

# Курс:

## Программирование под Android с нуля

Тренер:

Пономарев Алексей

[oleksii\\_android@web-academy.com.ua](mailto:oleksii_android@web-academy.com.ua)

Лекция 11

Preferences, files



# Write preferences

**//Создание (открытие) файла настроек**

```
SharedPreferences preferences =  
    getPreferences(MODE_PRIVATE) ;
```

**//Объект который управляет записью в настройки**

```
SharedPreferences.Editor editor = preferences.edit() ;
```

```
editor.putString("Test", "test") ;    //Запись настроек  
editor.commit() ;                      //Сохранение настроек
```

```
SharedPreferences preferences2 =
```

```
    getSharedPreferences("MyPref", MODE_PRIVATE) ;
```

```
SharedPreferences.Editor editor2 = preferences2.edit() ;
```

```
editor2.putString("Test", "test2") ;  
editor2.commit() ;
```

# Read preferences

```
//Подключение к файлу настроек
```

```
SharedPreferences preferences1 =  
    getPreferences(MODE_PRIVATE);
```

```
//Чтение настроек
```

```
String text1 = preferences1.getString("Test", "");
```

```
SharedPreferences preferences3 =  
    getSharedPreferences("MyPref", MODE_PRIVATE);
```

```
String text3 = preferences3.getString("Test", "");
```

# Preferences xml

```
<?xml version="1.0" encoding="utf-8"?>
<PreferenceScreen xmlns:android="http://schemas.android.com/apk/res/android">
    <CheckBoxPreference
        android:key="pref1" //id
        android:summary="Check box preference" //Заголовок
        android:title="Check box" //Описание
    </CheckBoxPreference>

    <EditTextPreference
        android:key="pref2"
        android:summary="Text preference"
        android:title="Text">
    </EditTextPreference>

    <PreferenceCategory //Группа
        android:title="Preference group">

        <EditTextPreference
            android:defaultValue="1000" //Значение по умолчанию
            android:key="pref3"
            android:numeric="integer" //Тип
            android:summary="Integer preference"
            android:title="Integer"/>
        </PreferenceCategory>
</PreferenceScreen>
```

# Preferences activity

```
public class PreferenceActivity extends PreferenceActivity {  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
  
        //Установка ресурса с настройками  
        addPreferencesFromResource(R.xml.preferences);  
    }  
}
```

```
//Чтение настроек приложения  
SharedPreferences preferences =  
    PreferenceManager.getDefaultSharedPreferences(this);
```

# Write internal file

```
private void saveInternalFile(String fileName, String data) {  
    try {  
        BufferedWriter writer = //Запись файла по частям  
            new BufferedWriter(  
                new OutputStreamWriter( //Поток для записи файла  
                    openFileOutput(fileName, Context.MODE_PRIVATE)));  
  
        writer.write(data);  
        writer.flush();  
        writer.close();  
    } catch (Exception e) {  
        e.printStackTrace();  
    }  
}
```

# Read internal file

```
private String readInternalFile(String fileName) {
    try {
        StringBuilder builder = new StringBuilder();

        //Поток для чтения файла
        InputStream inputStream = openFileInput(fileName);
        InputStreamReader inputStreamReader = new InputStreamReader(inputStream);

        //Чтение файла по строкам
        BufferedReader reader = new BufferedReader(inputStreamReader);
        String line;

        while ((line = reader.readLine()) != null) {
            builder.append(line);
        }

        reader.close();

        return builder.toString();
    } catch (Exception e) {
        e.printStackTrace();
    }

    return null;
}
```

# External files

```
//Разрешения на чтение и запись файлов
```

```
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>  
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />
```

```
//Проверка, что карта памяти подключена
```

```
Environment.getExternalStorageState()  
    .equals(Environment.MEDIA_MOUNTED)
```

```
//Подключение к карте памяти
```

```
File folder = Environment.getExternalStorageDirectory();
```

```
//Создание директории
```

```
folder = new File(folder.getAbsolutePath() + "/MyFolder");
```

```
if (!folder.exists()) {  
    folder.mkdirs();  
}
```



# Permissions

//Запрос к пользователю за разрешением

```
if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.M) {  
    String[] permissions = {  
        Manifest.permission.WRITE_EXTERNAL_STORAGE,  
        Manifest.permission.READ_EXTERNAL_STORAGE  
    };  
  
    requestPermissions(permissions, 0);  
}
```

//Проверка разрешений

```
if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.M) {  
    return ContextCompat.checkSelfPermission(this, permission) ==  
        PackageManager.PERMISSION_GRANTED;  
}
```

# Write external file

```
private void saveExternalFile(File folder, String fileName,
    String data) {
    File file = new File(folder, fileName);

    try {
        //Поток для записи файла
        BufferedWriter writer =
            new BufferedWriter(
                new OutputStreamWriter(new FileOutputStream(file),
                    "UTF8"));

        writer.write(data); //Передача текста в поток
        writer.flush();     //Запись потока в файл
        writer.close();     //Закрытие потока
    } catch (Exception e) {
        e.printStackTrace();
    }
}
```

# Read external file

```
private String readExternalFile(File folder, String fileName) {  
    File file = new File(folder, fileName);  
  
    try {  
        if (file.exists()) {                                //Подключение к файлу  
            StringBuilder builder = new StringBuilder();  
            BufferedReader reader = new BufferedReader(new FileReader(file));  
            String line;  
  
                                //Чтение файла по строкам  
            while ((line = reader.readLine()) != null) {  
                builder.append(line);  
            }  
  
            reader.close();                                //Закрытие потока  
  
            return builder.toString();                    //Получение результата  
        }  
    } catch (Exception e) {  
        e.printStackTrace();  
    }  
  
    return null;  
}
```

# JSON

```
{  
  "firstName": "Иван",  
  "lastName": "Иванов",  
  "address":  
    {  
      "streetAddress": "Московское ш., 101, кв.101",  
      "city": "Ленинград",  
      "postalCode": 101101  
    },  
  "phoneNumbers": [ "812 123-1234", "916 123-4567" ]  
}
```

# To JSON

```
//Подключить json в build.gradle  
compile 'com.google.code.gson:gson:+'
```

```
public class Student {  
    @SerializedName("FirstName")           //Атрибут для gson  
    public String FirstName;  
  
    @SerializedName("LastName")  
    public String LastName;  
  
    @SerializedName("Age")  
    public int Age;  
}
```

```
Student student = new Student("Ivan", "Ivanov", 22);
```

```
Gson gson = new GsonBuilder().create();
```

```
String json = gson.toJson(student); //JSON из объекта
```

# From JSON

```
String json = //JSON строка
```

```
Gson gson = new GsonBuilder().create();
```

```
//Объект из строки
```

```
Student student = gson.fromJson(json, Student.class);
```

# Simple XML

<http://simple.sourceforge.net>

//build.gradle

```
compile('org.simpleframework:simple-xml:2.7.+'){  
    exclude module: 'stax'  
    exclude module: 'stax-api'  
    exclude module: 'xpp3' }
```

//Создание файла

```
File file = new File(getFilesDir() + "/student.xml");
```

# Спасибо за внимание

Тренер:

Пономарев Алексей

[oleksii\\_android@web-academy.com.ua](mailto:oleksii_android@web-academy.com.ua)

Лекция 11  
Preferences, files

