоносные приложения.  
 \* Убедитесь, что здесь запрещены любые неизвестные фрагменты.  
 \*/* protected boolean isValidFragment(String fragmentName) {  
 return PreferenceFragment.class.getName().equals(fragmentName)  
 || GeneralPreferenceFragment  
 .class.getName().equals(fragmentName)  
 || DataSyncPreferenceFragment  
 .class.getName().equals(fragmentName)  
 || NotificationPreferenceFragment  
 .class.getName().equals(fragmentName);  
 }  
  
 */\*\*  
 \* Этот фрагмент показывает только общие настройки. Он используется, когда действие  
 \* отображает пользовательский интерфейс настроек с двумя панелями.  
 \*/* @TargetApi(Build.VERSION\_CODES.*HONEYCOMB*)  
 public static class GeneralPreferenceFragment  
 extends PreferenceFragment {  
 @Override  
 public void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 addPreferencesFromResource(R.xml.*pref\_general*);  
 setHasOptionsMenu(true);  
  
// Привязать сводки настроек EditText/List/Dialog/Ringtone  
 // к их ценностям. Когда их значения меняются, их сводки  
// обновляются, чтобы отразить новое значение, в соответствии с дизайном Android  
 // руководящие принципы.  
 *bindPreferenceSummaryToValue*(findPreference("example\_text"));  
 *bindPreferenceSummaryToValue*(findPreference("example\_list"));  
 }  
  
 @Override  
 public boolean onOptionsItemSelected(MenuItem item) {  
 int id = item.getItemId();  
 if (id == android.R.id.*home*) {  
 // // Сначала проверьте размер таблицы. Если true, запустите MainActivity.  
 boolean tabletSize = getResources()  
 .getBoolean(R.bool.*isTablet*);  
 if (tabletSize) {  
 startActivity(new Intent(getActivity(),  
 MainActivity.class));  
 } else {  
 startActivity(new Intent(getActivity(),  
 SettingsActivity.class));  
 }  
 return true;  
 }  
 return super.onOptionsItemSelected(item);  
 }  
 }  
  
 */\*\*  
 \* Этот фрагмент показывает только настройки уведомлений. Он используется, когда действие  
 \* отображает пользовательский интерфейс настроек с двумя панелями.  
 \*/* @TargetApi(Build.VERSION\_CODES.*HONEYCOMB*)  
 public static class NotificationPreferenceFragment  
 extends PreferenceFragment {  
 @Override  
 public void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 addPreferencesFromResource(R.xml.*pref\_notification*);  
 setHasOptionsMenu(true);  
  
// Привязать сводки настроек EditText/List/Dialog/Ringtone  
 // к их ценностям. Когда их значения меняются, их сводки  
// обновляются, чтобы отразить новое значение, в соответствии с дизайном Android  
 // руководящие принципы.  
 *bindPreferenceSummaryToValue* (findPreference("notifications\_new\_message\_ringtone"));  
 }  
  
 @Override  
 public boolean onOptionsItemSelected(MenuItem item) {  
 int id = item.getItemId();  
 if (id == android.R.id.*home*) {  
 // Check first for tabletSize. If true, start MainActivity.  
 boolean tabletSize = getResources()  
 .getBoolean(R.bool.*isTablet*);  
 if (tabletSize) {  
 startActivity(new Intent(getActivity(),  
 MainActivity.class));  
 } else {  
 startActivity(new Intent(getActivity(),  
 SettingsActivity.class));  
 }  
 return true;  
 }  
 return super.onOptionsItemSelected(item);  
 }  
 }  
  
 */\*\*  
 \* Этот фрагмент показывает только данные и настройки синхронизации. Он используется, когда действие  
 \* отображает пользовательский интерфейс настроек с двумя панелями.  
 \*/* @TargetApi(Build.VERSION\_CODES.*HONEYCOMB*)  
 public static class DataSyncPreferenceFragment  
 extends PreferenceFragment {  
 @Override  
 public void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 addPreferencesFromResource(R.xml.*pref\_account*);  
 setHasOptionsMenu(true);  
  
// Привязать сводки настроек EditText/List/Dialog/Ringtone  
 // к их ценностям. Когда их значения меняются, их сводки  
// обновляются, чтобы отразить новое значение, в соответствии с дизайном Android  
 // руководящие принципы.  
 *bindPreferenceSummaryToValue* (findPreference("sync\_frequency"));  
 }  
  
 @Override  
 public boolean onOptionsItemSelected(MenuItem item) {  
 int id = item.getItemId();  
 if (id == android.R.id.*home*) {  
 // // Сначала проверьте размер таблицы. Если true, запустите MainActivity.  
 boolean tabletSize = getResources()  
 .getBoolean(R.bool.*isTablet*);  
 if (tabletSize) {  
 startActivity(new Intent(getActivity(),  
 MainActivity.class));  
 } else {  
 startActivity(new Intent(getActivity(),  
 SettingsActivity.class));  
 }  
 return true;  
 }  
 return super.onOptionsItemSelected(item);  
 }  
 }  
}

На рисунке 1 показано начальное состояние программы.

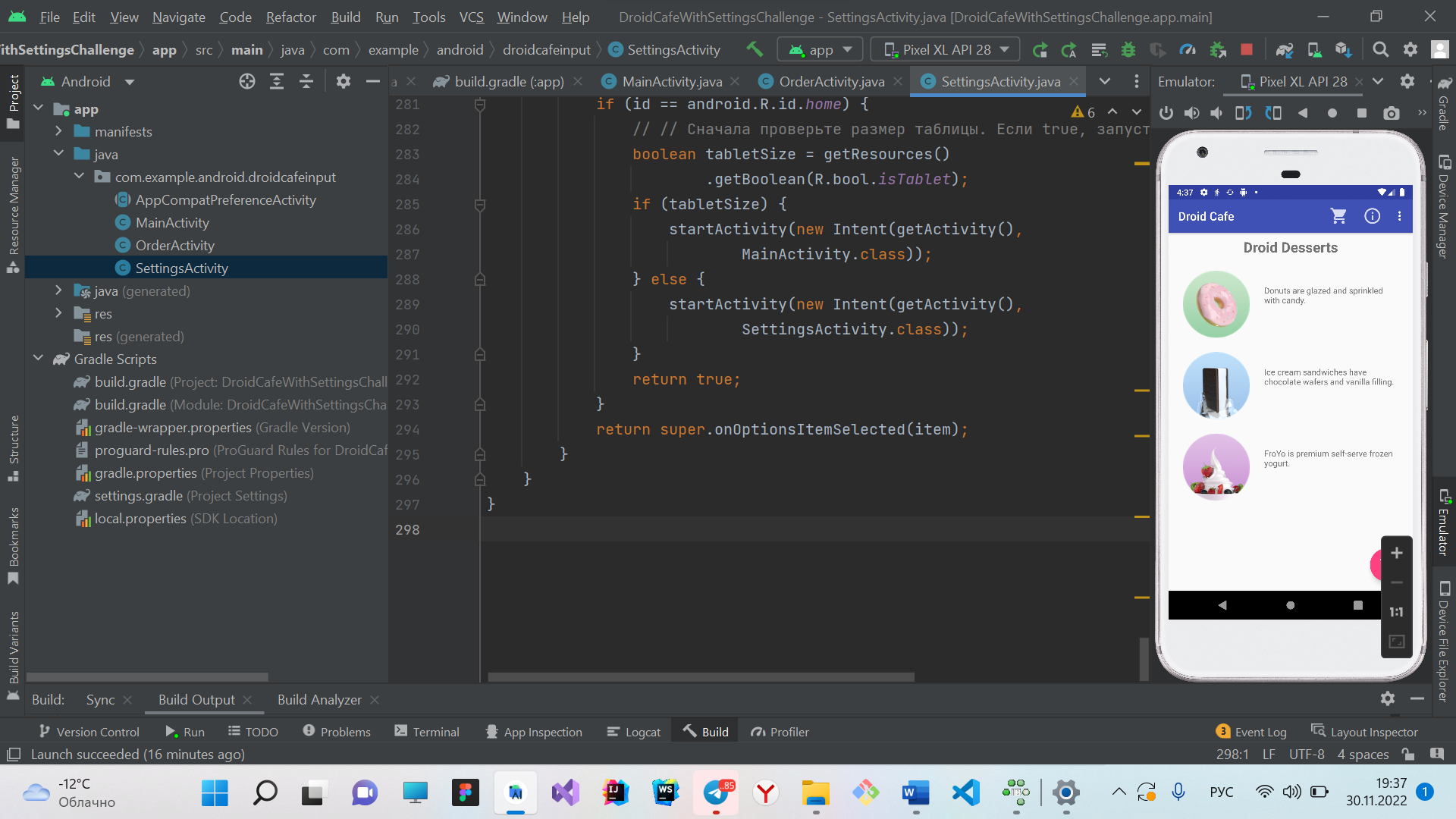


Рисунок 1 – Начальное состояние программы

На рисунке 2 показана работа кнопок и тост уведомлений.

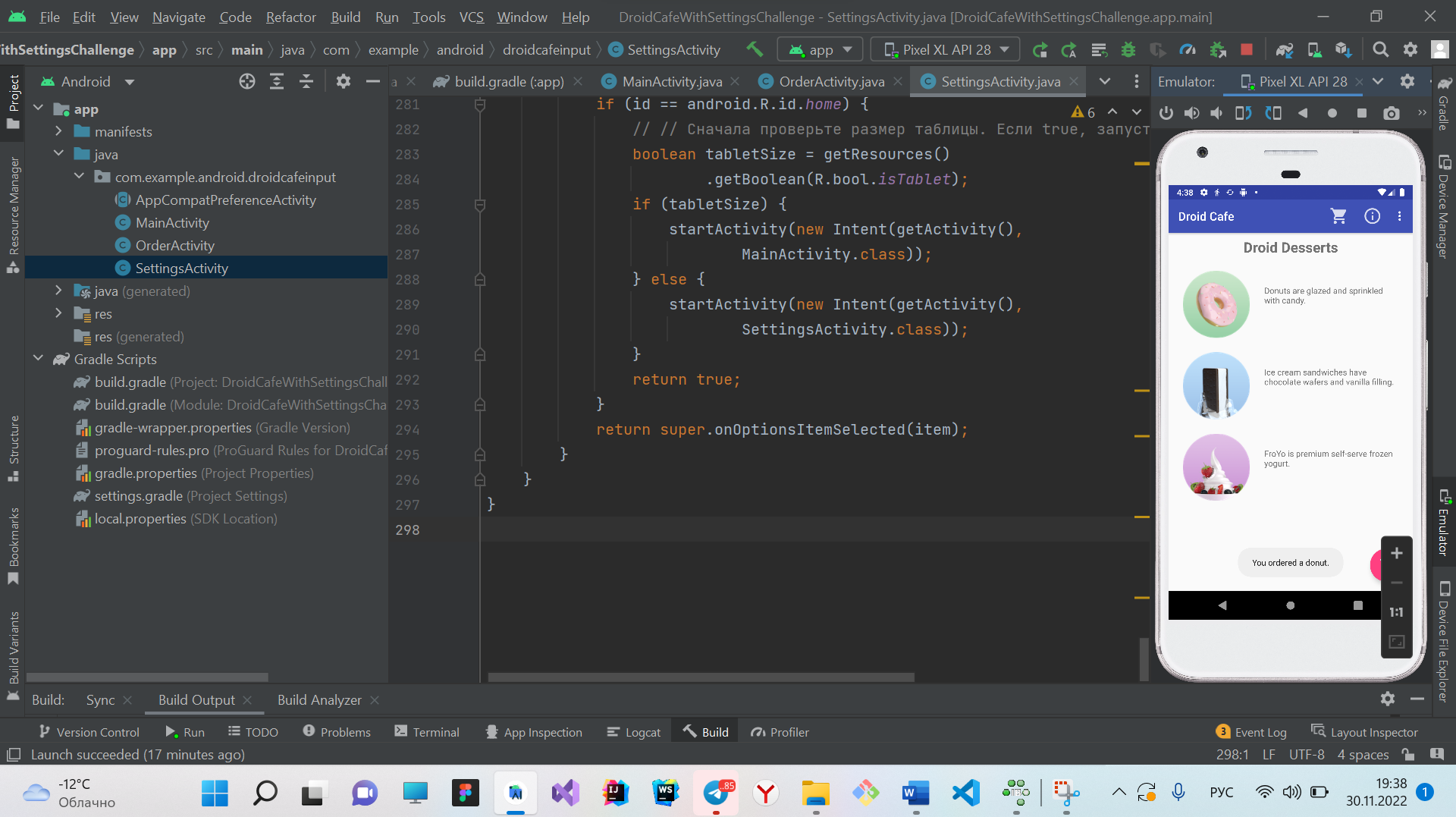


Рисунок 2 – Результат работы кнопок Тост

На рисунке 3 показана вторая страница программы, оформление заказа.

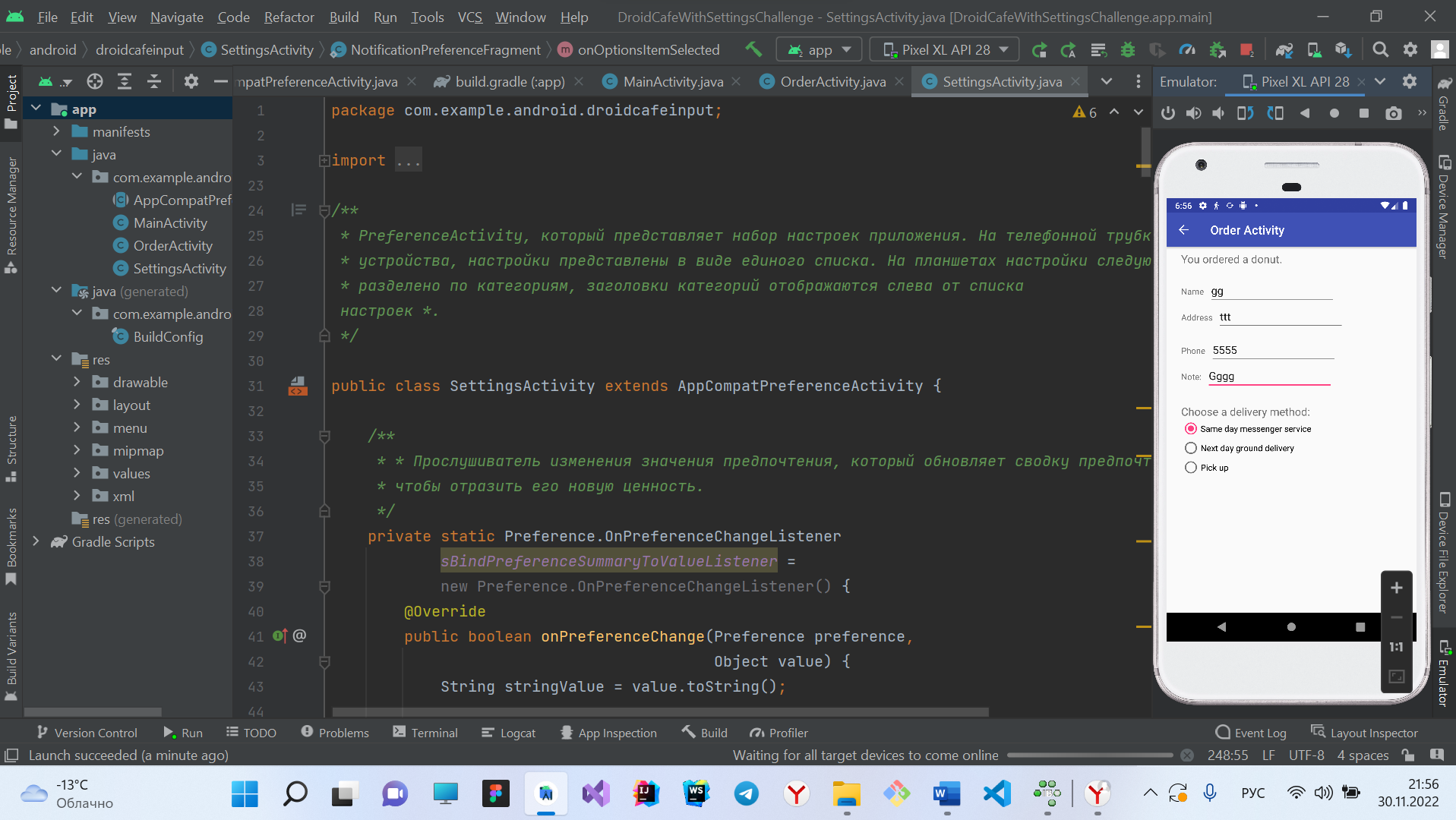


Рисунок 3 – Страница оформления заказа

## **Анализ полученных результатов**

В результате выполнения данной лабораторной работы мной был разработан проект, в котором я научился, как добавить в Фрагмент для управления настройками. Как создать XML-файл ресурсов настроек с их атрибутами. Как создать навигацию к настройкам Activity. Как установить значения настроек по умолчанию. Как прочитать значения настроек, измененные пользователем. Как настроить шаблон действия настроек.

.