

# Введение



# Информация о курсе

- Лекции

12 занятий

- Лабораторные работы

5 работ

- Проект
- Экзамен

нет письменного экзамена

Проект vs лабораторные работы 20% за каждую

- Пересдачи



# Цели курса

- Обзорный курс по мобильной платформе
- Необходимые знания:

Java (!?)

UI

XML

Databases and SQL

- Структура лекций

- UI

- Процессы

- Хранилища

- Сервисы, виджеты



# Android – что это ?

- **Android** – ОС для мобильных устройств с открытым исходным кодом, основанная на ядре Linux
- **Цель** – интегрировать усилия операторов мобильной связи, производителей карманных устройств, разработчиков приложений и в итоге – быстрее вводить новые технологии и качественно реагировать на нужды потребителей в области мобильной связи.



# Android – ЧТО ЭТО ?

Android is an operating system

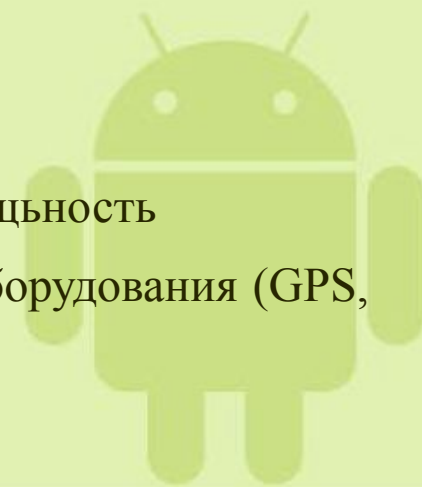
- “Программный стек для мобильных устройств, который включает в себя операционную систему, связующее ПО и ключевые приложения.”
- По существу, единая платформа для разработки приложений, которые будут работать на разных устройствах.

Android Target Machines

- 2.x smart phones
- 3.x also tablets, net-books, and ebook readers
- 4.x (released 2011) tablets, books, smart phones
- 5.x (released 2014) TVs, robots, fridges, watches

Target Machine Properties

- ограничения мощности батареи, размер экрана, вычислительная мощность
- дополнительные возможности: большое количество встроенного оборудования (GPS, Telephone, GPS, compass, movement/orientation sensors)



# Android – ЧТО ЭТО ?

Языки программирования:

- C, C++(NDK);
- C# (Xamarin, dot42);
- Java (Android SDK);
- Scripting Layer for Android

Множество повторно используемых компонент:

- доступно множество стандартных библиотек java;
- android.\* - специфические библиотеки для работы с Android API, UI, SQLite, audio/video, камерой, акселерометром, GPS, ...
- сторонние библиотеки (2/3D – графика OpenGL ES, OCR, TTS, FreeType, SSL, SMTP/POP3, ...)

Как находить ответы на вопросы

- Общая концепция объектов ОС
- [Developer.android.com](http://Developer.android.com)
- Общий поиск
- [www.stackoverflow.com](http://www.stackoverflow.com)



# Оснастка смартфона

- GSM(2G), WCDMA (3G), HSDPA(3,5G), Wi-Fi, Bluetooth, NFC (~10 см.), GPS

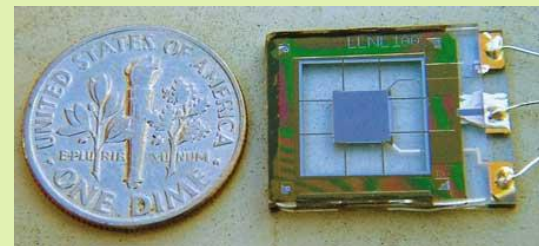
- сенсоры:

- **движение** – силы ускорения и вращения

по трем осям: акселерометр ( $\text{м/с}^2$ , встряски, наклоны), гироскоп (рад/с, вращения, повороты), ...

- **среда** – освещение, температура, давление: фотометр, термометр, барометр, сенсор приближения

- **позиционирование** – физическое положение устройства: сенсор ориентации, магнитометр



[http://developer.android.com/guide/topics/sensors/sensors\\_overview.html](http://developer.android.com/guide/topics/sensors/sensors_overview.html)

# История платформы Android

## Android - 22-месячный стартап

### Авторы :

**Andy Rubin** (соучред. Danger, Inc – ПО, архитектура и сервисы для моб. устройств)

**Rich Miner** (соучред. Wildfire Communications, Inc. – элементы голосового управления для мобильных телефонов: набор сообщений, совершать звонки, управлять входящими звонками и контактами ...)

**Nick Sears** (вице-президент T-Mobile – мобильный оператор США)

**Chris White** (ведущий разработчик дизайна и интерфейса в WebTV)



**Andy Rubin** – старший вице-президент мобильного отдела Google с 2005 г.





# История платформы Android

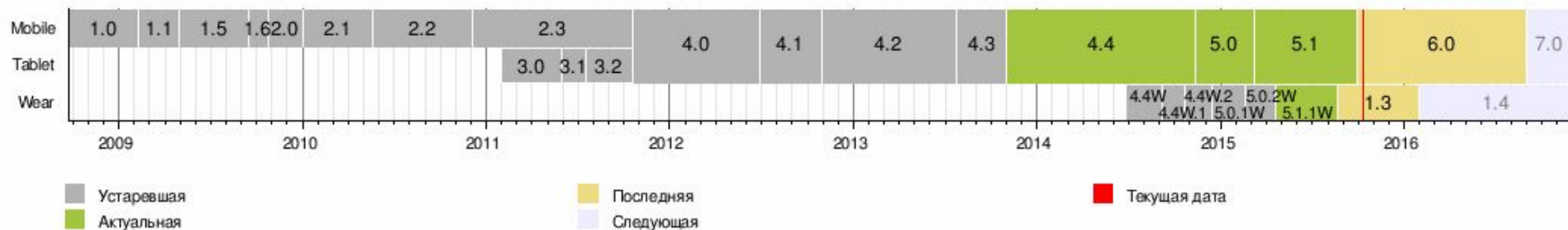
## Развитие:

**2003 г.** – основание Android Inc.(цель – создать «...smarter mobile devices that are more aware of its owner's location and preferences» ).

**2005 г.** – Google приобретает Android Inc. («We acquired Android because of the talented engineers and great technology. We're thrilled to have them here»). Под руководством **Andy Rubin** разрабатывается **мобильная платформа** на основе Linux-ядра.

**2007 г.** – Представление **Open Handset Alliance** (консорциума по разработке открытых стандартов для мобильных устройств из 34 (48) компаний: Broadcom, Google, HTC, Intel, LG, Motorola, NVidia, Samsung, T-Mobile, Asus, Sony Ericsson, Toshiba, ...). Выпуск первой версии Android и SDK к ней.

## Обновления версий Android



# Перспективы программирования под Android

## Android Market Shares - Smart Phone Sales

I 2010: Sold devices 296 millions

Nokia 37.6%, Android 22.7% , Blackberry 16.0%, iPhone 15.7%

I 2011: Sold devices 486 millions

Android 38.5%, iPhone 19.4%, Nokia 19.2%, Blackberry 13.1%

I 2012: Sold devices 722 millions

Android 70.1%, iPhone 21.0% , Blackberry 3.2%, Windows Phone 2.6 %

I 2012: Sold devices 722 millions

Android 79.1%, iPhone 21.0% , Blackberry 3.2%, Windows Phone 2.6 %

I 2013: Sold devices 967 millions

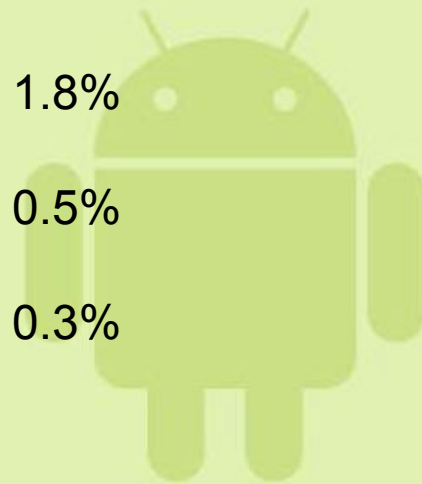
Android 78.4%, iPhone 17.8% , Windows Phone 3.2%, Blackberry 1.8%

I 2014 : Sold devices 1.3 billion

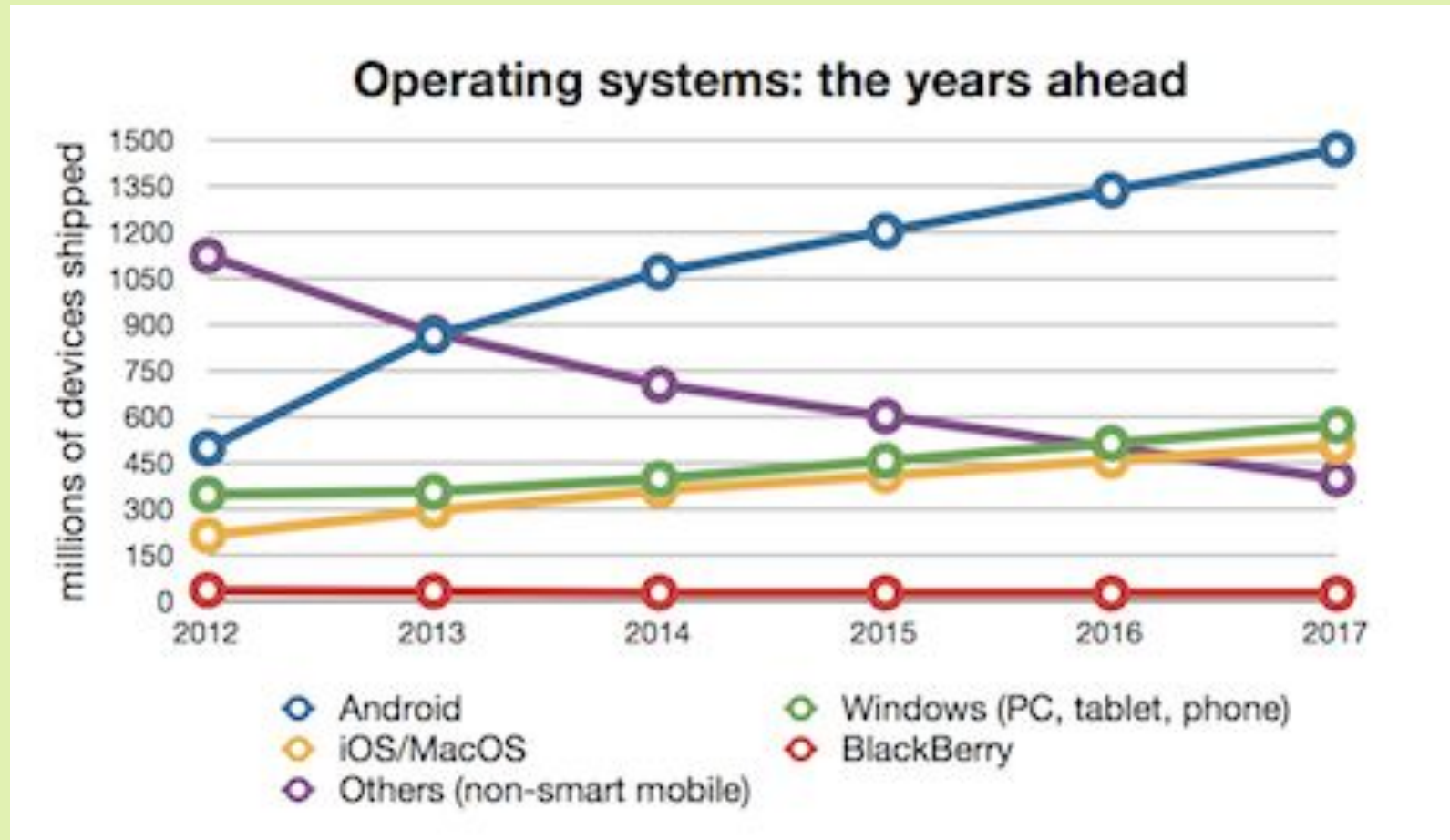
Android 81.2%, iPhone 15.2% , Windows Phone 2.5%, Blackberry 0.5%

I 2015: Quarter 1, Sold devices 334 millions

Android 78.0%, iPhone 18.3% , Windows Phone 2.7%, Blackberry 0.3%

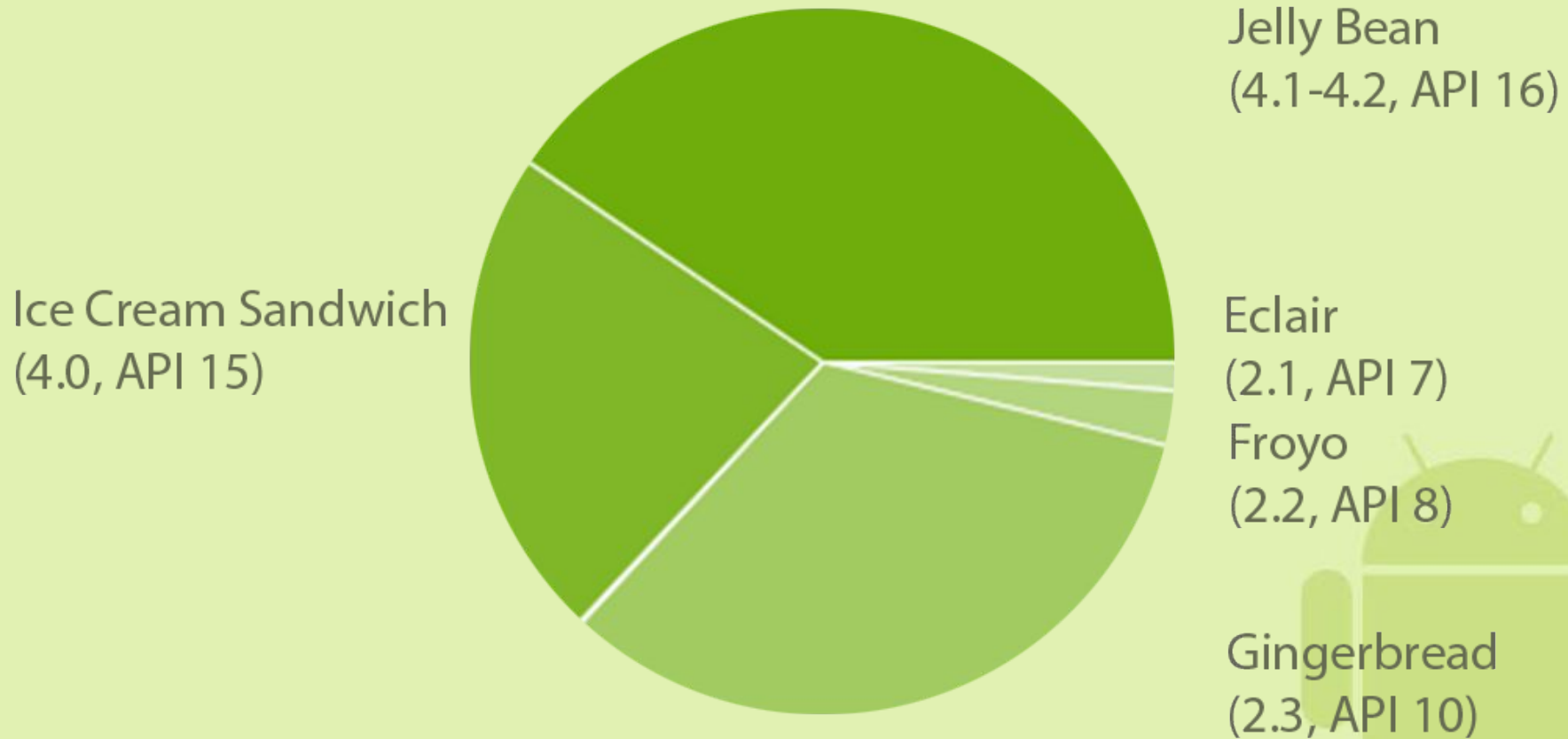


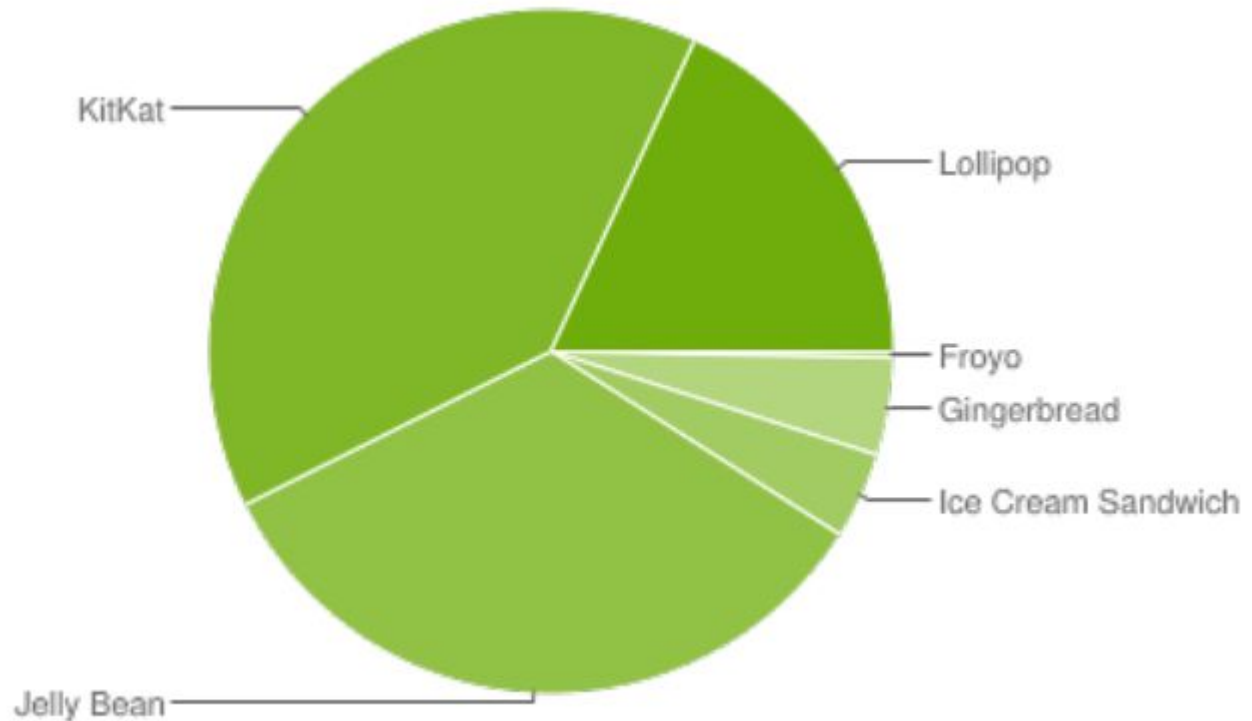
# Перспективы платформы Android



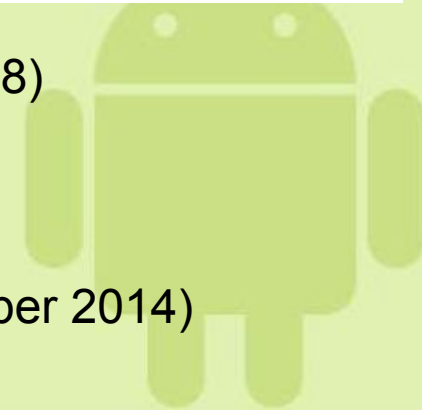
Gartner Inc

# Популярность различных версий Android

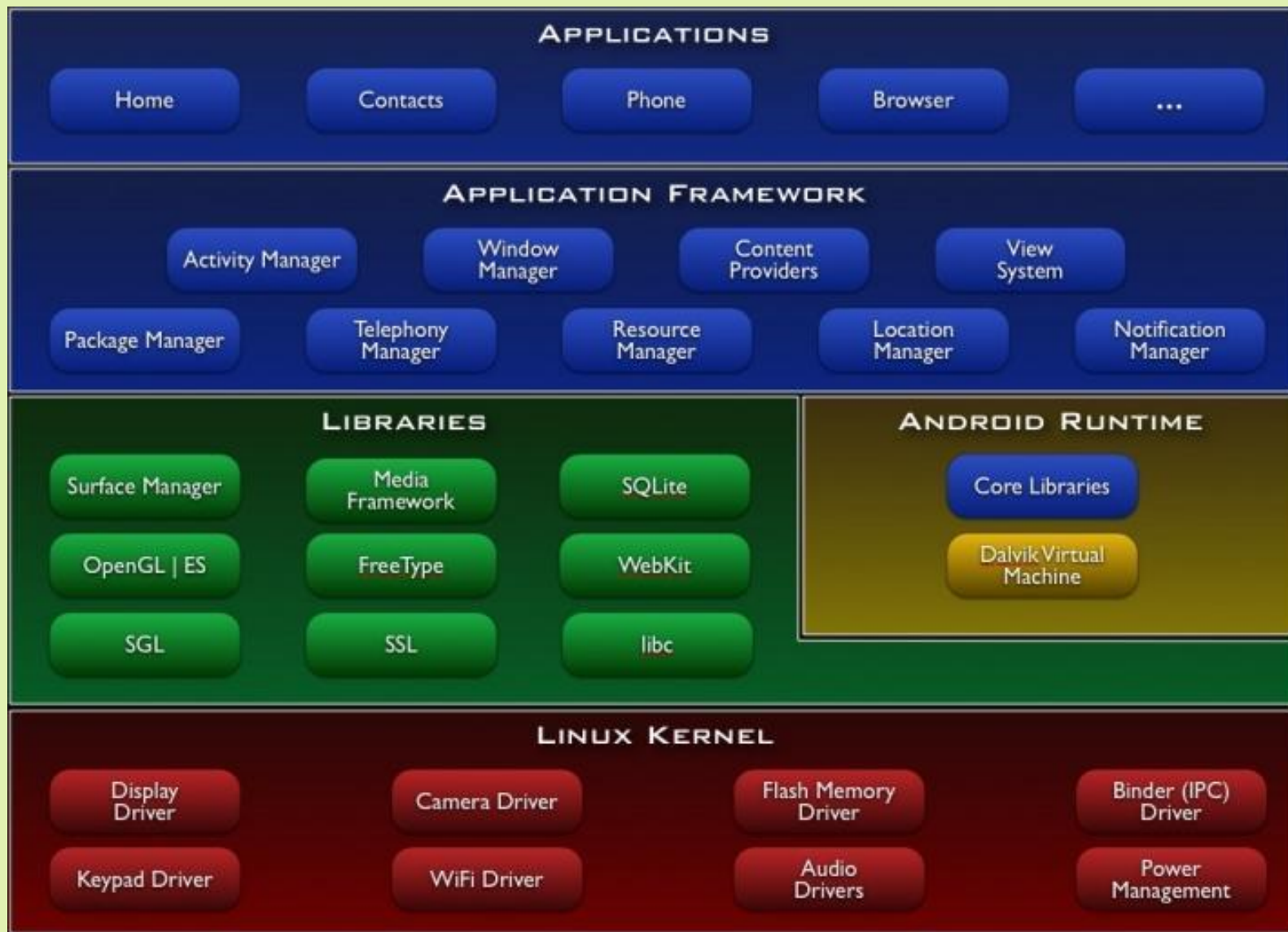




- Version 1.X: Targets smart phones, now outdated (September 2008)
- Version 2.X: Targets smart phones, rather stable (October 2009)
- Version 3.X: Targets tablets (February 2011)
- Version 4.X: Both smart phones and tablets (October 2011)
- Version 5.X: Support for watches, wearables, TV, cars, ... (November 2014)



# Android ИЗНУТРИ



# Среда исполнения приложения



- приложение - отдельный процесс.
- собственный экземпляр **регистр**-ориентированной виртуальной машины Dalvik.

```
move r0, v1
move r1, v2
add-int r2,r1,r0
move v3, r2
```

**Dalvik VM** преобразует классы, скомпилированные компилятором Java, в `.dex` формат, оптимизированный для использования минимального объема памяти, и исполняет их.



# Почему DalvikVM – регистровая машина?

- Стековые машины медленнее регистровых

- Больше операций с памятью

X+5

LOAD X to STACK

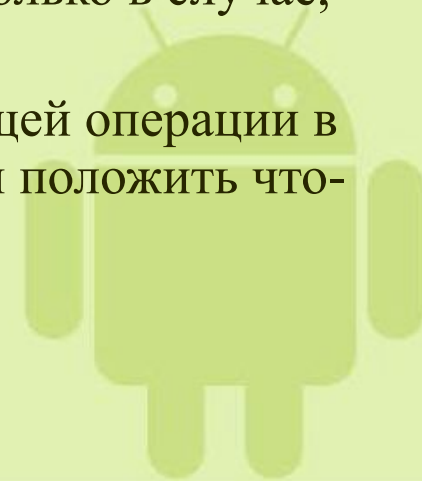
LOAD 5 to STACK

ADD ; pop 2 times from stack and push a result back to the stack

POP X

Целых 6 операций с памятью (на 3 операции больше, чем в регистровой машине)

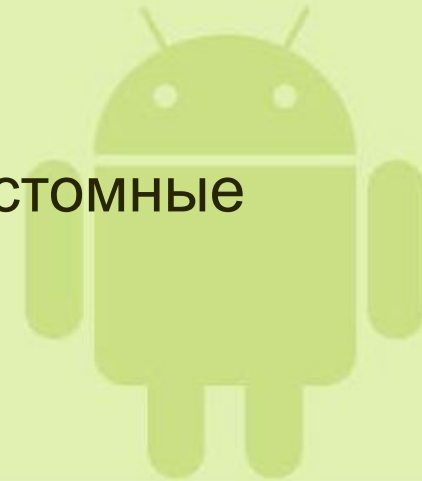
- Использование вычисленного подвыражения выгодно только в случае, если вычисление дороже считывания из памяти
- Невозможность запустить загрузку данных для следующей операции в параллель с вычислением текущей из-за невозможности положить что-то поверх вершины стека.





# Структура Android приложения

- **Activity** – единичная сущность, исполняемая приложением, обычно, представляющая пользователю посредством View (экранов).
- **ContentProvider** – позволяет сохранять структурированные (табличные) данные и делать их доступными для других приложений.
- **Service** – выполняют длительную работу «фоново» (в основном потоке приложения), может быть остановлен системой из-за нехватки ресурсов
- **BroadcastReceiver** – получает системные или кастомные события от ОС и других приложений.



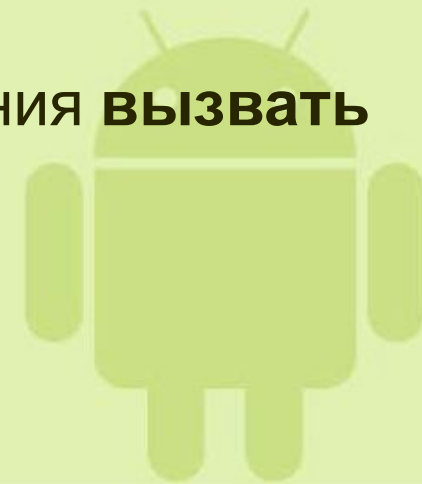
# Пример для музыкального плеера

- **Activity** - UI для просмотра плейлистов и выбора песен
- **ContentProvider** – сохраняет данные о плейлистах и песнях, предоставляет их вашему и любым другим приложениям
- **Service** – проигрывает музыку, поддерживает фоновую игру
- **Broadcast Receiver** – останавливает проигрывание, если поступает входящий звонок



# Понятие Intent (намерение) в Android

- Intent – абстрактное описание для операции, которую нужно выполнить («задача»). Определяется через:
  - **ACTION** – имя действия
  - **DATA** – над чем действие выполняется (mimeType, scheme, port, path, )
  - **COMPONENT** – кто должен выполнить действие
- Например, “Мальчик ударяет мяч”  
Выделите в этой операции Action, Data, Component?
- **Одному** Activity/Service/BroadcastReceiver **может** соответствовать **один или более Intent**.
- Через **Intent** можно из любого места приложения **вызвать** **нужную Activity**



# Поиск подходящей Activity по неявному Intent

- *Intent-фильтры ассоциируются с каждым компонентом*
  - Описывают Intent-ы, которые способен выполнять компонент

## Сценарий использования Intent

- Приложение вызывает `startActivity()` с параметрами  
`ACTION_VIEW` и `http://*`
- Для каждой зарегистрированной в системе Activity:
  - Поддерживает операцию `ACTION_VIEW`?
  - Поддерживает адреса вида `http://*`?



# Пример Intent -ОВ

- **ACTION\_VIEW** <http://www.cornell.edu>Launch  
Указание открыть Activity, умеющую отображать страницу по URL для отображения указанной страницы
- **ACTION\_DIAL** <tel:5551234>  
Указание запустить Activity, умеющую звонить, которая позвонит по заданному номеру.

**Внимание!** В примерах не указано, кто именно должен выполнить действие.  
Это *implicit* (неявные) *Intents*

*Explicit intents* включают в описание также и указание конкретного исполнителя



# Поиск подходящей Activity по неявному Intent

- *Intent-фильтры ассоциируются с каждым компонентом*
  - Описывают Intent-ы, которые способен выполнять компонент

## Сценарий использования Intent

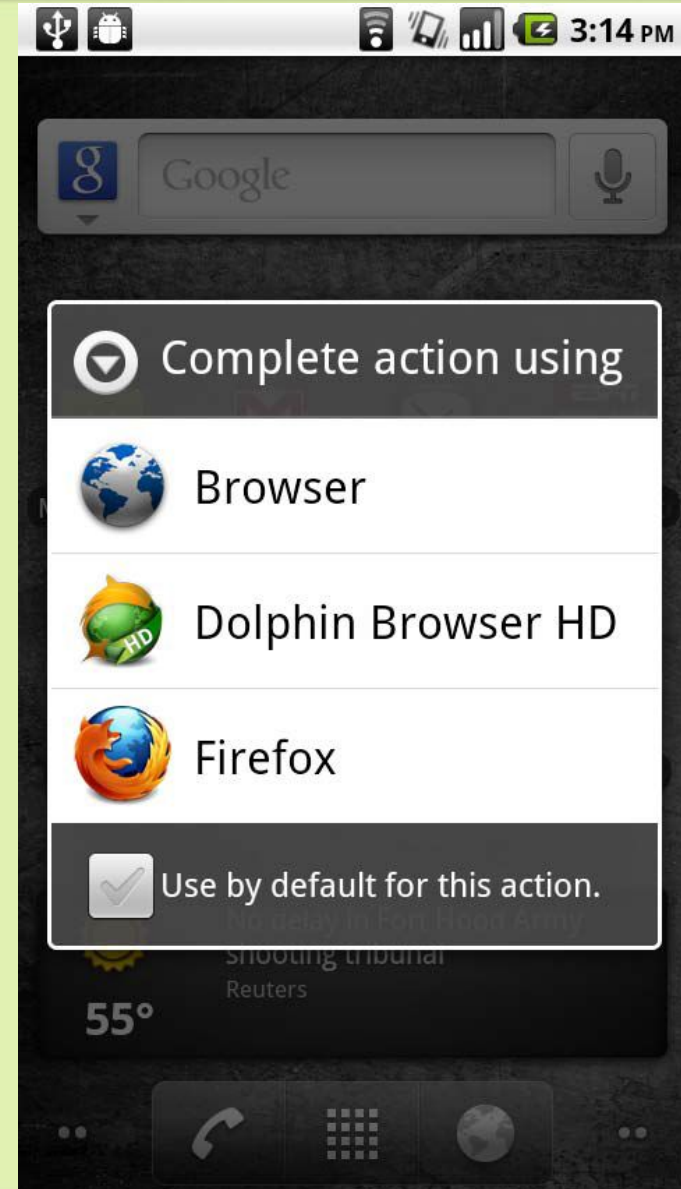
- Приложение вызывает `startActivity()` с параметрами  
`ACTION_VIEW` и `http://*`
- Для каждой зарегистрированной в системе Activity:
  - Поддерживает операцию `ACTION_VIEW`?
  - Поддерживает адреса вида `http://*`?



# Поиск подходящей Activity по неявному Intent

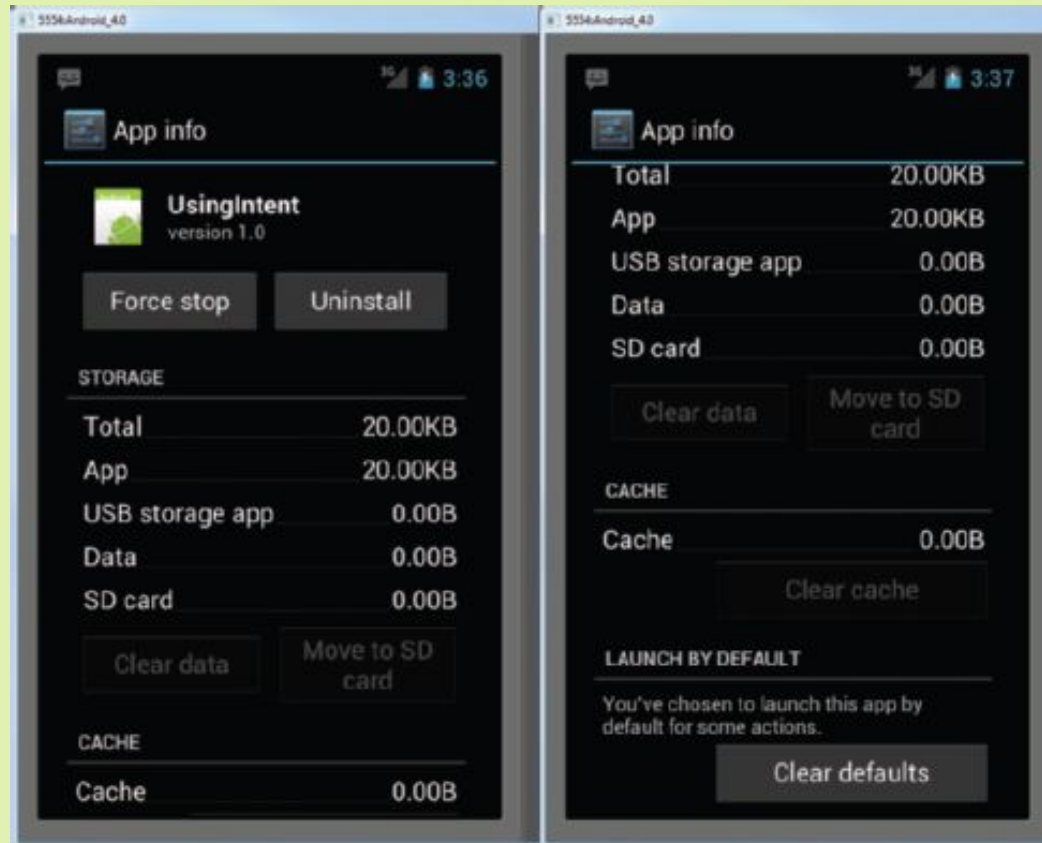
- Если фильтру удовлетворяет только один компонент – запустить его.
- Иначе – сделать выбор с помощью пользователя или на основе приоритета. Системные компоненты имеют самый низкий приоритет.
- Стандартизированные Intents делают Android расширяемым:

*«создайте приложение, получающее на вход картинки, – галерея автоматически позволит посылать в него изображения»*



# Сброс defaults-настроек приложения

- Android Main Settings -> Applications-> Find App->Select->Clear defaults





# Первое приложение

- Установить Java SDK
- Установить Android Studio
- Установить Android SDK
- Создаем проект (API 21)



# Первое приложение

- Установить Eclipse
- Установить Android SDK (теперь **adt-bundle**)  
<http://developer.android.com/sdk/index.html>
- Установить ADT плагин к Eclipse  
<http://developer.android.com/tools/sdk/eclipse-adt.html>  
<https://dl-ssl.google.com/android/eclipse/>
- Все это за один раз ставит **adt-bundle**
- Запустить SDK Manager и установить нужные инструменты и платформы (API 18, Android 4.3)
- Создаем проект



# Первое приложение

**Select a wizard**  
Create an Android Application Project

Wizards:  
type filter text

- Class
- Interface
- Java Project
- Java Project from Existing Ant Buildfile
- Plug-in Project
- General
- Android
  - Android Activity
  - Android Application Project**
  - Android Icon Set
  - Android Object
  - Android Project from Existing Code
  - Android Sample Project
  - Android Test Project

**New Android App**  
Creates a new Android Application

Application Name: MyHelloAndroid

Project Name: MyHelloAndroid

Package Name: knure.examples.helloandroid

Build SDK: Google APIs (Google Inc.) (API 16) Choose...

Minimum Required SDK: API 8: Android 2.2 (Froyo)

Create custom launcher icon

Mark this project as a library

Create Project in Workspace

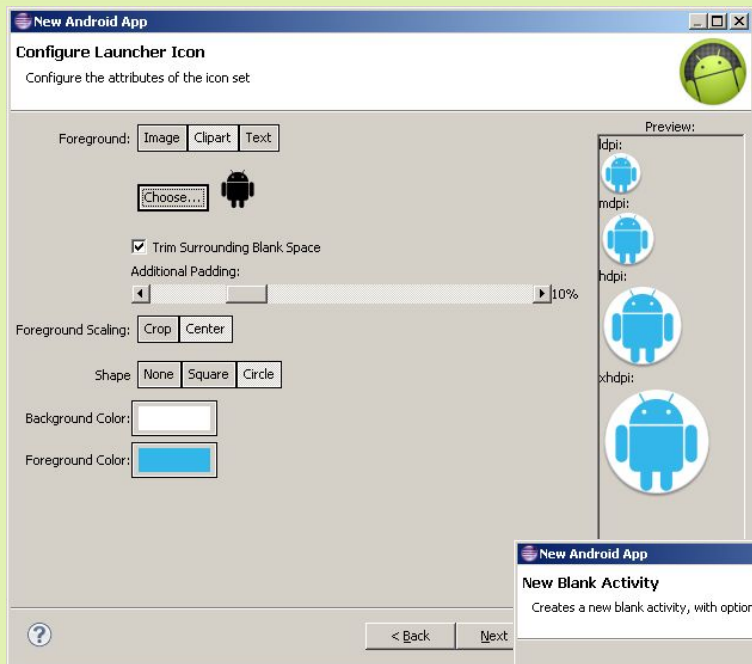
Location: D:\Work\Android\TestWS\MyHelloAndroid Browse...

Choose the lowest version of Android that your application will support. Lower API levels target more devices, but means fewer features are available. By targeting API 8 and later, you reach approximately 93% of the market.

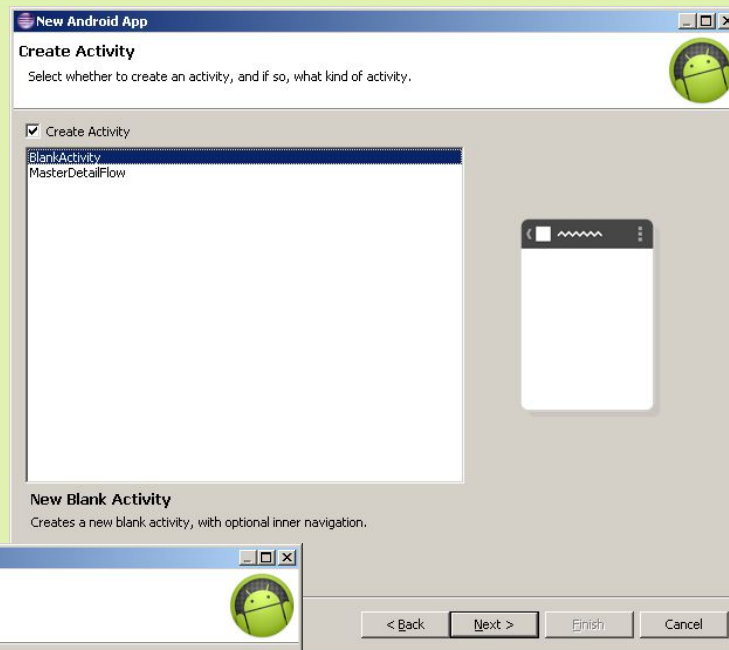
< Back Next > Finish

# Первое приложение

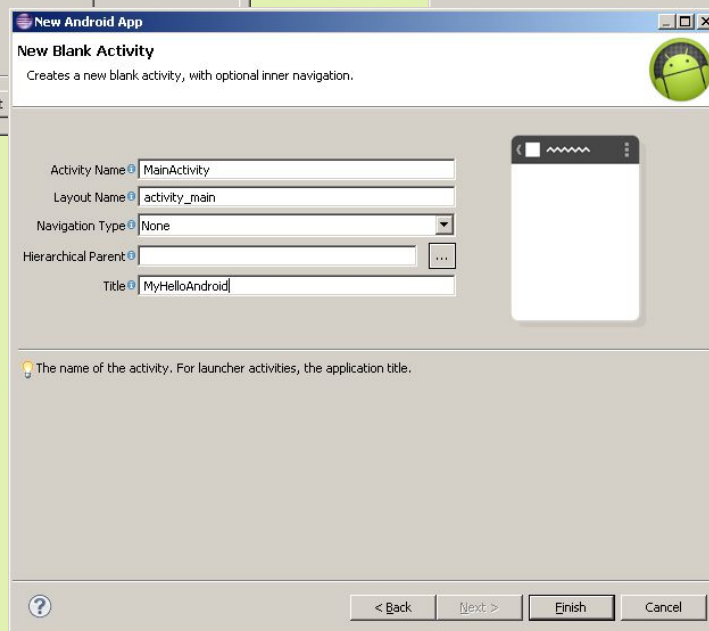
1



2



3



# Первое приложение (конфуз)

The screenshot displays an IDE interface with two main windows. On the left is the Package Explorer, showing the project structure for 'MyHelloAndroid'. The 'src' folder contains the package 'knure.examples.helloandroid', which includes 'MainActivity.java'. The 'gen' folder is highlighted, indicating generated files. On the right is the code editor, showing the 'MainActivity.java' file. The code is as follows:

```
package knure.examples.helloandroid;

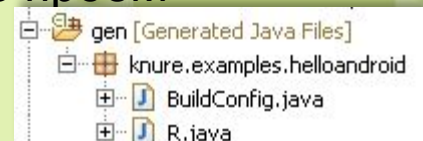
import android.os.Bundle;

public class MainActivity extends Activity {

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        getMenuInflater().inflate(R.menu.activity_main, menu);
        return true;
    }
}
```

Чтобы исправить сообщения об ошибке просто сохраните проект →  
файл R.java создается средой автоматически



# Первое приложение (принципы)

The screenshot displays the Android Studio IDE with several windows open:

- Palette:** Shows various form widgets like TextView, Button, Spinner, and RadioButtons.
- Preview:** Displays a mobile app interface with a header 'MyHelloAndroid' and a text view containing 'Hello world!'.
- Code Editor:** Shows the XML layout for the activity:

```
<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" >

    <TextView
        android:id="@+id/ma_tv_hello"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerHorizontal="true"
        android:layout_centerVertical="true"
        android:text="@string/hello_world"
        tools:context=".MainActivity" />

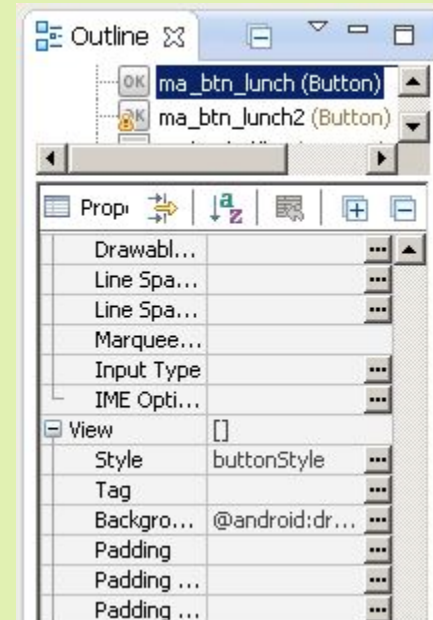
</RelativeLayout>
```
- Outline:** Shows the hierarchy of the layout: RelativeLayout containing TextView: @+id/ma\_tv\_hello.
- Android Resources (default):** Lists resources including strings.xml with the following elements:
  - app\_name (String)
  - hello\_world (String) - selected
  - menu\_settings (String)
  - title\_activity\_main (String)The 'Attributes for hello\_world (String)' panel shows:
  - Name: hello\_world
  - Value\*: Hello world!
- R.java:** Shows the generated resource IDs, including R.id.ma\_tv\_hello and R.string.hello\_world.
- Code Editor (bottom):** Shows the Java code for finding and setting the text view:

```
TextView tv = (TextView) findViewById(R.id.ma_tv_hello);
tv.setText(R.string.hello_world);
```

# Первое приложение(обработч.

соб.)  
Вариант I

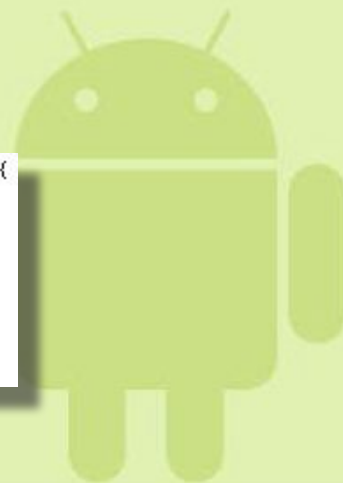
```
public class MyHelloActivity extends Activity {  
  
    @Override  
    public void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_hello);  
  
        ...  
  
        Button btn = (Button) findViewById(R.id.ma_btn_lunch);  
        btn.setOnClickListener(new OnClickListener() {  
  
            public void onClick(View v) {  
  
                ...  
  
            }  
  
        });  
  
        ...  
  
    }  
}
```



Вариант II

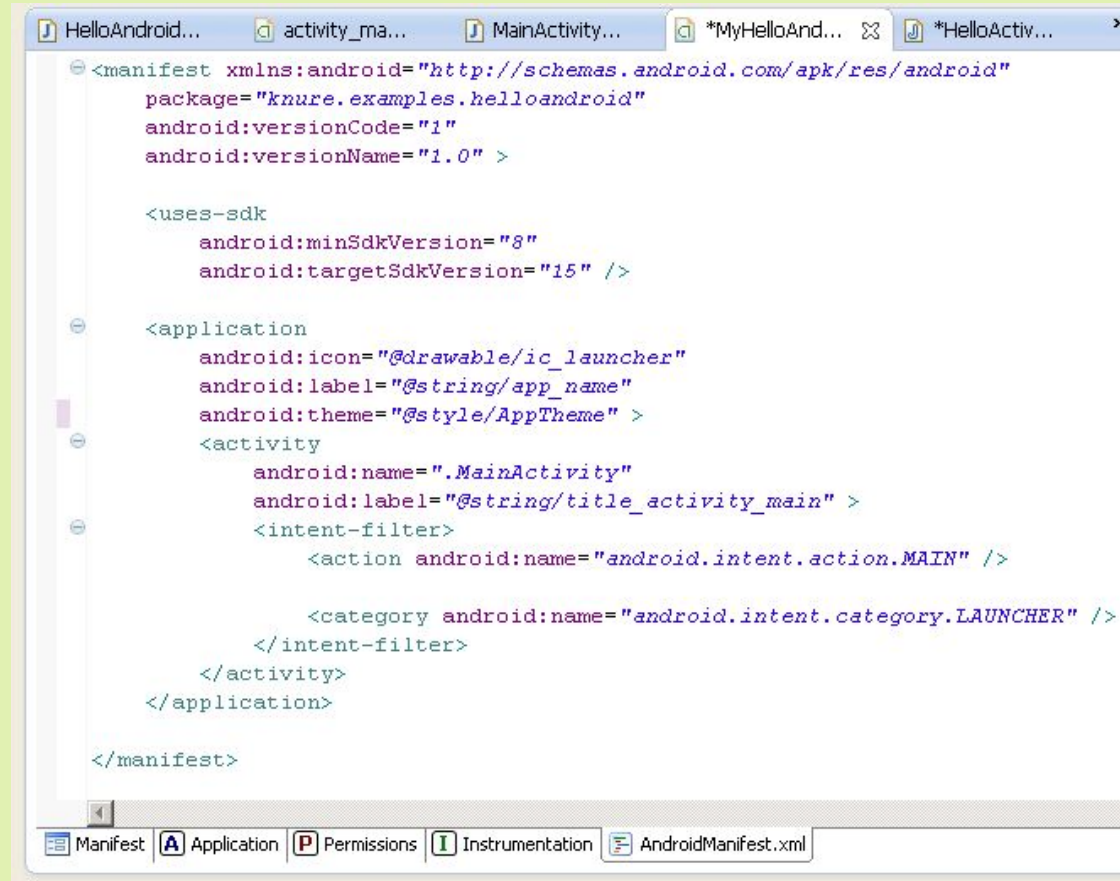
```
<Button  
    android:id="@+id/ma_btn_lunch"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_centerHorizontal="true"  
    android:layout_marginTop="16dp"  
    android:text="@string/main_lunch"  
    android:onClick="onClick"/>
```

```
public class MyHelloActivity extends Activity {  
    ...  
    public void onClick(View v) {  
  
    }  
    ...  
}
```



# Первое приложение. Манифест

- Имя, иконка, версия, требуемая версия Android, описание компонент, intent-фильтры для компонентов
- Какие права нужны приложению?
- Какими возможностями пользуется приложение (камера, bluetooth, ...)?



```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="knure.examples.helloandroid"
    android:versionCode="1"
    android:versionName="1.0" >

    <uses-sdk
        android:minSdkVersion="8"
        android:targetSdkVersion="15" />

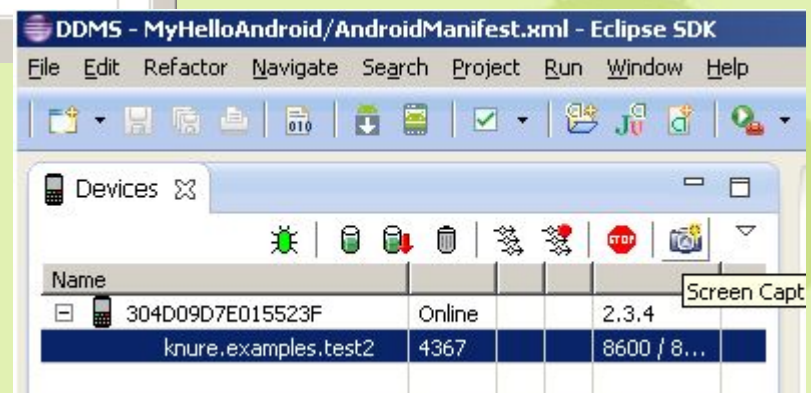
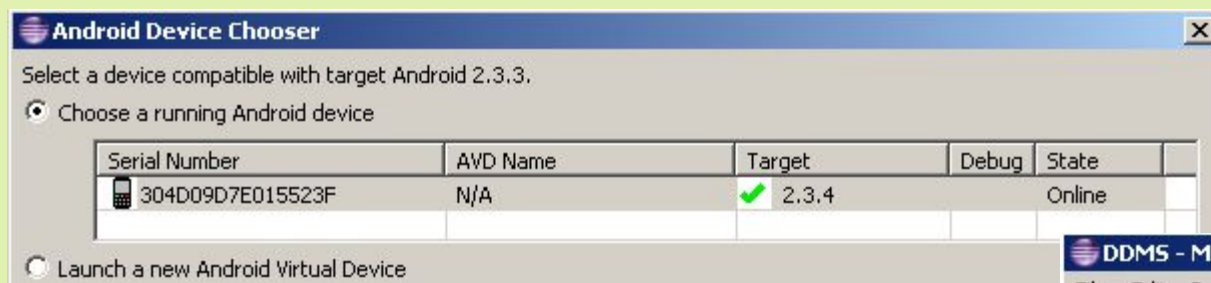
    <application
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/AppTheme" >
        <activity
            android:name=".MainActivity"
            android:label="@string/title_activity_main" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

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

</manifest>
```



# Запуск на эмуляторе или устройстве



## DDMS - Dalvik Debug Monitor Server

# Ресурсы приложения

Resource	Folder	Description
Текстовые значения	/res/values	Строковые ресурсы, цвета, размеры, статические массивы строк или целых чисел ( <b>strings.xml, colors.xml, dimens.xml, arrays.xml, integers.xml, bools.xml, ids.xml</b> )
Экраны и др. представления	/res/layout	Xml-описания макетов фрагментов экранов ( <b>*.xml</b> )
Стили и темы <b>Доступ из кода:</b>	/res/values	<b>styles.xml themes.xml</b> <b>attrs.xml</b> – custom-атрибуты тем
Меню <b>R.string.* , R.color.* , ..., R.layout.* , R.menu.* , R.style.* , ...</b>	/res/menu	xml-описания разных меню ( <b>*.xml</b> )

# Ресурсы приложения

Resource	Folder	Description
Бинарные данные	/res/raw	Любые файлы
XML-файлы	/res/xml	Произвольные xml-файлы
Анимация свойств	/res/anim	xml-описания временных анимаций любых свойств объектов (для property animation API)
Графика Доступ из кода: <b>R.raw.*</b> , <b>R.xml.*</b> , <b>R.drawable.*</b> , ... <code>res/values/strings.xml</code> :	/res/drawable	.png, .jpg, .gif, .9.png, .xml <b>Декларативное использование:</b>

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
  <string name="hello">Hello!</string>
</resources>
```

```
<TextView
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="@string/hello" />
```

## Получение в runtime из кода:

```
Resources res = getResources();
String string = res.getString(R.string.hello);
```

# Ресурсы приложения

Доступ из кода к содержимому файлов в `res/raw` :



```
InputStream inputStream = inputStream = getResources().openRawResource(R.raw.hello_world);
```

## Assets

Resource	Folder	Description
Аудио/видео	/assets	Хранит файлы произвольного типа, организованные в произвольную структуру папок, имена должны быть в нижнем регистре, размер файла до 1 Мб

# Папка Assets. Custom Шрифт.

## Пример

```
@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.main);

    final TextView textMyFont = (TextView)findViewById(R.id.editText1);
    textMyFont.setTypeface(Typeface.createFromAsset(
        getAssets(), "fonts/DigitalDream.ttf"));

    final TextView textMyFont2 = (TextView)findViewById(R.id.editText2);
    textMyFont2.setTypeface(Typeface.createFromAsset(getAssets(), "fonts/Catwalk.ttf"));
}
```

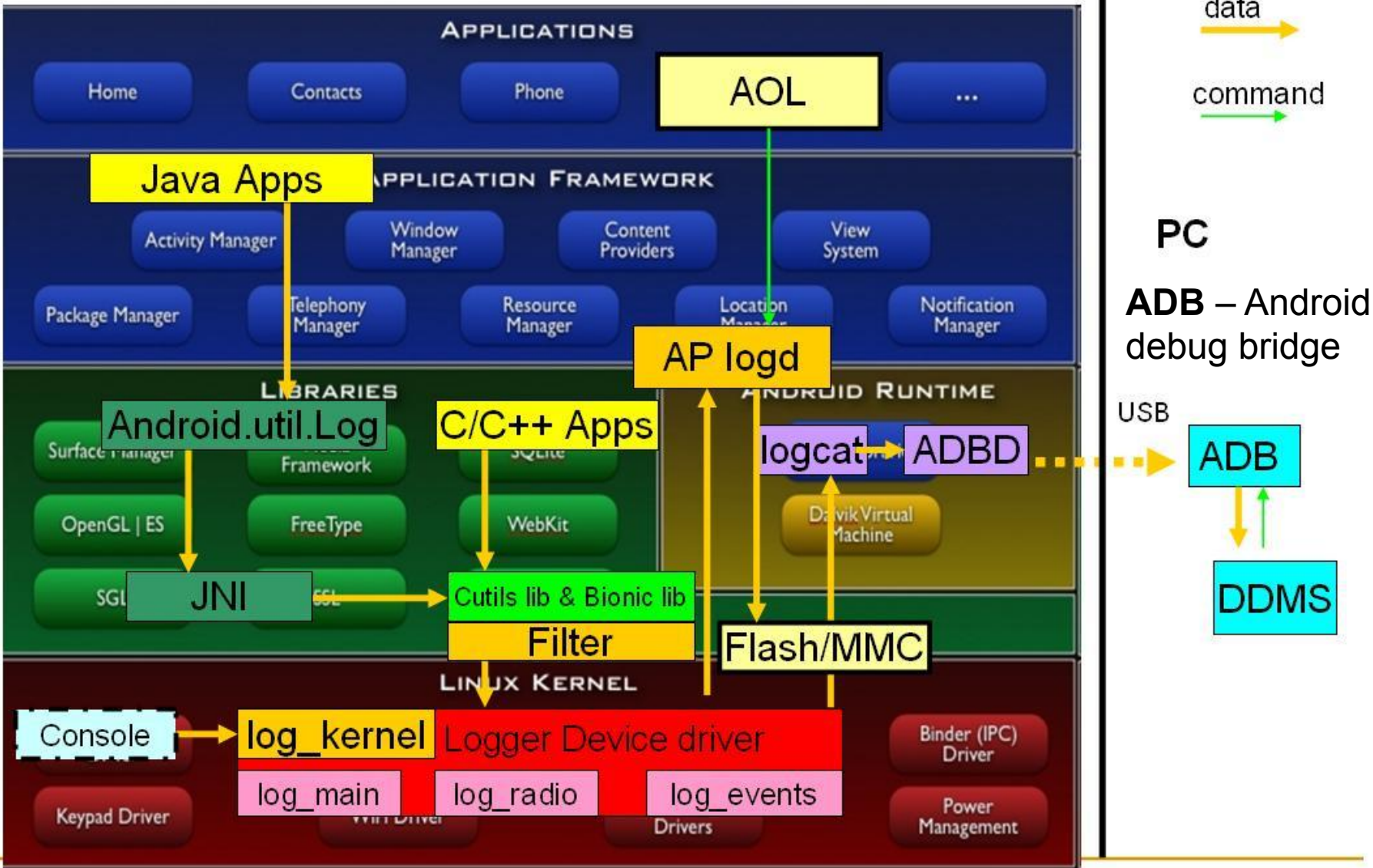
### Assets

#### Собственные шрифты

HAPPY NEW YEAR!



# ADB и Отладка приложения



# Отладка приложения

- Java-обертка `android.util.Log`

- `Log.e()` : ERROR

- `Log.w()` : WARN

- `Log.i()` : INFO

- `Log.d()` : DEBUG

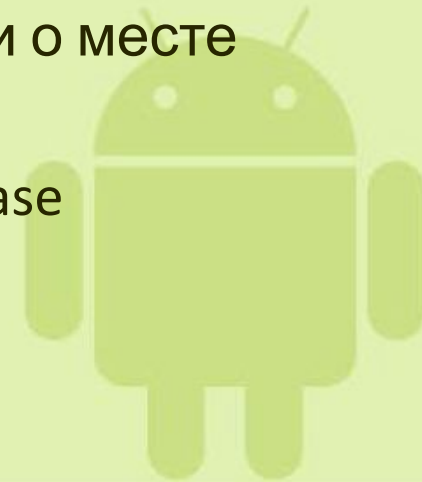
- `Log.v()` : VERBOSE

`public static int d (String tag, String msg)`

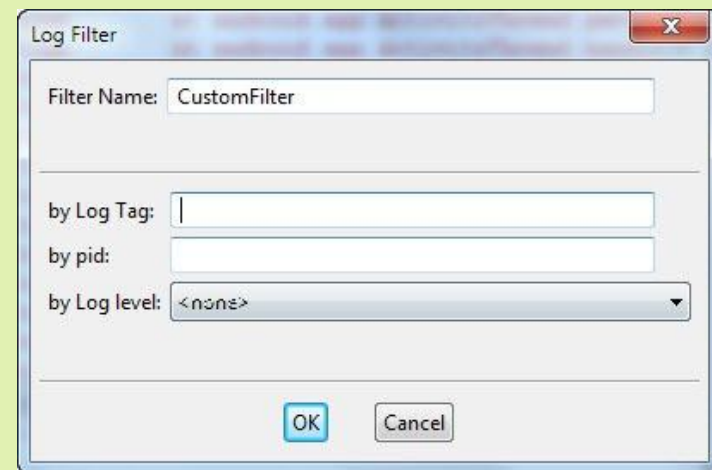
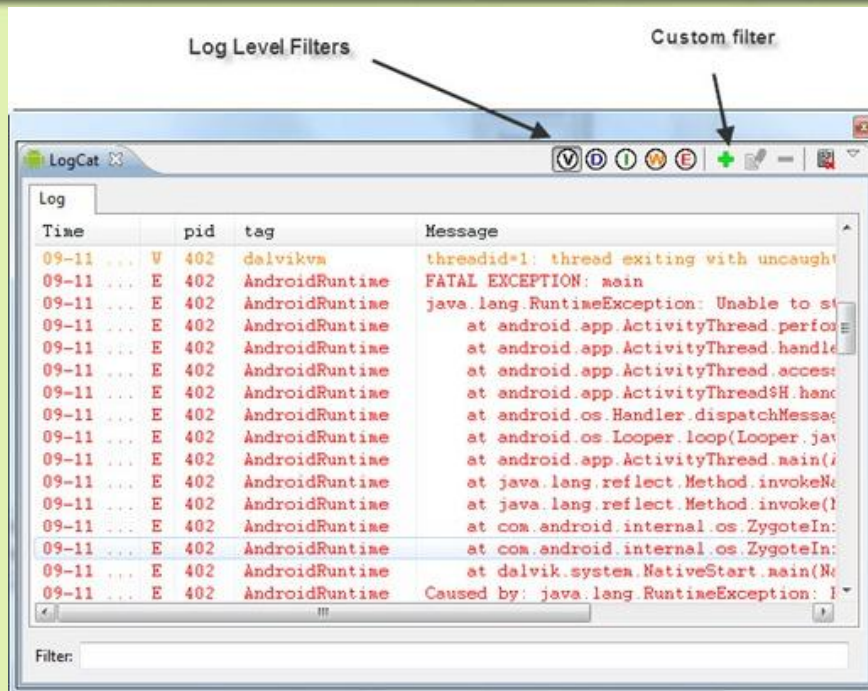
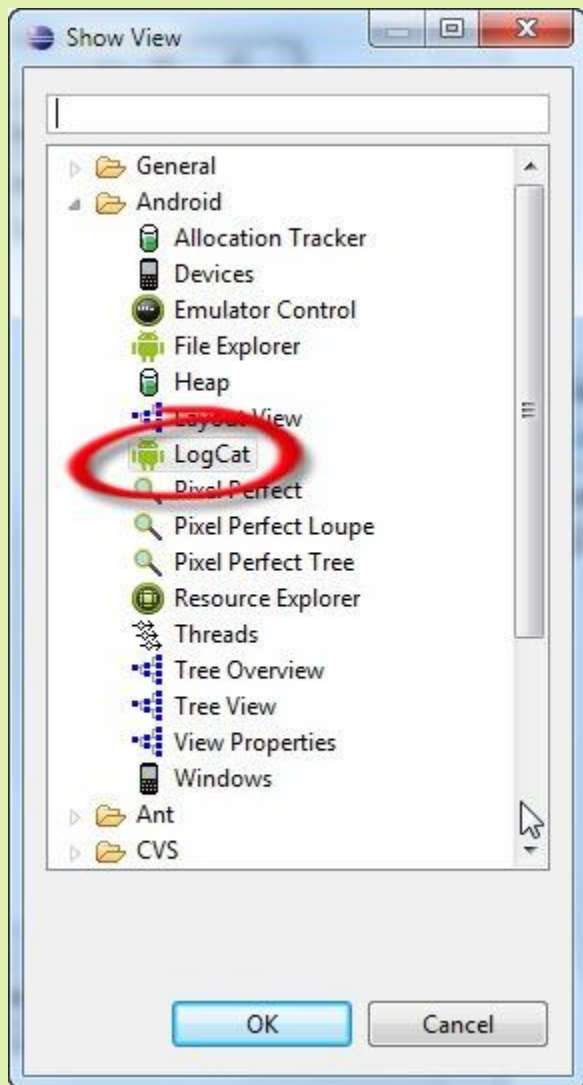
- `/dev/log/events`

- <http://habrahabr.ru/post/116376/> - пример удобной обертки для Log (автоматизирует получение информации о месте происхождения ошибки)

- Всё логирование должно быть отключено в Release



# Просмотр логов





# Консольное использование

## ADB

**Расположение:** .../sdk/platform-tools

**./adb devices** – список id подключенных устройств

**./adb -s <device id> shell** – запуск командного интерпретатора linux, позволяет использовать стандартные команды ls, cd, mkdir, rm

**5 must know** команд ADB: <http://execbit.ru/2013/08/25/adb-five/>

Например:

```
adb -s emulator-5554 shell
# sqlite3 /data/data/com.example.google.rss.rssexample/databases/rssitems.db
SQLite version 3.3.12
Enter ".help" for instructions
.... enter commands, then quit...
sqlite> .exit
```

# Консольное использование

## ADB

### Копирование файлов:

`./adb push <local-path-to-src-file> /sdcard` – на У.

`./adb pull /sdcard/ <local-path-to-src-file>` – с У.

### Работа с пакетами:

`./adb install <path-to-apk>` - установка на Устр.

`./adb uninstall <packagename>` - удаление с Устр.

### Перенаправление портов

`./adb forward tcp:6100 tcp:7100` – с 6100 на компьютере на 7100 на эмулятор



# Консольное использование

## ADB

### Управление Activity Manager

```
./adb shell am start -a -W <INTENT>
```

```
./adb shell am startservice <INTENT>
```

```
./adb shell am broadcast <INTENT>
```

**Например:** `./adb -s emulator-5554 shell am start`

`-W com.android.camera`

```
./adb shell am start -a
```

`android.intent.action.DIAL -d tel:555-5555`

-a запуск Activity без поиска фильтра LAUNCHER

-W – ожидание окончания запуска



## ADB

### Перезапуск adb (при потере эмулятора)

```
./adb kill-server
```

```
./adb start-server
```

### Подробнее:

[developer.android.com/tools/help/adb.html](http://developer.android.com/tools/help/adb.html)



# Управление эмулятором

## Полезные сочетания клавиш эмулятора:

**Ctrl+F11/F12** – смена ориентации

**F2** – MENU

**HOME** – HOME Screen

**ESC** – Back

**KEYPAD\_PLUS\MINUS** (Ctrl+F5/F6) – Уровень звука

**Ctrl-KEYPAD\_5** – Камера

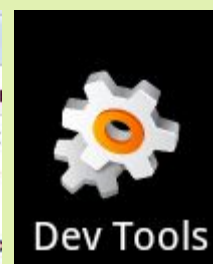
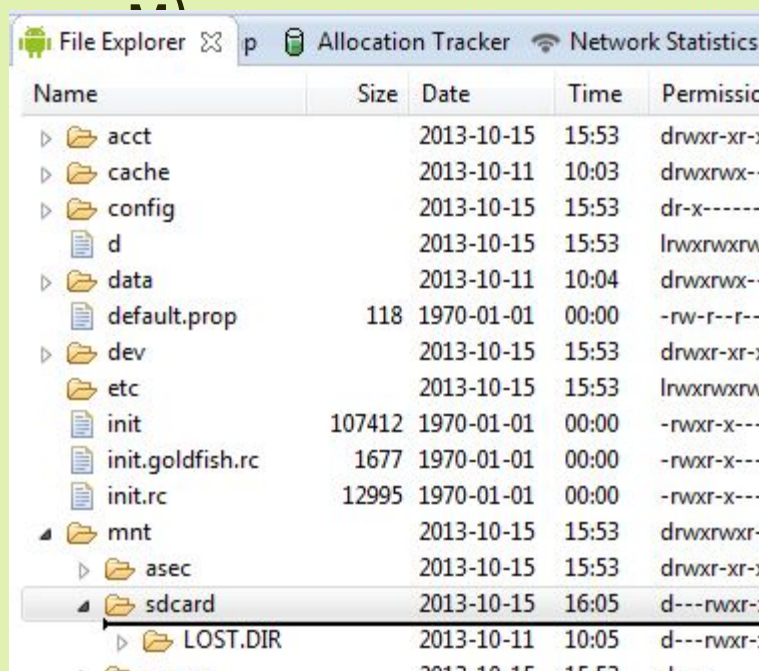
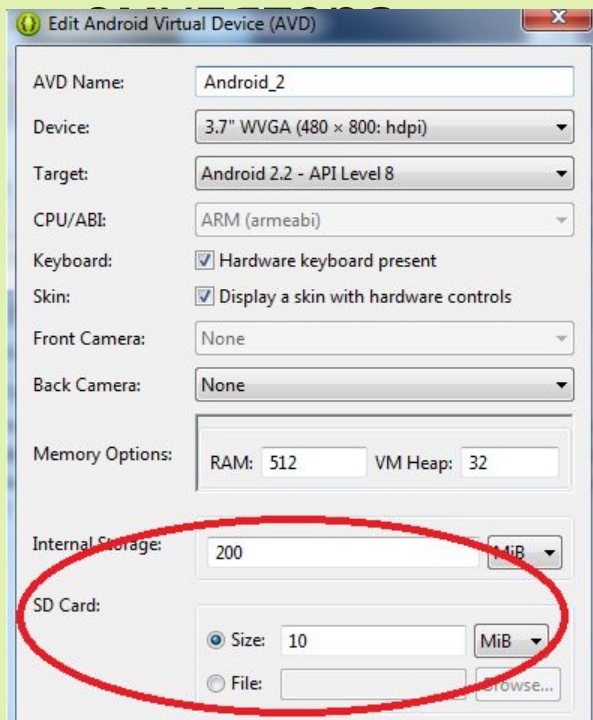
## Подробнее:

<http://developer.android.com/tools/help/emulator.html>

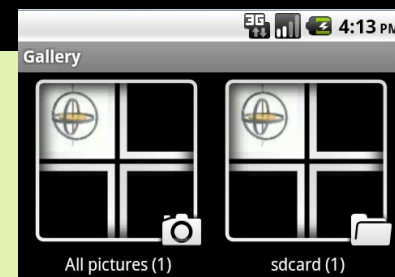
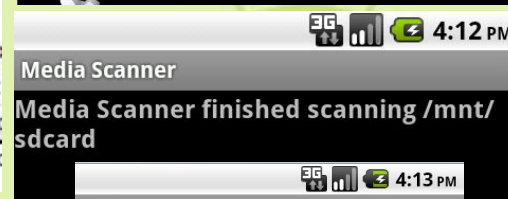
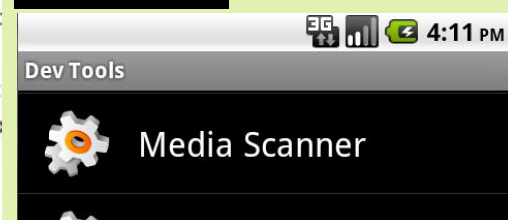


# Использование SD Card

1. Задание размера SD Card в настройках
2. Запись файлов на SD Card в DDMS (перетаскивание)
3. Индексация media-контента



Приложение на эмуляторе



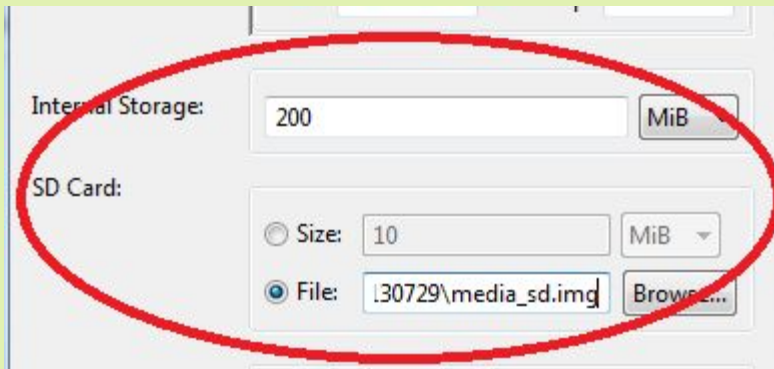
Теперь галерея и плеер «знают» про этот контент

# Переносимый файл SD Card

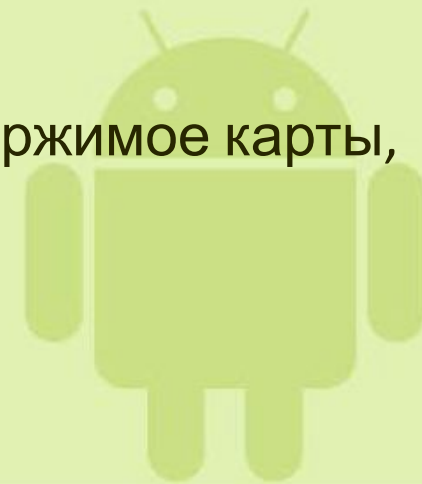
- Создание файла карты

```
\sdk\tools>mksdcard.exe -l media 256Mib media_sd.img
```

- Задание имени файла в настройках эмулятора



- Все изменения, вносимые приложениями в содержимое карты, отражаются в файле



# Управление эмулятором (AVD)

.../sdk/tools/android – утилита

- управления эмуляторами
- управление проектами
- управление SDK

Стандартно AVD располагаются в ~/.android/avd

**Может вызвать проблемы, если имя пользователя содержит не только латиницу**

Перенесение AVD на новое место:

```
android move avd -n <Имя ус-ва> -p <новый путь>
```

В новом пути не должна существовать последняя папка

Буква диска у старого и нового пути должны совпадать





# Особенности работы с сетью

- Loopback 127.0.0.1
- Внешний IP эмулятора 10.0.2.15
- Для эмулятора IP компьютера, на котором запущен эмулятор, - это 10.0.2.2

**Особенности для отладки сетевого приложения между двумя эмуляторами и**

**Звонок или SMS между эмуляторами и**

см. <http://developer.android.com/tools/devices/emulator.html>



# Интернет на эмуляторе через прокси

Зайти в "Settings" -> "Wireless & Networks" ->  
"Mobile Networks" -> "Access Point Names" ->  
"Telkita«

Указать Proxy и Port

ИЛИ

<http://developer.android.com/tools/devices/emulator.html#proxy>



# Особенности работы с сетью

- Loopback 127.0.0.1
- Внешний IP эмулятора 10.0.2.15
- Для эмулятора IP компьютера, на котором запущен эмулятор, - это 10.0.2.2

**Особенности для отладки сетевого приложения между двумя эмуляторами и**

**Звонок или SMS между эмуляторами и**

см. <http://developer.android.com/tools/devices/emulator.html>



# Симуляция сенсоров

<http://code.google.com/p/openintents/wiki/SensorSimulator>

yaw & pitch   roll & pitch   move

Sensor update: 23,50 ms

accelerometer: 0,00, 5,34, 8,22  
magnetic field: 15,68, -8,69, -45,85  
orientation: 333,00, -33,00, 0,00

Write emulator command port and click on set to create connection.  
Possible IP addresses:  
10.0.2.2  
192.168.0.207

Sensor Simulator

- Download the latest senso
- Start bin/sensorsimulator-x
- Install bin/SensorSimulator
- Launch [SensorSimulator](#) o
- Enter IP address and sock

yaw & pitch   roll & pitch   r

5554:Android\_4

Sensor Simulator Settings

Settings   Testing

Test the connection to the Sensor Simulator.  
Disable the sensors and disconnect after  
testing.

Connect   Disconnect

Sensor simulator data:  
accelerometer  
0.00, 5.34, 8.22  
magnetic field  
15.68, -8.69, -45.85

Basic Controls

Hardware Buttons

DPAD not enabled in AVD

Hardware Keyboard  
Use your physical keyboard to provide input



# Полезные ссылки от Google

- Краткие уроки от Google

<http://developer.android.com/training/index.html>

- Решение часто встречающихся задач

<http://developer.android.com/guide/faq/commontasks.html>

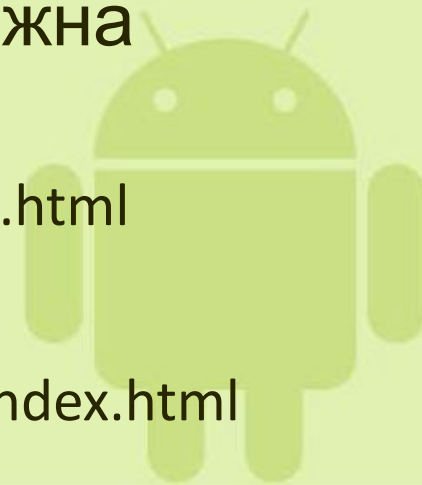
- Рекомендации от Google

– требования к интерфейсу (стиль иконок, поведение Activity, например, что должна делать кнопка «Back»)

<http://developer.android.com/design/index.html>

и пр.

<http://developer.android.com/guide/practices/index.html>



# Сторонние полезные источники

- Удачная подборка уроков

<http://developer.alexanderklimov.ru/android>

<http://startandroid.ru>

<http://www.vogella.com/tutorials.html>

- Книги

**Сатья Коматинени «Pro Android 4», Apress  
и напоследок**

<http://source.android.com/source/code-style.html>

