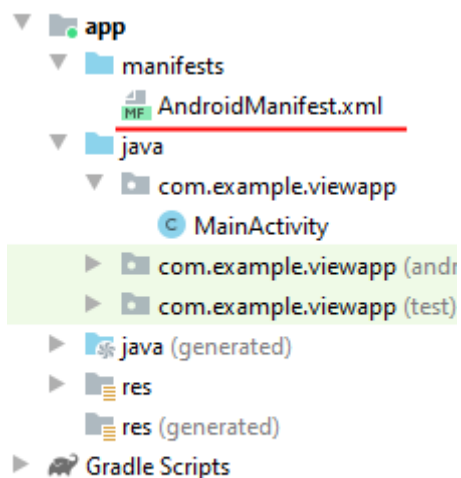




Лекция #17.1 Файл манифеста AndroidManifest.xml

Каждое приложение содержит файл манифеста **AndroidManifest.xml**. Данный файл определяет важную информацию о приложении - название, версию, иконки, какие разрешения приложение использует, регистрирует все используемые классы activity, сервисы и т.д. Данный файл можно найти в проекте в папке **manifests**:



Файл манифеста может выглядеть так:

```

1 <?xml version="1.0" encoding="utf-8"?>
2 <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3     package="com.example.viewapp">
4
5     <application
6         android:allowBackup="true"
7         android:icon="@mipmap/ic_launcher"
8         android:label="@string/app_name"
9         android:roundIcon="@mipmap/ic_launcher_round"
10        android:supportsRtl="true"
11        android:theme="@style/Theme.ViewApp">
12        <activity android:name=".MainActivity">
13            <intent-filter>
14                <action android:name="android.intent.action.MAIN" />
15
16                <category android:name="android.intent.category.LAUNCHER" />
17            </intent-filter>
18        </activity>
19    </application>
20
21 </manifest>

```

Элементом корневого уровня является узел `manifest`. В данном случае только определяется пакет приложения - `package="com.example.viewapp"`. Собственно это определение файла манифеста по умолчанию. В каждом конкретном случае может отличаться пакет приложения, остальное содержимое при создании проекта с пустой `activity` будет аналогичным.

Большинство настроек уровня приложения определяются элементом `application`. Ряд настроек задаются с помощью атрибутов. По умолчанию применяются следующие атрибуты:

- **android:allowBackup** указывает, будет ли для приложения создаваться резервная копия. Значение `android:allowBackup="true"` разрешает создание резервной копии.
- **android:icon** устанавливает иконку приложения. При значении `android:icon="@mipmap/ic_launcher"` иконка приложения берется из каталога **res/mipmap**
- **android:roundIcon** устанавливает круглую иконку приложения. Также берется из каталога **res/mipmap**
- **android:label** задает название приложение, которое будет отображаться на мобильном устройстве в списке приложений и в заголовке. В данном случае оно хранится в строковых ресурсах - `android:label="@string/app_name"`.

- **android:supportsRtl** указывает, могут ли использоваться различные RTL API - специальные API для работы с правосторонней ориентацией текста (например, для таких языков как арабский или фарси).
- **android:theme** устанавливает тему приложения. Подробно темы будут рассмотрены далее, а пока достаточно знать, что тема определяет общий стиль приложения. Значение `@style/Theme.ViewApp` берет тему "Theme.ViewApp" из каталога **res/values/themes**

Вложенные элементы `activity` определяют все используемые в приложении `activity`. В данном случае видно, что в приложении есть только одна `activity` - `MainActivity`.

```

1 <activity android:name=".MainActivity">
2     <intent-filter>
3         <action android:name="android.intent.action.MAIN" />
4
5         <category android:name="android.intent.category.LAUNCHER" />
6     </intent-filter>
7 </activity>

```

Элемент `intent-filter` в `MainActivity` указывает, как данная `activity` будет использоваться. В частности, с помощью узла `action` `android:name="android.intent.action.MAIN"`, что данная `activity` будет входной точкой в приложение и не должна получать какие-либо данные извне.

Элемент `category android:name="android.intent.category.LAUNCHER"` указывает, что `MainActivity` будет представлять стартовый экран, который отображается при запуске приложения.

Файл манифеста может содержать множество элементов, которые имеют множество атрибутов. И все возможные элементы, и их атрибуты можно найти в документации. Здесь же рассмотрим некоторые примеры использования.

Определение версии

С помощью атрибутов элемента `manifest` можно определить версию приложения и его кода:

```

1 <?xml version="1.0" encoding="utf-8"?>
2 <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3     package="com.example.viewapp"
4     android:versionName="1.0"
5     android:versionCode="1">
6
7     <!-- остальное содержимое -->
8
9 </manifest>

```

Атрибут `android:versionName` указывает на номер версии, который будет отображаться пользователю и на которую будут ориентироваться пользователи при работе с приложением.

Тогда как атрибут `android:versionCode` представляет номер версии для внутреннего использования. Этот номер только определяет, что одна версия приложения более новая, чем какая-то другая с меньшим номером версии. Этот номер не отображается пользователям.

При желании мы также можем определить версию в ресурсах, а здесь ссылаться на ресурс.

Установка версии SDK

Для управления версией android sdk в файле манифеста определяется элемент `<uses-sdk>`. Он может использовать следующие атрибуты:

- `minSdkVersion`: минимальная поддерживаемая версия SDK
- `targetSdkVersion`: оптимальная версия
- `maxSdkVersion`: максимальная версия

Версия определяется номером API, например, Jelly Beans 4.1 имеет версию 16, а Android 11 имеет версию 30:

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3     package="com.example.viewapp"
4     android:versionName="1.0"
5     android:versionCode="1">
6     <uses-sdk android:minSdkVersion="22" android:targetSdkVersion="30" />
7     <!-- остальное содержимое -->
8
9 </manifest>
```

Установка разрешений

Иногда приложению требуются разрешения на доступ к определенным ресурсам, например, к списку контактов, камере и т.д. Чтобы приложение могло работать с тем же списком контактов, в файле манифесте необходимо установить соответствующие разрешения. Для установки разрешений применяется элемент `<uses-permission>`:

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3     package="com.example.viewapp">
4     <uses-permission android:name="android.permission.READ_CONTACTS" />
5     <uses-permission android:name="android.permission.CAMERA" android:maxSdkVersion="30" />
6     <!-- остальное содержимое -->
7
8 </manifest>
```

Атрибут `android:name` устанавливает название разрешения: в данном случае на чтение списка контактов и использование камеры. Опционально можно установить максимальную версию `sdk` посредством атрибута `android:maxSdkVersion`, который принимает номер API.

Поддержка разных разрешений

Мир устройств Android очень сильно фрагментирован, здесь встречаются как гаджеты с небольшим экраном, так и большие широкоэкранные телевизоры. И бывают случаи, когда надо ограничить использование приложения для определенных разрешений экранов. Для этого в файле манифеста определяется элемент `<supports-screens>`:

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3     package="com.example.viewapp">
4
5     <supports-screens
6         android:largeScreens="true"
7         android:normalScreens="true"
8         android:smallScreens="false"
9         android:xlargeScreens="true" />
10 <!-- остальное содержимое -->
11
12 </manifest>
```

Данный элемент принимает четыре атрибута:

- `android:largeScreens` - экраны с диагональю от 4.5 до 10"
- `android:normalScreens` - экраны с диагональю от 3 до 4.5"
- `android:smallScreens` - экраны с диагональю меньше 3"
- `android:xlargeScreens` - экраны с диагональю больше 10"

Если атрибут имеет значение `true`, то приложение будет поддерживаться соответствующим размером экрана

Запрет на изменение ориентации

Приложение в зависимости от положения гаджета может находиться в альбомной и портретной ориентации. Не всегда это бывает удобно. Мы можем сделать, чтобы приложение вне зависимости от поворота гаджета использовало только одну ориентацию. Для этого в файле манифеста у требуемой `activity` надо установить атрибут **`android:screenOrientation`**. Например, запретим альбомную ориентацию:

```

1 <?xml version="1.0" encoding="utf-8"?>
2 <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3     package="com.example.viewapp"
4     android:versionName="1.0"
5     android:versionCode="1" >
6
7     <application
8         android:allowBackup="true"
9         android:icon="@mipmap/ic_launcher"
10        android:label="@string/app_name"
11        android:roundIcon="@mipmap/ic_launcher_round"
12        android:supportsRtl="true"
13        android:theme="@style/Theme.ViewApp">
14        <activity android:name=".MainActivity"
15
16            android:screenOrientation="portrait">
17
18            <intent-filter>
19                <action android:name="android.intent.action.MAIN" />
20
21                <category android:name="android.intent.category.LAUNCHER" />
22            </intent-filter>
23        </activity>
24    </application>
25 </manifest>

```

Значение `android:screenOrientation="portrait"` указывает, что данная activity будет находиться только в портретной ориентации. Если же надо установить только альбомную ориентацию, тогда надо использовать значение `android:screenOrientation="landscape"`.