# Программные и аппаратные средства информатики

Лекция 5. Программирование для мобильных устройств

Содержание лекции:

- Расширяемый язык разметки XML
- Пример разработки в Android Studio

Преподаватель курса: Нефедов Денис Геннадьевич, к.т.н., доцент

# Расширяемый язык разметки XML

**XML** (eXtensible Markup Language — расширяемый язык разметки) — рекомендованный W3C универсальный язык разметки, предназначенный для описания структурированных данных, обмена информацией между программами и создания специализированных языков разметки (XML-словарей).

## Применение

- Традиционная обработка данных. XML платформо-независимый стандарт хранения и представления информации. Используется для обмена данными между сервером и клиентом.
- Программирование, управляемое документом. XML-документы могут служить контейнерами для построения приложений из существующих интерфейсов и компонентов.
- Архивирование компонентов. Архивирование компонентов, требующее единообразного подхода к их хранению и использованию, позволяет легко собрать компоненты в единое целое с помощью несложного дополнительного кодирования.
- Внедрение данных. Позволяет автоматизировать рутинную обработку данных.

## Структура XML-документа

- декларации,
- элементы,
- комментарии,
- специальные символы,
- директивы.

Синтаксис XML похож на HTML, т.к. оба являются диалектом языка **SGML** (англ. Standard Generalized Markup Language — стандартный обобщённый язык разметки)

XML — это теговый язык разметки документов. Иными словами, любой документ на языке XML представляет собой набор элементов, причем начало и конец каждого элемента обозначается специальными пометками, называемыми тегами.

### Элемент состоит из трех частей:

- начального тега,
- содержимого,
- конечного тега.

**Тег** — это текст, заключенный в угловые скобки "<" и ">". Конечный тег имеет то же имя, что начальный тег, но начинается с косой черты "/".

Пример XML-элемента:

<author>Сергей Довлатов</author>

Любой элемент может иметь **атрибуты**, содержащие дополнительную информацию об элементе. Атрибуты всегда включаются в начальный тег элемента и имеют вид:

имя\_атрибута="значение\_атрибута"

Атрибут обязан иметь значение, которое всегда должно быть заключено в одинарные или двойные кавычки. Пример элемента, имеющего атрибут:

<author country="USA">Сергей Довлатов</author>

Любой XML-документ состоит из пролога и корневого элемента, например:

```
<?xml version="1.0"?>
<books>
<book isbn="0345374827">
<title>Марш одиноких</title>
<author>Довлатов, Сергей</author>
cprex</book>
</books>
```

Здесь пролог – единственная **Директива** (первая строка документа), указывающая версию XML. За ней следует XML-элемент с уникальным именем, который содержит в себе все остальные элементы и называется **корневым**.

Директива – выражение, заключенное в специальные теги "<?" и "?>", которое содержит указания программе, обрабатывающей XML-документ.

<?xml version="1.0" encoding="ISO-8859-1" standalone="yes"?>
Aтрибут encoding задает кодировку символов документа (по умолчанию UTF-8),
Aтрибут standalone говорит о том, содержит ли данный документ внешние разделы.

#### Пример комментариев:

```
<!-- это комментарий -->
```

Все имена элементов, атрибутов и разделов должны начинаться с буквы Unicode и состоять из букв, цифр, символов точки (.), подчеркивания (\_) и дефиса (-).

**Данные**, т. е. содержимое элементов и значения атрибутов, могут состоять из любых символов, кроме:

```
левая угловая скобка ("<") &lt; правая угловая скобка (">") &gt; aмперсант ("&") &amp; двойная кавычка (") в значениях атрибутов &quot; одинарная кавычка (') в значениях атрибутов &apos;
```

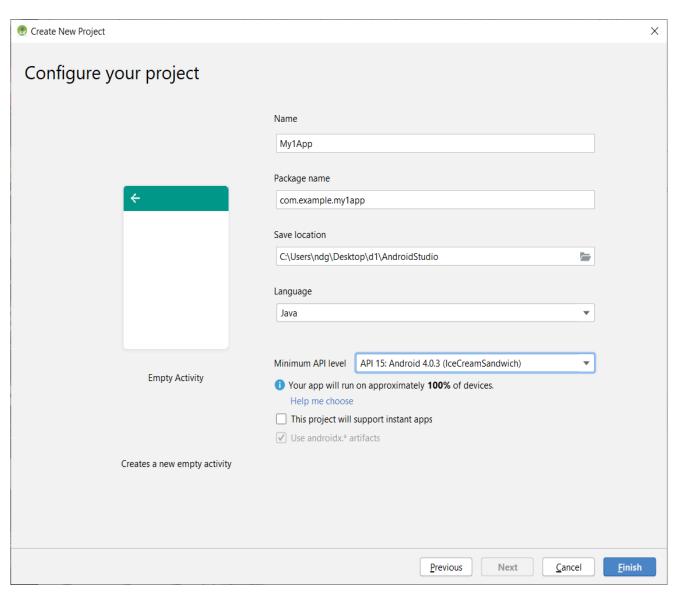
Второй способ включения недопустимых символов использование секции CDATA:

```
<layout>
<! [CDATA[<H1>Заголовок</H1>]]>
</layout>
```

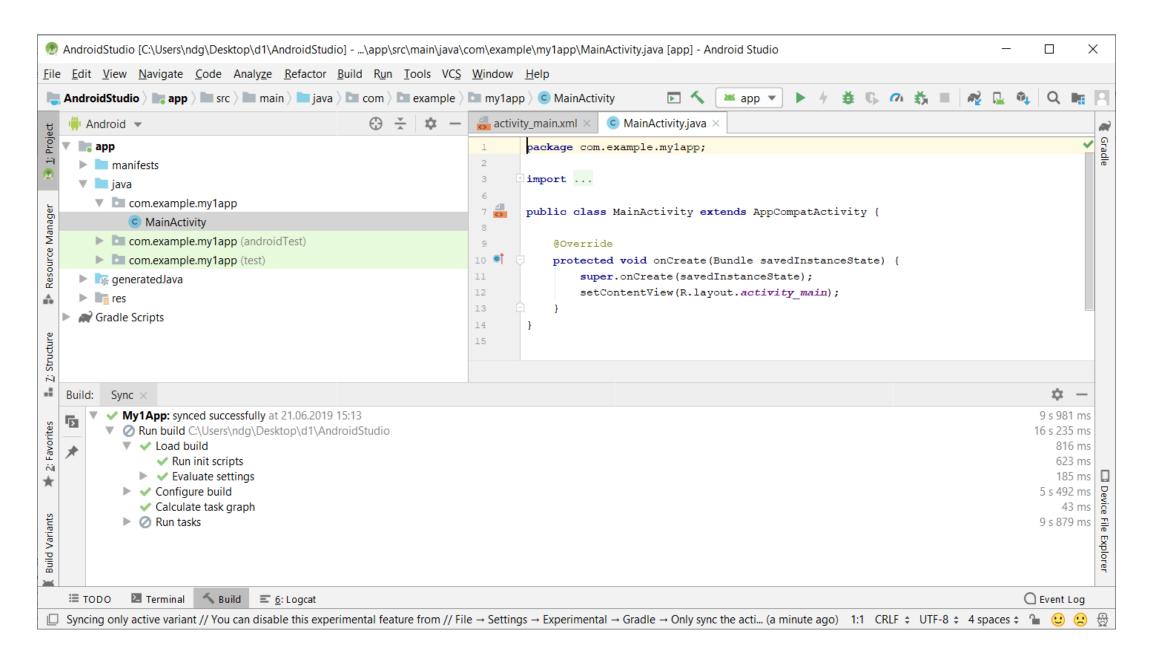
# Пример разработки в Android Studio

Для создания приложений должен быть установлен пакет JDK 8, который необходим для разработки на языке Java. JDK 8 можно найти на сайте компании Oracle.

Создадим проект в Android Studio. При создании проекта нам будет предложено несколько шаблонов для различных ситуаций, но самыми распространенными являются Basic Activity и Empty Activity. Выберем шаблон Empty Activity, который нам предлагается по умолчанию. Назовем наше приложение Му1Арр и выберем язык Java.



#### После создания проекта структура проекта Android отображается в следующем виде.

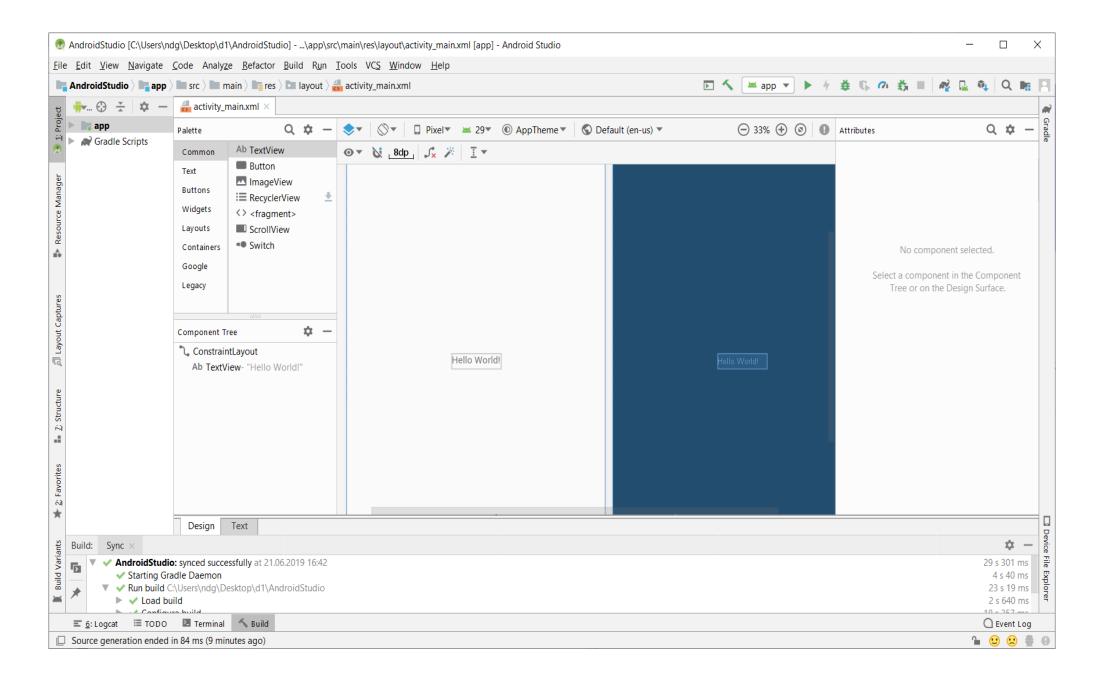


Проект Android может состоять из различных модулей. По умолчанию, когда мы создаем проект, создается один модуль – арр. Он имеет четыре подпапки:

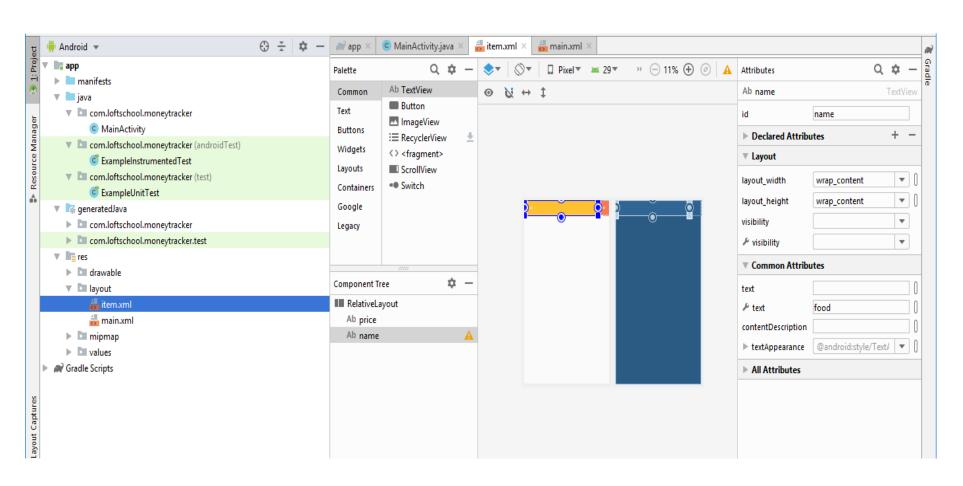
- 1) manifests: хранит файл манифеста AndroidManifest.xml, который определяет конфигурацию;
- 2) java: хранит файлы кода на языке java, которые структурированы по отдельным пакетам;
- 3) res: содержит используемые в приложении ресурсы;
- 4) generatedJava: содержит сгенерированные исходные файлы.

Отдельный элемент Gradle Scripts содержит ряд скриптов gradle (как для модуля арр или других возможных модулей, так и для всего проекта), которые используются при построении приложения.

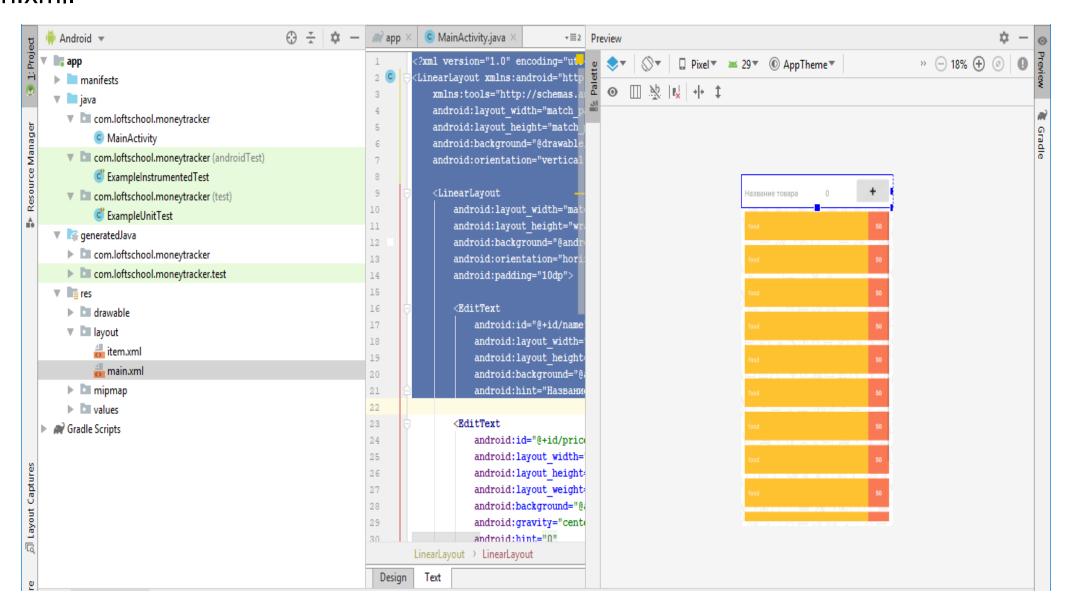
В студии по умолчанию должен быть открыт файл activity\_main.xml, который содержит определение графического интерфейса приложения — здесь размещаются компоненты. Стандартный проект уже содержит единственный размещенный компонент — TextView — с текстом "Hello world!". Соответственно, единственный функционал приложения — вывод этой текстовой строки на экране.



Создадим новое приложение, которое будет представлять собой пополняемый список покупок, состоящий из наименования товара и его количества. Для этого заполним экран компонентами, просто перенося их на форму, указав при этом им нужный цвет и текстовое содержание. Таким образом, графическое содержание будет определено файлом item.xml.



Графическое содержание после добавления элементов списка определено в файле main.xml.



#### Логика работы приложения описывается в файле MainActivity.java

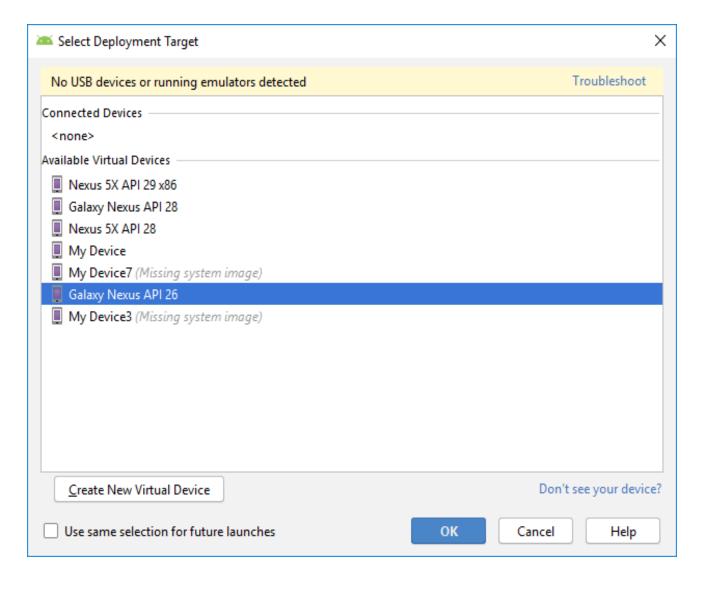
```
package com.loftschool.moneytracker;
import android.os.Bundle;
import android.support.annotation.NonNull;
import android.support.annotation.Nullable;
import android.support.v7.app.AppCompatActivity;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ListView;
import android.widget.TextView;
 * Copyright (c) 2017.
 * All Rights Reserved.
 * 
 * Author: Dmitry Gordeev <netimen@gmail.com>
 * Date: 05.04.17
 */
public class MainActivity extends AppCompatActivity {
    static class Item {
        String name;
        int price;
```

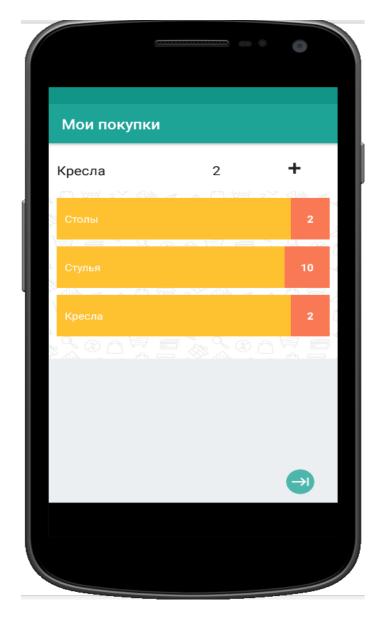
```
Item(String name, int price) {
           this.name = name;
           this.price = price;
   @Override
   protected void onCreate(@Nullable Bundle savedInstanceState) {
       super.onCreate(savedInstanceState);
       setContentView(R.layout.main);
       final EditText name = (EditText) findViewById(R.id.name);
       final EditText price = (EditText) findViewById(R.id.price);
       final Button add = (Button) findViewById(R.id.add);
       final ListView items = (ListView) findViewById(R.id.items);
       final ItemsAdapter adapter = new ItemsAdapter();
       items.setAdapter(adapter);
       add.setOnClickListener(new View.OnClickListener() {
       @Override
            public void onClick(View v) {
                 adapter.add(new Item(name.getText().toString(),
Integer.valueOf(price.getText().toString()));
        });
```

```
private class ItemsAdapter extends ArrayAdapter<Item> {
        ItemsAdapter() {
            super (MainActivity.this, R.layout.item);
        @NonNull
        @Override
        public View getView(int position, @Nullable View convertView, @NonNull ViewGroup parent) {
            final View view = getLayoutInflater().inflate(R.layout.item, null);
            final Item item = getItem(position);
            ((TextView) view.findViewById(R.id.name)).setText(item.name);
            ((TextView) view.findViewById(R.id.price)).setText(String.valueOf(item.price));
            return view;
```

Запустим проект, нажав на зеленую стрелочку на панели инструментов. Начнется построение проекта. Данный процесс может занять некоторое время, после чего отобразится диалоговое окно для выбора устройства для запуска. Здесь мы можем выбрать подключенный к компьютеру гаджет, либо эмулятор. В нашем случае выберем эмулятор Galaxy Nexus с образом системы Oreo API level 26 (Android 8.0).

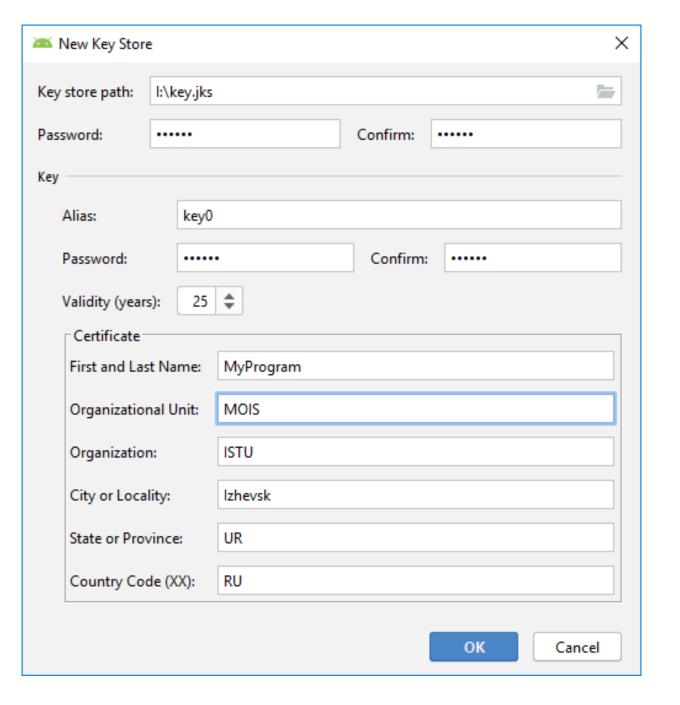
Выберем устройство и нажмем на кнопку ОК. И после запуска мы увидим наше приложение на экране устройства.





Теперь можно создать исполняемый файл-приложение для Android – apk-файл. Для этого необходимо сделать следующее.

- 1. Откроем окно Build Variants и напротив нашего модуля в поле Build Variant переключим режим сборки с debug на release.
- 2. В основном меню открываем Build → Generate Signed APK.
- 3. Открывается окно помощника генерации подписи для apk-файла. Все распространяемые приложения для Android должны иметь цифровую подпись автора, иначе приложение не установится на устройства пользователей. Сгенерированная подпись должна храниться в специальном файле-хранилище, расположенном на вашем компьютере, откуда потом она будет извлекаться для подписания apk-файлов. Для создания хранилища нажимаем кнопку Create New... и заполняем открывшиеся в окне поля:
- Путь для размещения файла хранилища
- Пароль и подтверждение для доступа к хранилищу
- Имя подписи, по которому она будет вызываться
- Пароль и подтверждение для доступа к подписи
- Срок действия подписи (по умолчанию 25 лет, оставляем без изменений)
- Хотя бы одно из полей сертификата. Обычно заполняют имя и фамилию, город и страну (RU).



- 4. Нажимаем ОК. Окно закрывается, и мы видим, что все поля в предыдущем окне автоматически заполнились введёнными нами данными. Ставим галочку в поле Remember Password, чтобы каждый раз не набирать пароль, и нажимаем ОК.
- 5. В следующем окне контролируем путь и имя apk-файла. По умолчанию оно равно имени модуля. Выбираем версию подписей V1 (подпись Jar) и V2 (полная подпись APK) при генерации подписанного apk в AndroidStudio. Нажимаем Finish и ждём, посматривая в строку состояния.
- 6. Когда сборка проекта закончится, отобразится окошко с предложением открыть папку с полученным арк-файлом. Открываем её и видим наш файл.

Мы можем переименовать файл, сохранив его расширение, и выложить в Play Market или на любой сайт или сразу загрузить на мобильное устройство. После загрузки на телефон/планшет достаточно нажать на него, и с помощью стандартного установщика пакетов приложение будет установлено.