



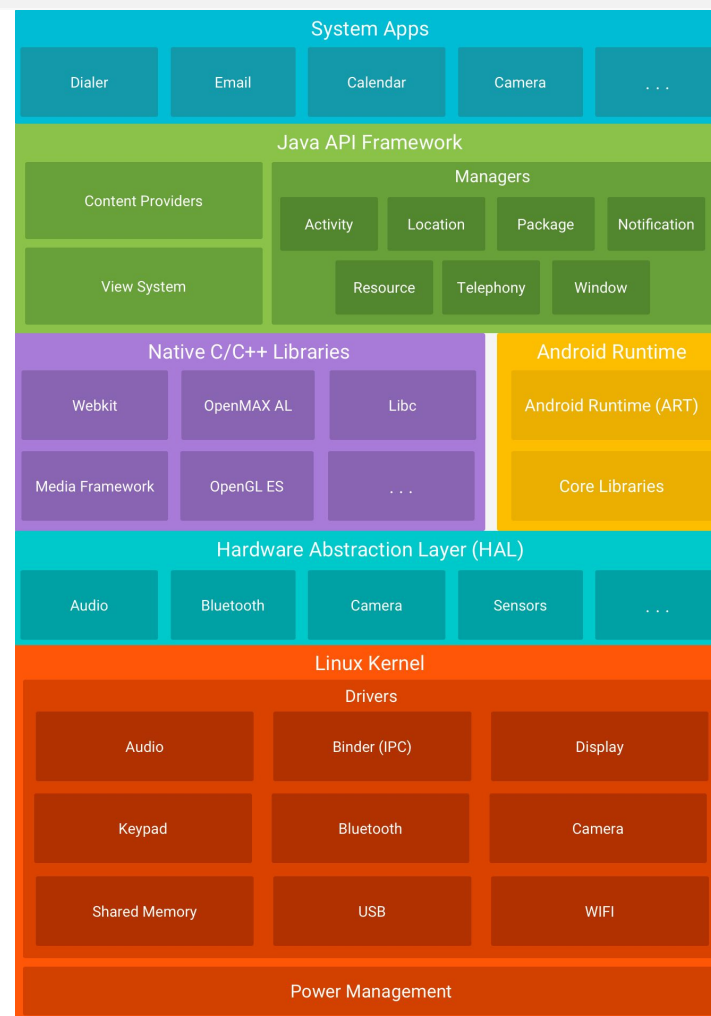
# Введение. Знакомство с Android.

Лекция №1

A horizontal line extends from the right side of the text "Лекция №1", followed by a downward-pointing arrow that leads to the author's name.

Попов Даниил

# Архитектура ОС Android



# Приложения в Android

---



- Приложения поставляются в виде скомпилированных арк-файлов;
- Каждое установленное приложение - отдельный пользователь в контексте Linux;
- Приложение “живет” в своем собственном процессе;
- Каждый процесс имеет свою изолированную JVM.



# Android SDK и Api Level



Platform Version	API Level	VERSION_CODE	Notes
Android 7.0	24	N	Platform Highlights
Android 6.0	23	M	Platform Highlights
Android 5.1	22	LOLLIPOP_MR1	Platform Highlights
Android 5.0	21	LOLLIPOP	
Android 4.4W	20	KITKAT_WATCH	KitKat for Wearables Only
Android 4.4	19	KITKAT	Platform Highlights
Android 4.3	18	JELLY_BEAN_MR2	Platform Highlights
Android 4.2, 4.2.2	17	JELLY_BEAN_MR1	Platform Highlights
Android 4.1, 4.1.1	16	JELLY_BEAN	Platform Highlights
Android 4.0.3, 4.0.4	15	ICE_CREAM_SANDWICH_MR1	Platform Highlights
Android 4.0, 4.0.1, 4.0.2	14	ICE_CREAM_SANDWICH	
Android 3.2	13	HONEYCOMB_MR2	
Android 3.1.x	12	HONEYCOMB_MR1	Platform Highlights
Android 3.0.x	11	HONEYCOMB	Platform Highlights
Android 2.3.4	10	GINGERBREAD_MR1	Platform Highlights
Android 2.3.3			
Android 2.3.2	9	GINGERBREAD	
Android 2.3.1			
Android 2.3			
Android 2.2.x	8	FROYO	Platform Highlights
Android 2.1.x	7	ECLAIR_MR1	Platform Highlights
Android 2.0.1	6	ECLAIR_0_1	
Android 2.0	5	ECLAIR	
Android 1.6	4	DONUT	Platform Highlights
Android 1.5	3	CUPCAKE	Platform Highlights
Android 1.1	2	BASE_1_1	
Android 1.0	1	BASE	

Version	Codename	API	Distribution
2.2	Froyo	8	0.1%
2.3.3 - 2.3.7	Gingerbread	10	1.7%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	1.6%
4.1.x	Jelly Bean	16	6.0%
4.2.x		17	8.3%
4.3		18	2.4%
4.4	KitKat	19	29.2%
5.0	Lollipop	21	14.1%
5.1		22	21.4%
6.0	Marshmallow	23	15.2%



# Структура Android-приложения



- Исходный код приложения
- Исходный код *Android* и *Unit* тестов
- Ресурсы
- Сторонние и нативные библиотеки
- *AndroidManifest.xml*
- *build.gradle*
- *proguard-rules.pro*



```
▼ app
  ► build
  ► libs
  ▼ src
    ► androidTest
    ▼ main
      ▼ java
        ▼ com.example.dpopov.androidtest
          ◉ Blurshiter
          ◉ MainActivity
      ▼ res
        ► drawable
        ► layout
        ► mipmap-hdpi
        ► mipmap-mdpi
        ► mipmap-xhdpi
        ► mipmap-xxhdpi
        ► mipmap-xxxhdpi
        ► values
        ► values-w820dp
        ► AndroidManifest.xml
      ► test
    .gitignore
    app.iml
    build.gradle
    proguard-rules.pro
```

# Компоненты приложения

---



## 1. ***Activity***

Одиночный экран с пользовательским интерфейсом

## 2. ***Service***

Компонент без пользовательского интерфейса для выполнения длительных фоновых задач

## 3. ***Content Provider***

Обертка над хранилищем данных, как правило предоставляющая доступ другим приложениям

## 4. ***Broadcast Receiver***

Приемник широковещательных сообщений, рассылаемых системой или другими приложениями

***Intent*** - “сообщение” для взаимодействия компонентов приложения между собой и системой.

Обычно используются для:

1. Запуска *Activity*;
2. Запуска *Service*;
3. Рассылки широковещательных сообщений.

***Intent Filter*** - информация для системы о том, какие *Intent*'ы может обрабатывать данный компонент

# Содержимое Intent'a



- Имя компонента-получателя (*Component Name*)
  - Явные Intent'ы (*Explicit*)
  - Неявные Intent'ы (*Implicit*)
- Требуемое действие (*Action*)
  - `ACTION_VIEW`, `ACTION_SEND` и т.д.
- Данные (*Data*)
- Категория (*Category*)
  - `CATEGORY_LAUNCHER`, `CATEGORY_BROWSABLE` и т. д.
- Дополнительная информация (*Extras*)
- Флаги (*Flags*)





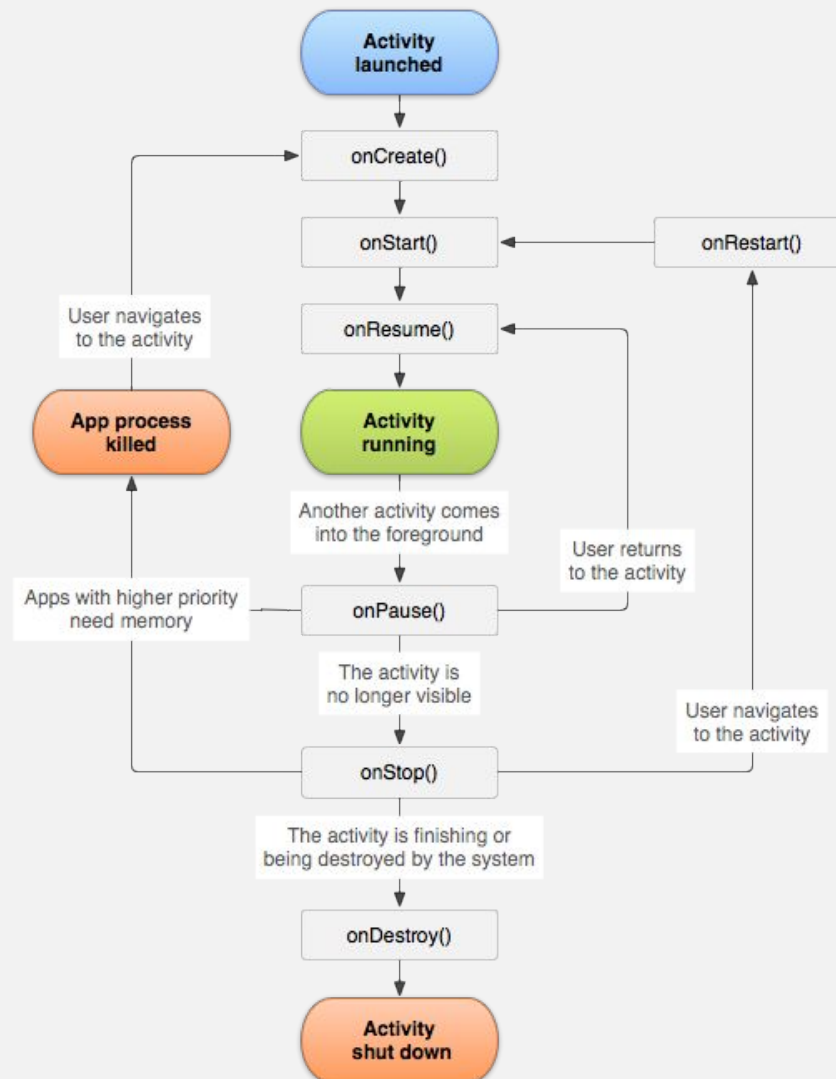
**Activity** - “экран” для отображения *GUI* и для взаимодействия с пользователем.

Для создания *Activity* необходимо:

1. Создать класс, унаследованный от *Activity*;
2. Реализовать необходимые методы;
3. Добавить описание в *Android Manifest*.

```
<activity android:name=".ExampleActivity" android:icon="@drawable/app_icon">  
    <intent-filter>  
        <action android:name="android.intent.action.MAIN" />  
        <category android:name="android.intent.category.LAUNCHER" />  
    </intent-filter>  
</activity>
```

# Жизненный цикл Activity

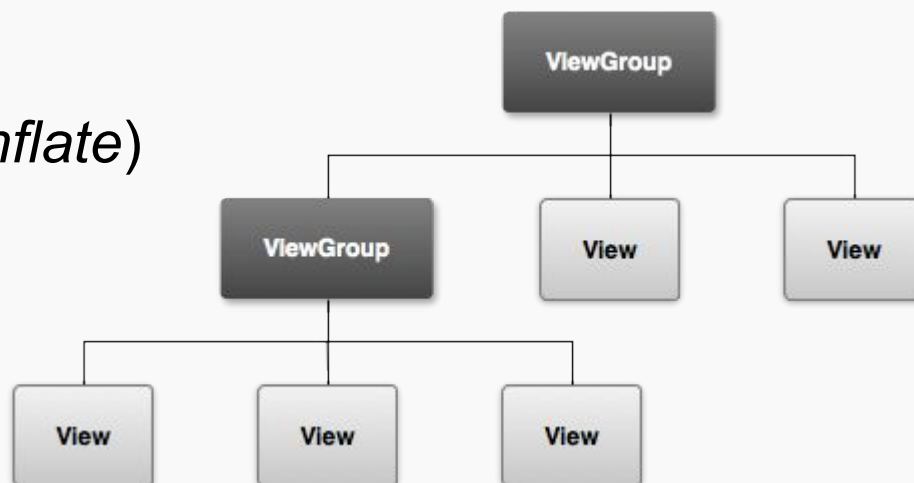


**View** - прямоугольная область экрана, отображающая информацию пользователю и взаимодействующая с НИМ.

**ViewGroup** - несколько объединенных вместе *View*.

Создание интерфейса:

1. Через *XML*-разметку (*Inflate*)
2. Программно



**Service** - компонент приложения для выполнения фоновых задач, не имеющий пользовательского интерфейса.

Для создания сервиса необходимо:

1. Создать класс, унаследованный от *Service*;
2. Реализовать необходимые методы;
3. Добавить описание в *Android Manifest*.

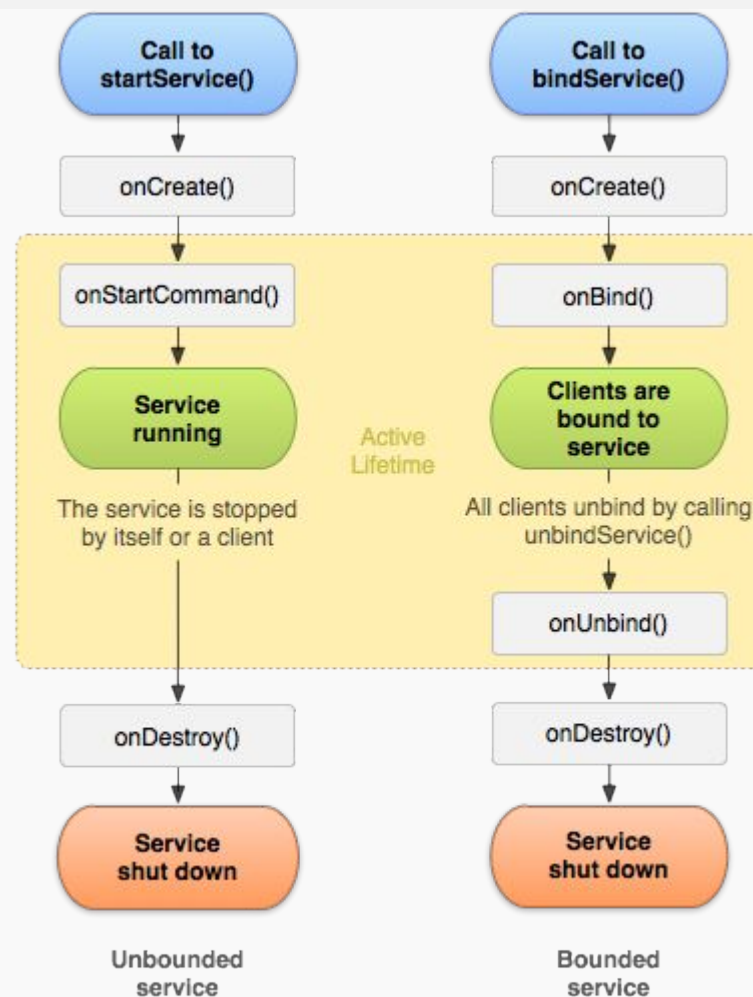
```
<service  
    android:name=".TestService"  
    android:exported="false"/>
```

# Жизненный цикл сервиса



Два типа сервисов:

1. *Started*
2. *Bound*



# Широковещательные рассылки



Рассылки бывают:

1. Глобальные
  - a. Обычные (*Normal*)
  - b. Упорядоченные (*Ordered*)
2. Локальные
  - a. Синхронные
  - b. Асинхронные



***BroadcastReceiver*** - приемник широковещательных сообщений.

# Создание BroadcastReceiver'a



1. Унаследовать класс *BroadcastReceiver*;
2. Имплементировать абстрактный метод *onReceive()*;
3. Зарегистрировать в *Android Manifest*

```
<receiver  
    android:name=".BatteryBroadcastReceiver"  
    android:exported="false">  
    <intent-filter>  
        <action android:name="android.intent.action.BATTERY_LOW"/>  
    </intent-filter>  
</receiver>
```

или программно

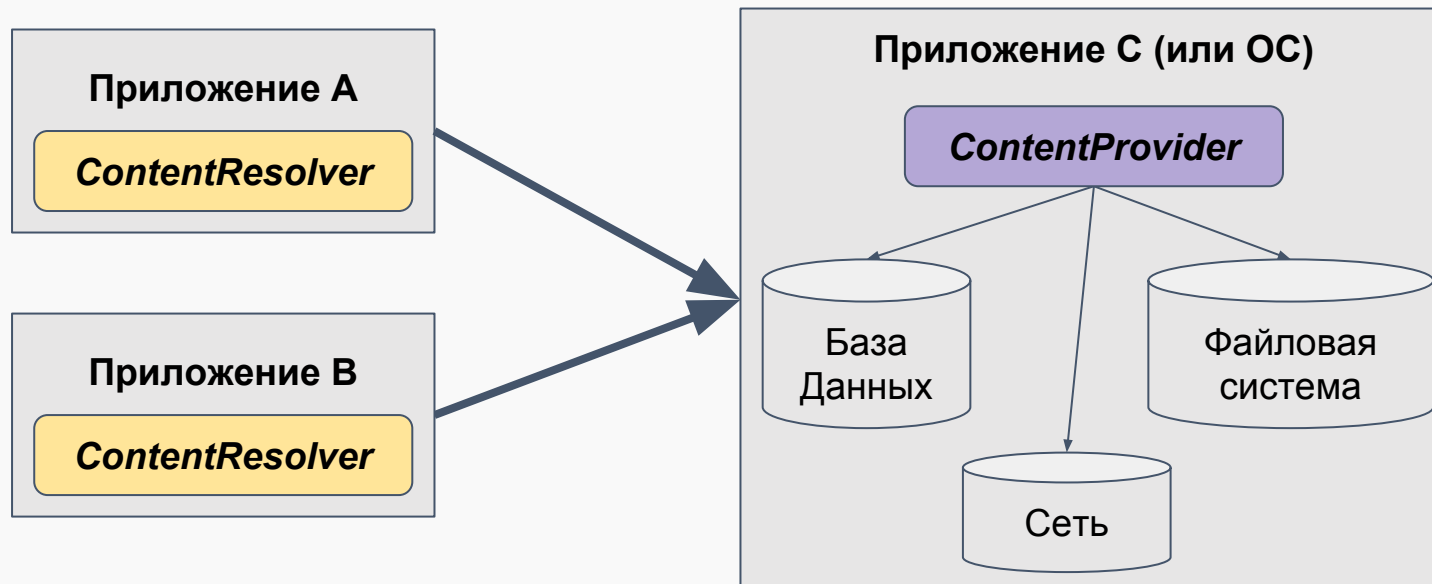
```
BroadcastReceiver receiver = new BatteryBroadcastReceiver();  
IntentFilter filter = new IntentFilter();  
filter.addAction(Intent.ACTION_BATTERY_LOW);  
registerReceiver(receiver, filter);  
// ...  
unregisterReceiver(receiver);
```



# ContentProvider и данные



***ContentProvider*** - абстракция для доступа к структурированным данным из сторонних приложений.





# AndroidManifest.xml



- Имя пакета - уникальный идентификатор приложения (*org.telegram.messenger*);
- Версия приложения и его название;
- Компоненты приложения: *Activity*, *Service*, *BroadcastReceiver* и *ContentProvider*;
- Разрешения;
- Минимальная поддерживаемая версия *Android*;
- Требования к аппаратному обеспечению;
- Настройки приложения.



# Пример AndroidManifest.xml



```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.dpopov.androidtest">

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

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="AndroidTest"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>

        <receiver
            android:name=".BatteryBroadcastReceiver">
            <intent-filter>
                <action android:name="android.intent.action.BATTERY_LOW" />
            </intent-filter>
        </receiver>
    </application>
</manifest>
```



**Спасибо за внимание!**

**А теперь к практике!**