**SharedPreferences в Android сохраняем настройки**

В этом уроке мы будем использовать SharedPreferences для хранения пользовательских данных в виде пар ключ-значение.

**SharedPreferences в Android. Теория**

SharedPreferences позволяют приложениям сохранять настройки в виде пар ключ-значение. Эти настройки сохраняются даже когда пользователь закрывает приложение.

Android хранит SharedPreferences в виде XML файла в папке shared\_prefs по адресу

data --> data --> [имя пакета] -->.

Папку data можно получить с помощью вызова метода Environment.getDataDirectory().

SharedPreferences зависят от приложения, то есть сохраненные данные будут удалены из Android устройства после **удаления приложения** или после **очистки данных** приложения.

**Обзор API SharedPreferences**

Android предоставляет три метода получения доступа к SharedPreferences:

* Метод getPreferences() используется внутри Activity для доступа к одному общему файлу настроек приложения.
* Метод getSharedPreferences() используется внутри Activity для доступа к разным файлам настроек приложения.
* Метод getDefaultSharedPreferences() используется классом PreferenceManager, чтобы получить доступ к стандартному файлу настроек.

В этом уроке мы познакомимся с методом getSharedPreferences() подробнее.

Этот метод используется следующим образом:

|  |
| --- |
| getSharedPreferences (String prefs\_name, int mode) |

где prefs\_name — имя файла с настройками, а mode —  способ получения доступа.

Всего Android предоставляет несколько способов получения доступа:

* **MODE\_PRIVATE** — режим по умолчанию, где созданный файл может быть доступен только вызывающему приложению
* **MODE\_WORLD\_READABLE** — создание файлов, к которым могут получить доступ на чтение из любого другого приложения (опасный метод, который часто приводит к проблемам в безопасности приложения)
* **MODE\_WORLD\_WRITEABLE** — создание файлов, к которым могут получить доступ (чтение и запись) из любого другого приложения.
* **MODE\_MULTI\_PROCESS** был помечен как **deprecated** в Android API 23, так как не работал правильно в некоторых версиях Android и не был потокобезопасным.
* **MODE\_APPEND** — добавляет новые настройки к уже существующим.

**SharedPreferences в Android. Практика**

Начиная с этого урока мы будем выносить текст или названия виджетов, используемых в макете в файл strings.xml, который находится в папке res/values вашего проекта. Это является хорошей практикой программирования под Android, которой мы будем следовать. Вот файл strings.xml для нашего проекта:

|  |
| --- |
| <resources>      <string name="app\_name">SharedPreferencesApp</string>      <string name="save\_txt">Сохранить</string>      <string name="retrieve\_txt">Получить</string>      <string name="clear">Очистить</string>  </resources> |

Макет activity\_main.xml состоит из одного виджета EditText и трех кнопок для сохранения, очистки и получения пользовательских данных из SharedPreferences:

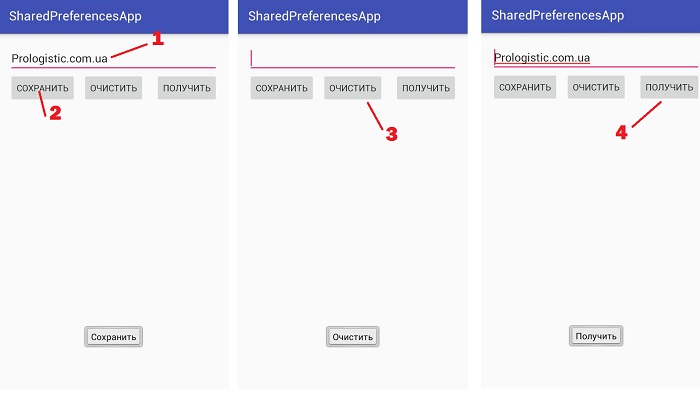
|  |
| --- |
| <?xml version="1.0" encoding="utf-8"?>  <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"      xmlns:tools="http://schemas.android.com/tools"      android:layout\_width="match\_parent"      android:layout\_height="match\_parent"      android:paddingBottom="@dimen/activity\_vertical\_margin"      android:paddingLeft="@dimen/activity\_horizontal\_margin"      android:paddingRight="@dimen/activity\_horizontal\_margin"      android:paddingTop="@dimen/activity\_vertical\_margin"      tools:context="ua.com.prologistic.sharedpreferencesapp.MainActivity">        <Button          android:id="@+id/saveBtn"          android:layout\_width="wrap\_content"          android:layout\_height="wrap\_content"          android:layout\_alignParentLeft="true"          android:layout\_alignParentStart="true"          android:layout\_alignTop="@+id/clearBtn"          android:onClick="onClick"          android:text="@string/save\_txt" />        <Button          android:id="@+id/getBtn"          android:layout\_width="wrap\_content"          android:layout\_height="wrap\_content"          android:layout\_alignEnd="@+id/editName"          android:layout\_alignRight="@+id/editName"          android:layout\_below="@+id/editName"          android:onClick="onClick"          android:text="@string/retrieve\_txt" />        <Button          android:id="@+id/clearBtn"          android:layout\_width="wrap\_content"          android:layout\_height="wrap\_content"          android:layout\_below="@+id/editName"          android:layout\_centerHorizontal="true"          android:onClick="onClick"          android:text="@string/clear" />        <EditText          android:id="@+id/editName"          android:layout\_width="match\_parent"          android:layout\_height="wrap\_content"          android:layout\_alignParentTop="true"          android:inputType="text" />    </RelativeLayout> |

Обратите внимание, что мы назначили в атрибуте onClick один и тот же метод. Ниже в листинге класса MyActivity.class вы увидите как мы обрабатываем нажатия для каждой кнопки:

|  |
| --- |
| package ua.com.prologistic.sharedpreferencesapp;    import android.content.Context;  import android.content.SharedPreferences;  import android.os.Bundle;  import android.support.v7.app.AppCompatActivity;  import android.view.View;  import android.widget.EditText;  import android.widget.Toast;    public class MainActivity extends AppCompatActivity implements View.OnClickListener {        private SharedPreferences sharedPrefs;      private EditText editText;        public static final String myPrefs = "myprefs";      public static final String nameKey = "nameKey";    *@Override*      protected void onCreate(Bundle savedInstanceState) {          super.onCreate(savedInstanceState);          setContentView(R.layout.activity\_main);            editText = (EditText) findViewById(R.id.editName);            // при старте проверяем есть ли в файлах настроек          // данные по ключу nameKey          sharedPrefs = getSharedPreferences(myPrefs, Context.MODE\_PRIVATE);          if (sharedPrefs.contains(nameKey)) {              // если есть, то ставим значение этого ключа в EditText              editText.setText(sharedPrefs.getString(nameKey, ""));          }      }        // метод для сохранения текста в файл настроек      public void saveText() {          // получаем текст          String editTextValue = editText.getText().toString();            // получаем доступ к файлу          SharedPreferences.Editor editor = sharedPrefs.edit();          // сохраняем по текст из EditText по ключу nameKey          editor.putString(nameKey, editTextValue);          editor.apply();      }        // метод для получения текста из SharedPreferences по ключу      public void getText() {          sharedPrefs = getSharedPreferences(myPrefs, Context.MODE\_PRIVATE);          if (sharedPrefs.contains(nameKey)) {              editText.setText(sharedPrefs.getString(nameKey, ""));          }      }        // получаем id нажатого виджета и вызываем соответственный ему метод  *@Override*      public void onClick(View v) {          int id = v.getId();          switch (id) {              case R.id.saveBtn:                  // сохраняем текст в файл настроек                  saveText();                  Toast.makeText(this, "Сохранить", Toast.LENGTH\_LONG).show();                  break;                case R.id.clearBtn:                  // очищает текст в виджете EditText                  editText.setText("");                  Toast.makeText(this, "Очистить", Toast.LENGTH\_LONG).show();                  break;                case R.id.getBtn:                  // получаем текст из SharedPreferences                  Toast.makeText(this, "Получить", Toast.LENGTH\_LONG).show();                  getText();                  break;          }      }  } |

Обратите внимание, что мы реализуем интерфейс View.OnClickListener и переопределяем метод onClick(), в котором с помощью [оператора switch](http://prologistic.com.ua/primer-ispol-zovaniya-string-v-switch-na-java.html) находим id виджета, который был нажат. В макете activity\_main.xml именно этот метод мы назначили общим для всех кнопок.

Результат работы приложения сохранения настроек SharedPreferences в Android:

[](http://prologistic.com.ua/wp-content/uploads/2016/03/result.jpg)

Как видим, на рисунку представлен ход выполнения операций: сначала пишем текст в поле (1), затем нажимаем Сохранить (2), потом нажимаем Очистить (3) и в конце тестируем получение настроек из SharedPreferences — нажимаем Получить (4).