



Введение в разработку мобильных приложений

Введение в разработку приложений для смартфонов на ОС Android

Содержание

- ▶ Устройство платформы Android
- ▶ Обзор сред программирования
- ▶ Android для разработчика
- ▶ Эмуляторы
- ▶ Возможности отладки на реальных устройствах
- ▶ Примеры приложений

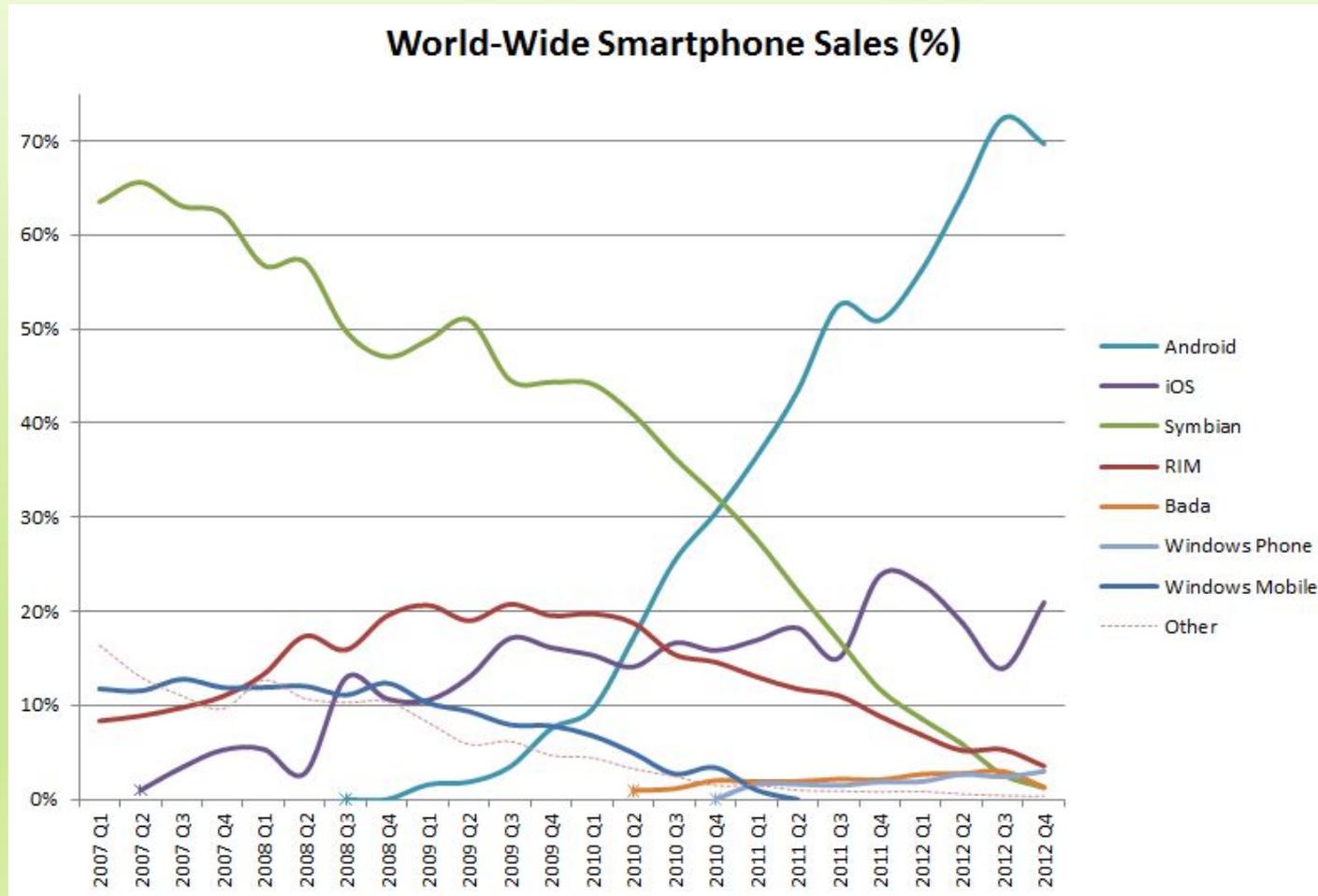
Устройство платформы Android

Android

- ▶ Android – операционная система для мобильных устройств (смартфонов, КПК, планшетных компьютеров), основанная на ядре Linux
- ▶ Позволяет создавать Java-приложения, управляющие устройством через разработанные Google библиотеки
- ▶ Есть возможность писать приложения на Си и некоторых других языках программирования
- ▶ Официальный сайт: www.android.com



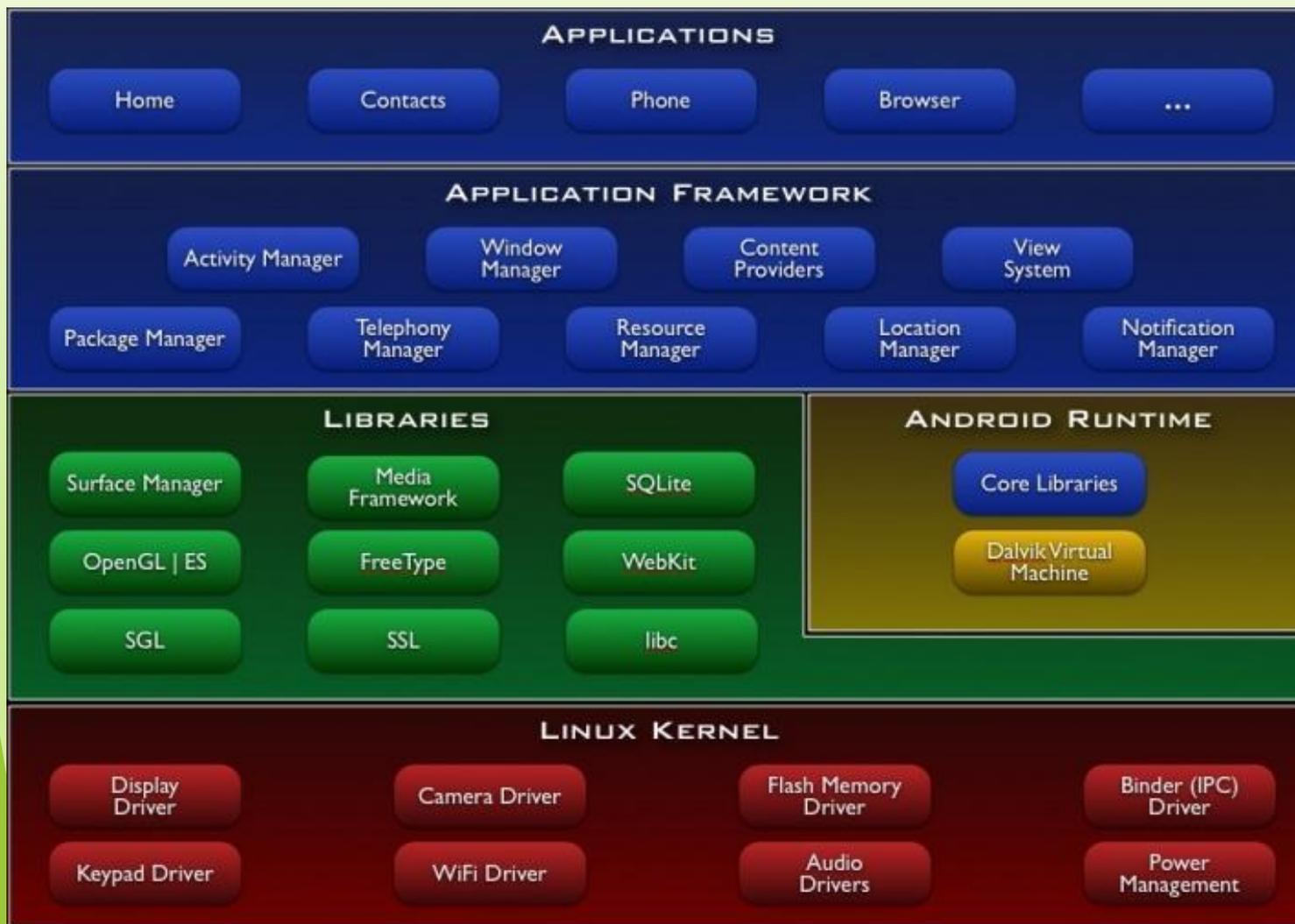
Доля Android на рынке



Достижения и проблемы

- ▶ Фрагментация рынка (более 4000 моделей)
- ▶ 700 млн. устройств продано, 850 тыс. устройств активируются каждый день
- ▶ Проблемы с аппаратной совместимостью
- ▶ Безопасность
- ▶ Обратная стороны популярности и открытости платформы
- ▶ Недостаток централизованной поддержки

Архитектура Android



Базовый уровень

Уровень абстракции между аппаратным обеспечением и программным стеком:

- ▶ В основе лежит ядро ОС Linux (несколько урезанное)
- ▶ Обеспечивает функционирование системы;
- ▶ Отвечает за безопасность;
- ▶ Управляет памятью, энергосистемой и процессами;
- ▶ Предоставляет сетевой стек и модель драйверов.

LINUX KERNEL

Display
Driver

Camera Driver

Flash Memory
Driver

Binder (IPC)
Driver

Keypad Driver

WiFi Driver

Audio
Drivers

Power
Management

Набор библиотек

Обеспечивает важнейший базовый функционал для приложений:

- ▶ Алгоритмы для вышележащих уровней
- ▶ Поддержка файловых форматов
- ▶ Кодирование и декодирование информации
- ▶ Отрисовка графики и т. д.



Библиотеки

Реализованы на C/C++ и скомпилированы под конкретное аппаратное обеспечение устройства, вместе с которым они поставляются:

- ▶ **Surface Manager**
- ▶ **Media Framework**
- ▶ **SQLite**
- ▶ **3D библиотеки**
- ▶ **LibWebCore**
- ▶ **SGL (Skia Graphics Engine)**
- ▶ **SSL**



Среда выполнения Android Runtime

- ▶ Библиотеки ядра, обеспечивающие большую часть низкоуровневой функциональности, доступной библиотекам ядра языка Java
- ▶ Виртуальная машина Dalvik, позволяющая запускать приложения



Application Framework - уровень каркаса приложений

- ▶ Обеспечивает разработчикам доступ к API, предоставляемым компонентами системы уровня библиотек
- ▶ Любому приложению предоставляются уже реализованные возможности других приложений, к которым разрешено получать доступ

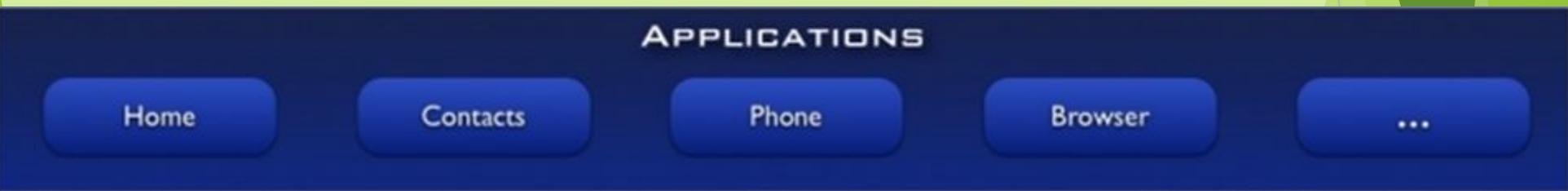


Application Framework

- ▶ Богатый и расширяемый набор представлений (**Views**)
- ▶ Контент-провайдеры (**Content Providers**)
- ▶ Менеджер ресурсов (**Resource Manager**)
- ▶ Менеджер оповещений (**Notification Manager**)
- ▶ Менеджер действий (**Activity Manager**)
- ▶ Менеджер местоположения (**Location Manager**)

Applications - уровень приложений

- ▶ Набор базовых приложений, который предустановлен на ОС Android. Например, браузер, почтовый клиент, программа для отправки SMS, карты, календарь, менеджер контактов и др.



Обзор сред программирования

Среды разработки

- ▶ Android IDE
- ▶ Android Studio (IntelliJ IDEA)
- ▶ Intel XDK

Android IDE



- ▶ Реализован полный цикл разработки редактирование-компиляция-выполнение
- ▶ Автодополнение кода, проверка ошибок в реальном времени, рефакторинг, "умная" навигация по коду и запуск вашего приложения в одно касание



Android IDE

Возможна разработка настоящих приложений для Андроида прямо на устройствах с ОС Андроид:

- ▶ Андроид-планшет с клавиатурой может стать полноценным местом разработки
- ▶ Можно просматривать и редактировать код прямо на смартфоне



Android IDE

- ▶ Поддерживает разработку с использованием Java/XML и Android SDK так же хорошо, как с C/C++ и Android NDK
- ▶ Полностью совместима с проектами Eclipse
- ▶ Интеграция с Dropbox
- ▶ Поддерживает GIT для профессиональной разработки приложений

Android IDE

The screenshot displays the Android Studio IDE with the following components:

- Package Explorer (Left):** Shows the project structure for 'HelloWorld', including 'src', 'gen', 'Android 4.4', 'Android Private Libraries', 'assets', 'bin', 'libs', 'res' (with subfolders for various screen densities and 'layout'), 'menu', 'values', and 'AndroidManifest.xml'.
- Code Editor (Center):** Displays the XML layout for 'activity_main.xml'. The code defines a `<RelativeLayout>` containing a `<TextView>` with the text '@string/hello_world'.
- Outline (Right):** Shows the 'RelativeLayout' structure.
- LogCat (Bottom):** Shows the execution log for the emulator 'emulator-5554', including messages for 'HOME is up', 'Uploading HelloWorld.apk', 'Installing HelloWorld.apk', 'Success!', and 'Starting activity com.mypackage.helloworld.MainActivity'.

```
<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"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context=".MainActivity" >

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/hello_world" />

</RelativeLayout>
```

81M of 183M

Android IDE

The screenshot displays the Android IDE (ADT) interface. The main window shows the source code for `MainActivity.java` in the package `com.mypackage.helloworld`. The code includes imports for `android.os.Bundle`, `android.app.Activity`, and `android.view.Menu`. The `MainActivity` class extends `Activity` and implements two methods: `onCreate` and `onCreateOptionsMenu`.

```
package com.mypackage.helloworld;

import android.os.Bundle;
import android.app.Activity;
import android.view.Menu;

public class MainActivity extends Activity {

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

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if it is present.
        getMenuInflater().inflate(R.menu.main, menu);
        return true;
    }
}
```

The Package Explorer on the left shows the project structure, including the `src` directory with `MainActivity.java` and `MainActivity` class. The Outline view on the right shows the class hierarchy and methods. The Console window at the bottom displays the following log output:

```
Android
[2013-11-09 21:39:47 - HelloWorld] Launching a new emulator with Virtual Device 'MyDevice'
[2013-11-09 21:40:08 - Emulator] emulator: emulator window was out of view and was recentered
[2013-11-09 21:40:08 - Emulator]
[2013-11-09 21:40:08 - HelloWorld] New emulator found: emulator-5554
[2013-11-09 21:40:08 - HelloWorld] Waiting for HOME ('android.process.acore') to be launched...
```

Android IDE

The screenshot displays the Android Studio IDE with the following components:

- Package Explorer (Left):** Shows the project structure for 'HelloWorld', including 'src', 'gen', 'Android 4.4', 'Android Private Libraries', 'assets', 'bin', 'libs', 'res' (with subfolders for different screen densities and 'layout'), 'menu', 'values', and 'AndroidManifest.xml'.
- Palette (Top Left):** Lists various Android widgets such as Form Widgets, Text Fields, Layouts, Composite, Images & Media, Time & Date, Transitions, Advanced, Other, and Custom & Library Views.
- Graphical Layout (Center):** Shows a preview of the application on a Nexus One emulator. The screen displays 'HelloWorld' and 'Hello world!'.
- Outline (Top Right):** Shows the current layout structure: 'RelativeLayout' containing a 'TextView - "Hello world!"'.
- Properties (Bottom Right):** Shows '<No properties>'.
- Console (Bottom):** Displays the following log output:

```
Android
[2013-11-08 20:22:40 - HelloWorld] HOME is up on device 'emulator-5554'
[2013-11-08 20:22:40 - HelloWorld] Uploading HelloWorld.apk onto device 'emulator-5554'
[2013-11-08 20:22:41 - HelloWorld] Installing HelloWorld.apk...
[2013-11-08 20:24:43 - HelloWorld] Success!
[2013-11-08 20:24:43 - HelloWorld] Starting activity com.mypackage.helloworld.MainActivity on device emulator-5554
```

Android Studio (IntelliJ IDEA)

- ▶ Позволяет редактировать приложение в реальном времени, отображая его поведение одновременно на устройствах с различными диагоналями экранов
- ▶ Доступно мгновенное переключение на различные типы верстки и размеры экранов
- ▶ Встроены инструменты оттачивания качества приложений и выстраивания путей их монетизации

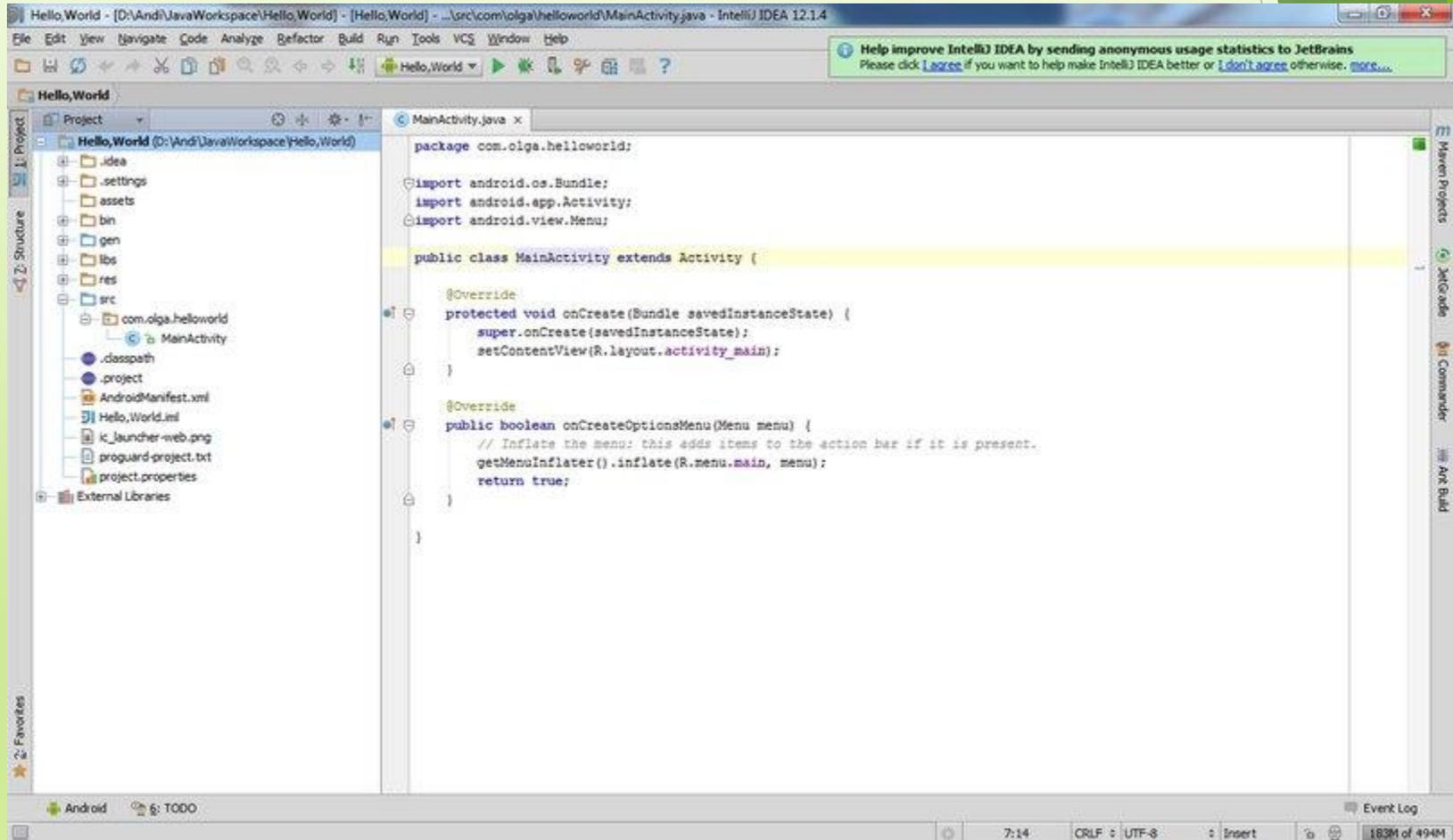


Android Studio (IntelliJ IDEA)

- ▶ Раздел с подсказками и советами по оптимизации с тематическими разделами
- ▶ Средство взаимодействия с бета-тестерами
- ▶ Позволяет ускорить процесс разработки софтверных изделий, сделав его более продуктивным



Android Studio (IntelliJ IDEA)



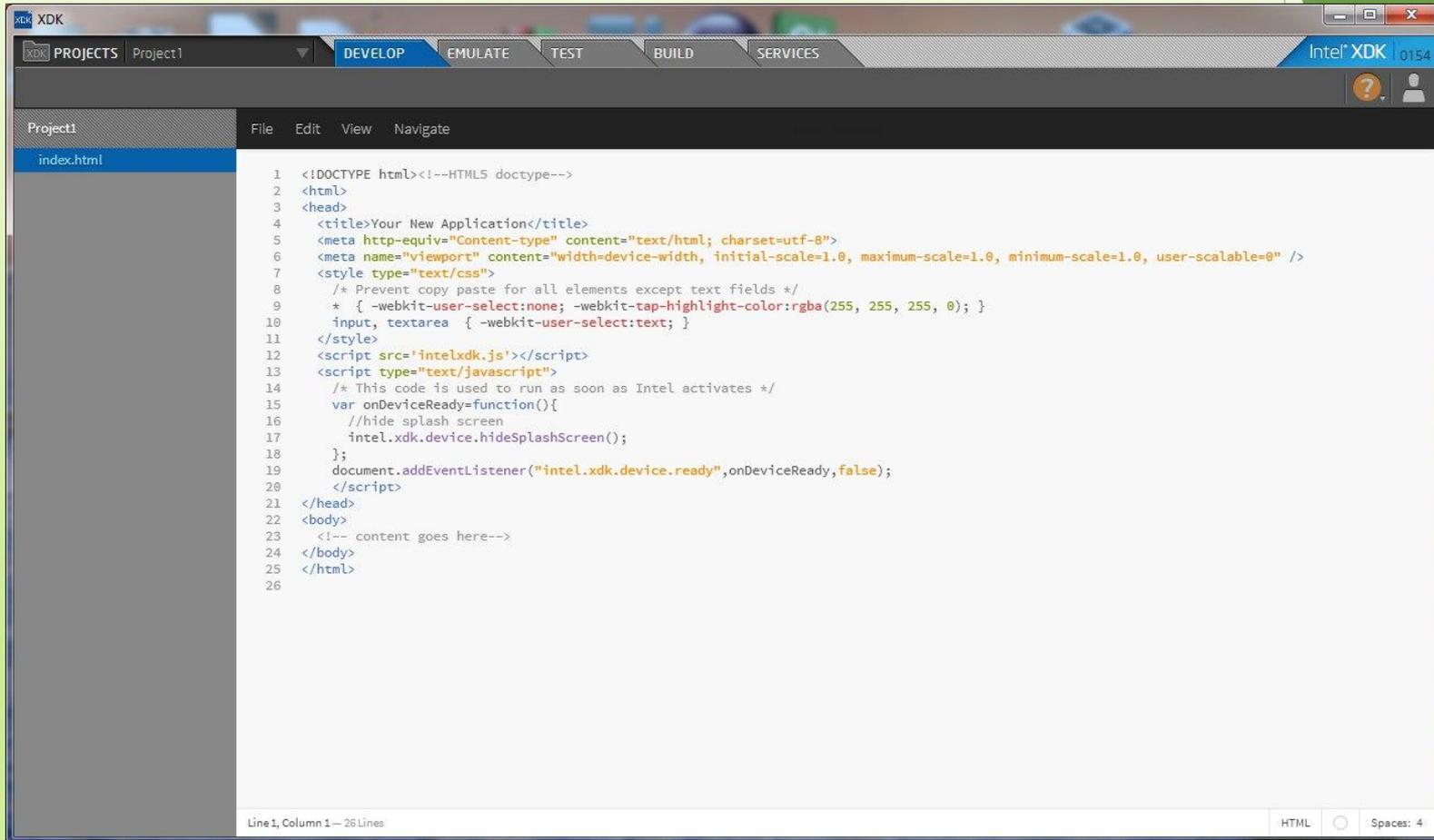


Intel XDK

- ▶ Позволяет легко разрабатывать кроссплатформенные приложения
- ▶ Включает в себя инструменты для создания, отладки и сборки ПО, а также эмулятор устройств
- ▶ Поддерживает разработку для Android, Apple iOS, Microsoft Windows 8, Tizen
- ▶ Языки разработки HTML5 и JavaScript



Intel XDK



The screenshot shows the Intel XDK IDE interface. The top menu bar includes PROJECTS, DEVELOP, EMULATE, TEST, BUILD, and SERVICES. The main editor displays the following code for index.html:

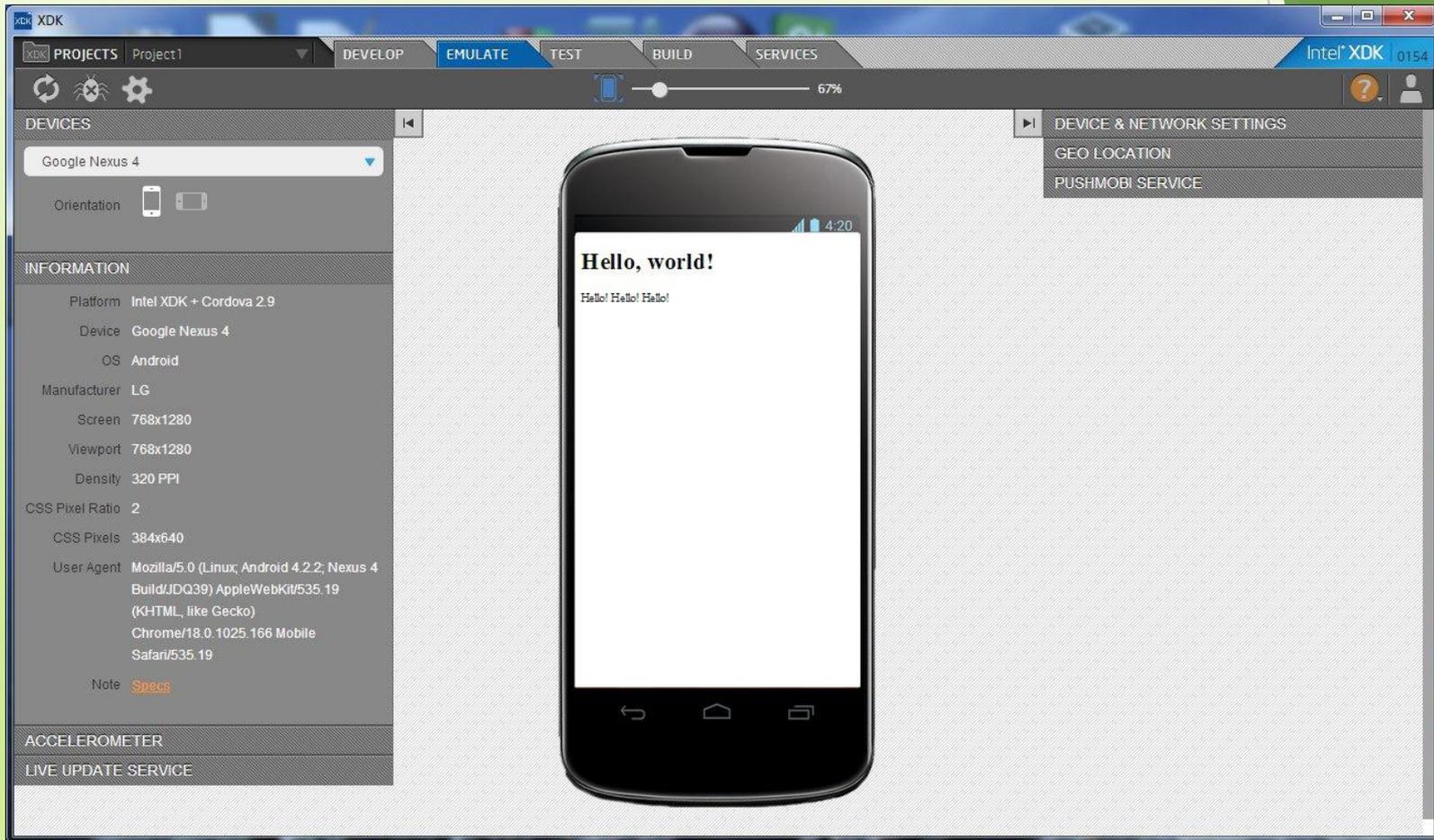
```

1 <!DOCTYPE html><!--HTML5 doctype-->
2 <html>
3 <head>
4 <title>Your New Application</title>
5 <meta http-equiv="Content-type" content="text/html; charset=utf-8">
6 <meta name="viewport" content="width=device-width, initial-scale=1.0, maximum-scale=1.0, minimum-scale=1.0, user-scalable=0" />
7 <style type="text/css">
8     /* Prevent copy paste for all elements except text fields */
9     * { -webkit-user-select:none; -webkit-tap-highlight-color:rgba(255, 255, 255, 0); }
10    input, textarea { -webkit-user-select:text; }
11 </style>
12 <script src='intelxdk.js'></script>
13 <script type="text/javascript">
14     /* This code is used to run as soon as Intel activates */
15     var onDeviceReady=function(){
16         //hide splash screen
17         intel.xdk.device.hideSplashScreen();
18     };
19     document.addEventListener("intel.xdk.device.ready",onDeviceReady,false);
20 </script>
21 </head>
22 <body>
23 <!-- content goes here-->
24 </body>
25 </html>
26

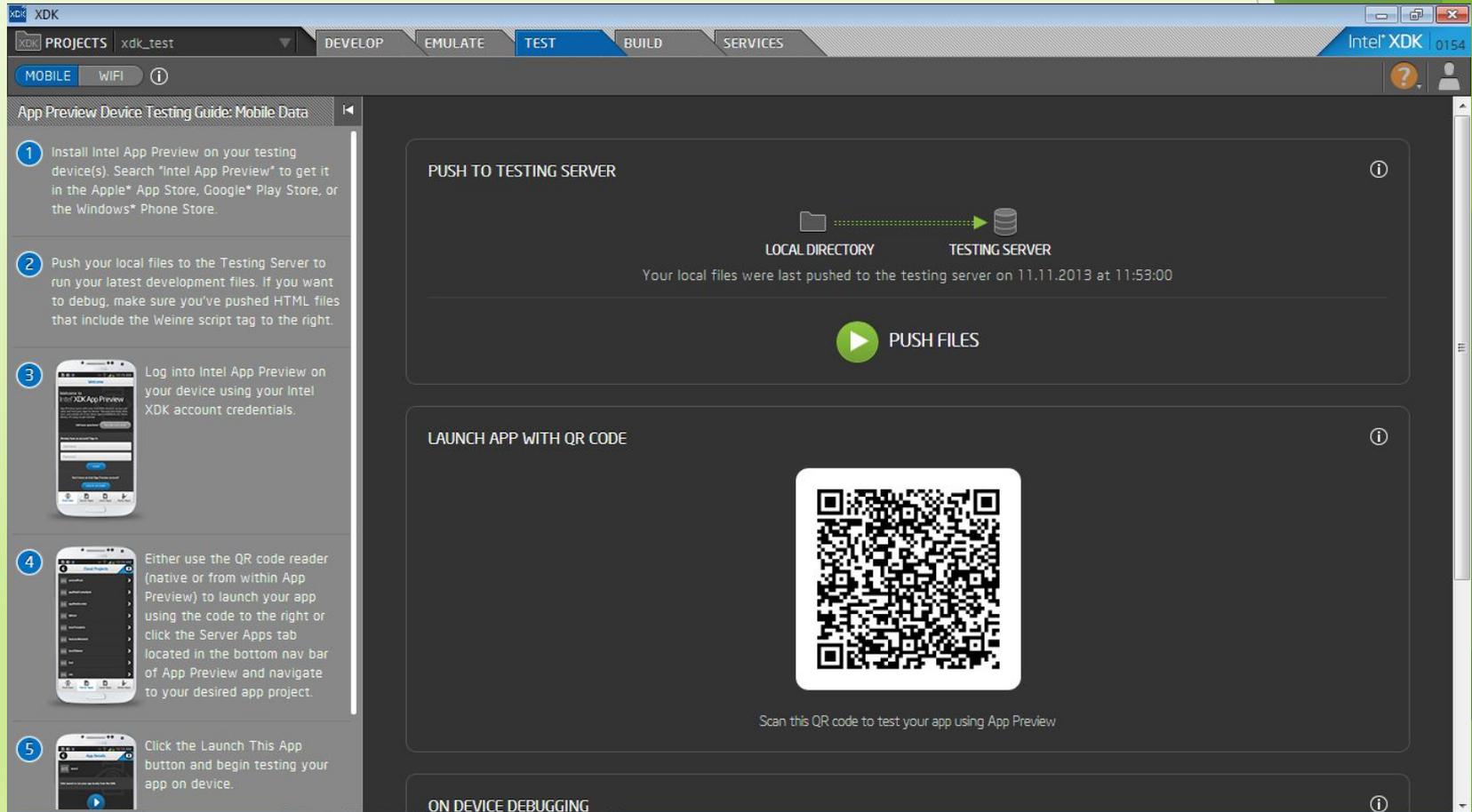
```

The status bar at the bottom indicates "Line 1, Column 1 - 26 Lines" and "HTML" with a "Spaces: 4" setting.

Intel XDK



Intel XDK



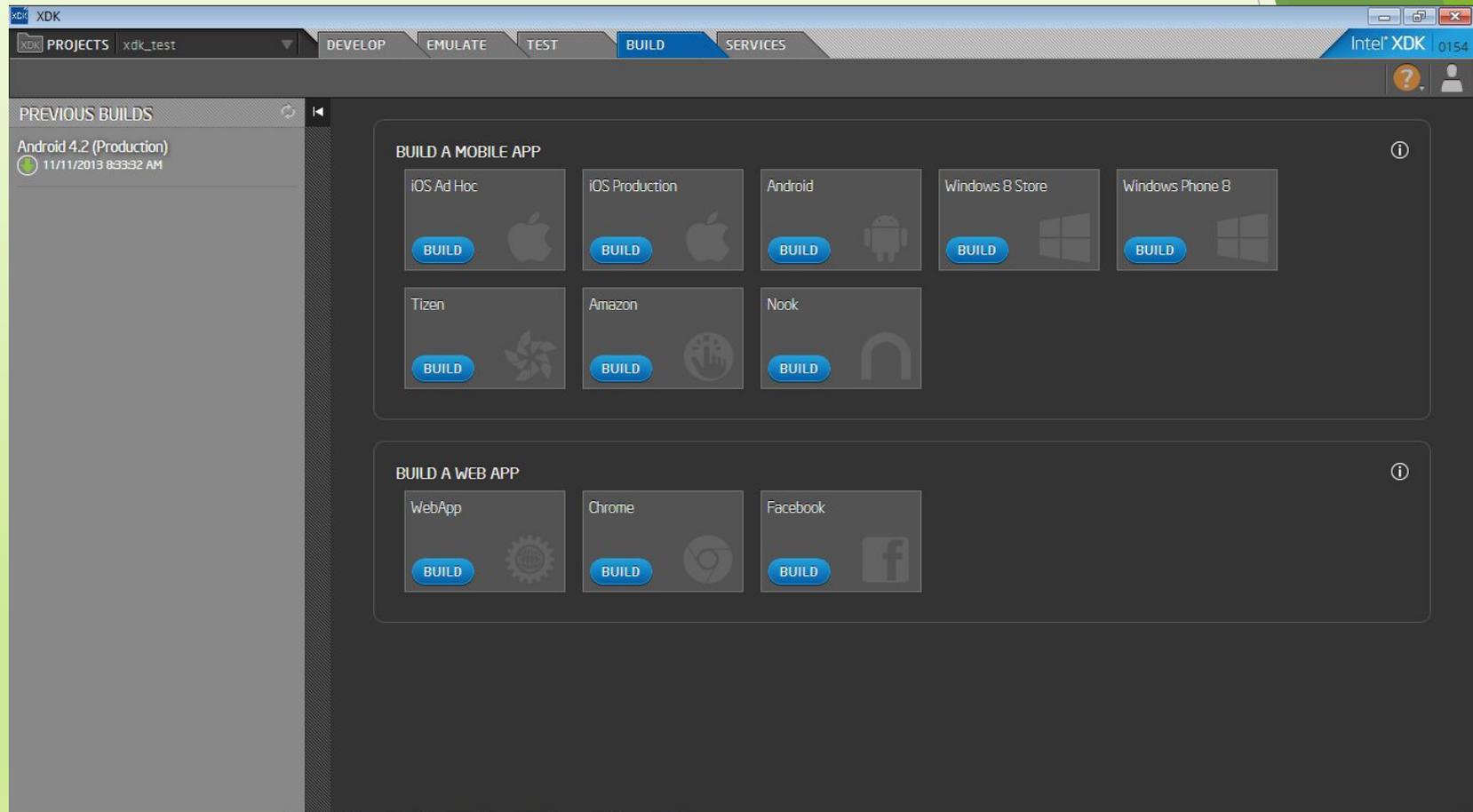
The screenshot shows the Intel XDK interface with the 'TEST' tab selected. The interface is divided into a left sidebar with a 'MOBILE' tab and a main content area. The sidebar contains a 'App Preview Device Testing Guide: Mobile Data' section with five numbered steps:

- 1 Install Intel App Preview on your testing device(s). Search "Intel App Preview" to get it in the Apple® App Store, Google® Play Store, or the Windows® Phone Store.
- 2 Push your local files to the Testing Server to run your latest development files. If you want to debug, make sure you've pushed HTML files that include the Weinre script tag to the right.
- 3 Log into Intel App Preview on your device using your Intel XDK account credentials.
- 4 Either use the QR code reader (native or from within App Preview) to launch your app using the code to the right or click the Server Apps tab located in the bottom nav bar of App Preview and navigate to your desired app project.
- 5 Click the Launch This App button and begin testing your app on device.

The main content area displays the following sections:

- PUSH TO TESTING SERVER**: A diagram shows a folder icon labeled 'LOCAL DIRECTORY' connected by a dotted arrow to a server icon labeled 'TESTING SERVER'. Below the diagram, it states: "Your local files were last pushed to the testing server on 11.11.2013 at 11:59:00". A large green play button icon is followed by the text "PUSH FILES".
- LAUNCH APP WITH QR CODE**: A large QR code is displayed in the center. Below it, the text reads: "Scan this QR code to test your app using App Preview".
- ON DEVICE DEBUGGING**: This section is partially visible at the bottom of the main content area.

Intel XDK



Инструменты разработки

- ▶ Google Android SDK (ADT Bundle)
- ▶ Android NDK
- ▶ Marmalade SDK
- ▶ Cygwin (for Microsoft Window OS)

Marmalade SDK



- ▶ Кроссплатформенное SDK от Ideaworks3D Limited
- ▶ Представляет собой набор библиотек, образцов, инструментов и документаций необходимых для разработки, тестирования и развертывания приложений для мобильных устройств
- ▶ Используется для разработки игр
- ▶ Проприетарная 😞

Инструменты Intel

- ▶ Intel* Hardware Accelerated Execution Manager (Intel* HAXM)
- ▶ Intel* Graphics Performance Analyzers (Intel* GPA) System Analyzer
- ▶ Intel* Integrated Performance Primitives (Intel* IPP) Preview
- ▶ Intel* Threading Building Blocks (Intel* TBB)
- ▶ Intel* Software Manager



Beacon Mountain

- ▶ Разработка Intel для Android
- ▶ Предоставляет инструменты для проектирования, разработки, отладки и оптимизации приложений под Android
- ▶ Следит за обновлениями и добавляет их в среду разработки по мере появления
- ▶ Поддерживает разработку для целевых платформ на основе процессоров Intel Atom и ARM



Beacon Mountain

The screenshot shows the Intel website's product page for Beacon Mountain Beta for Android*. The page features a navigation bar with links for Intel Software, App Development, Intel Platform Development, Intel Technologies, and Business Resources. A search bar and 'Login or Register' link are also present. The main content area includes a product image, a description, a list of features, and a download button. A sidebar on the right contains support and related content links.

Beacon Mountain Beta for Android*

Codename for Intel's development framework for native Android* apps on Intel® architecture and ARM* processor-based devices

- Supports Jelly Bean and up.
- Runs on Apple OS X* and Microsoft Windows* 7 and 8, 64-bit host environments.
- Provides Eclipse* plugins, and supports Android SDK, NDK, and more.

Download

OVERVIEW | DOCUMENTATION & SUPPORT | BEACON MOUNTAIN DETAILS | UPGRADE NOTES

NEW Now supporting Apple OS X* and Microsoft Windows* 7 & 8 host systems

Speed Development of Android Apps for Devices Using ARM* and Intel® Atom™ Processors

Beacon Mountain provides productivity-oriented design, coding, and debugging tools for native apps targeting Android-based ARM and Intel Atom processor-based devices, including smartphones and tablets. The tools are compatible with Eclipse and support popular Android SDKs including the Android NDK.

Key features:

- Simple, fast installation of popular Intel® developer and third-party tools for creating Android applications
- Compatible with and augments existing Android SDK and NDK toolkits
- Supports Apple OS X*, Microsoft Windows* 7 and 8 host systems

SUPPORT

- Support Forum ›
- Frequently Asked Questions ›
- Beacon Mountain Product Brief ›

RELATED CONTENT

- Intel® HAXM ›
- Intel® IPP for Android* ›
- Intel® GPA ›
- Intel® TBB ›
- Intel® USB Driver for Android* ›
- Intel® XDK ›
- Intel Android* Zone ›

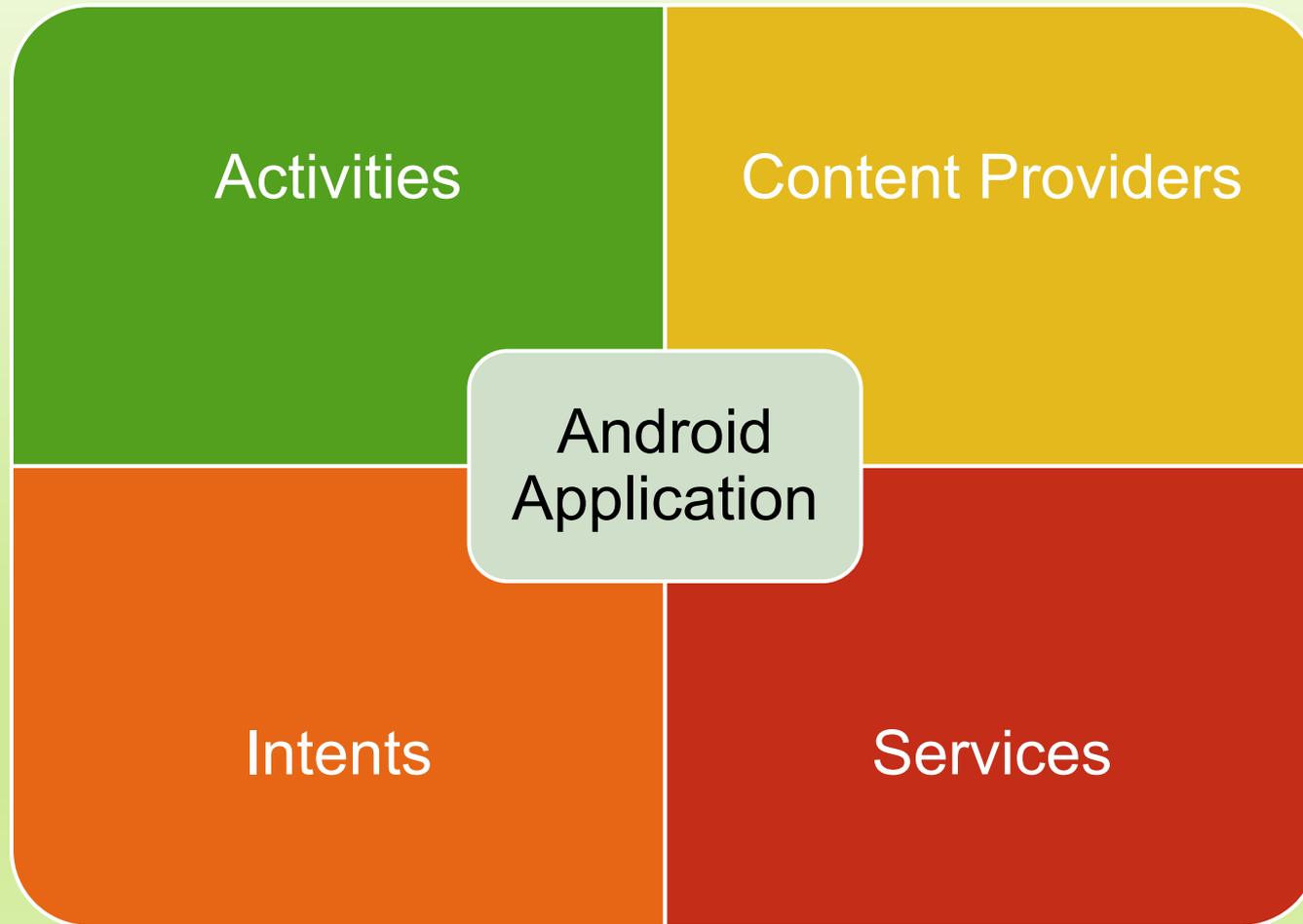
Android для разработчика

Необходимое ПО

- ▶ JDK (Java Development Kit – бесплатно распространяемый компанией Oracle Corporation комплект разработчика приложений на языке Java)
- ▶ Android IDE
- ▶ USB драйвер для Windows (для отладки на устройствах)

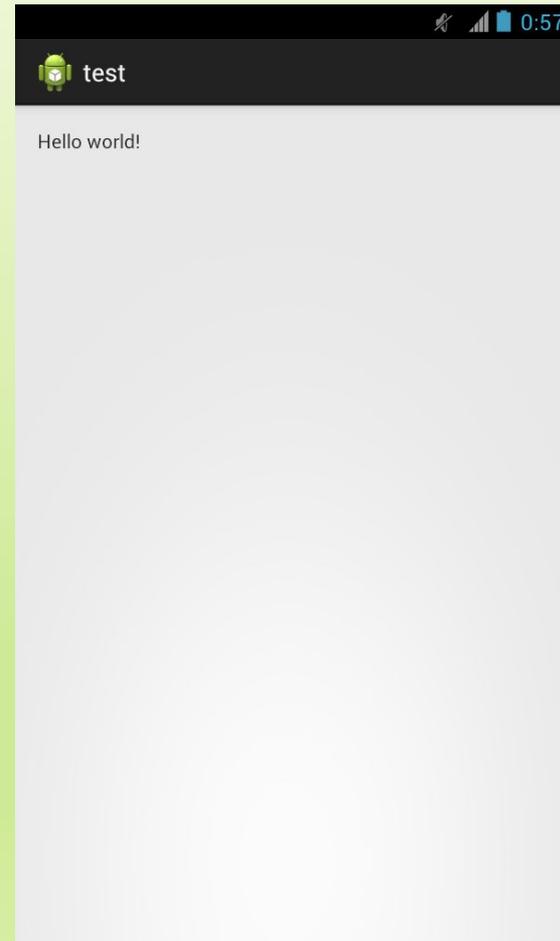


Состав приложения



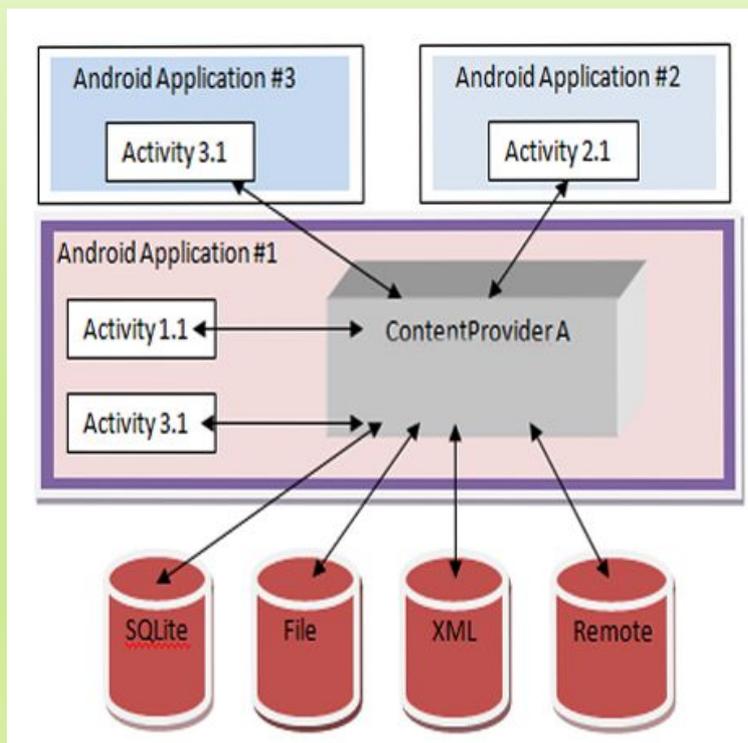
Activity

- ▶ Activity - основная единица графического интерфейса (аналог окна или экранной формы)



Content Provider

- ▶ Content Provider - “прослойка” между приложением и хранилищами данных



Intent

Intents - системные сообщения, позволяющие приложениям обмениваться информацией между собой и с операционной системой:

- ▶ поступление телефонного звонка
- ▶ приход sms-сообщения
- ▶ вставлена SD-карта
- ▶ запущена новая активность

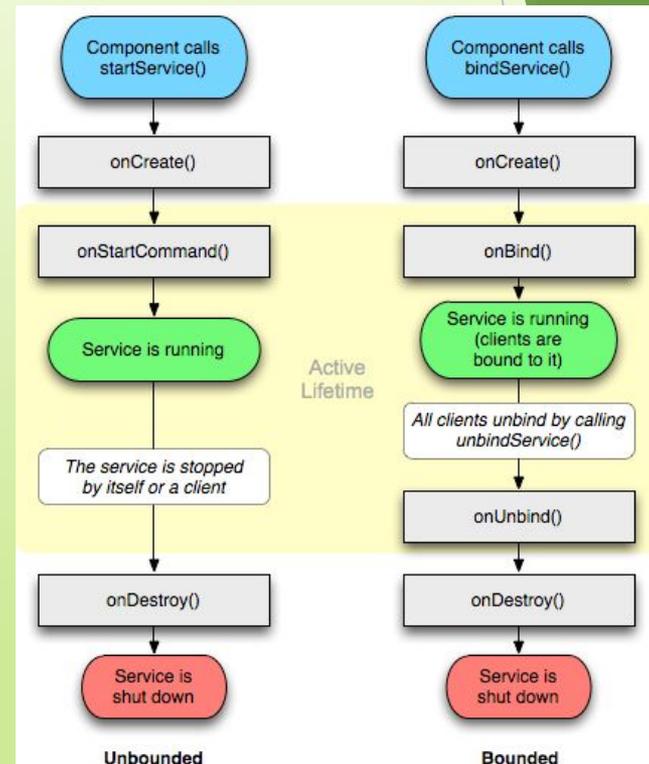
Intents - рекомендованный способ взаимодействия компонентов приложения.

Services

Приложения, не имеющие GUI и выполняющиеся в фоновом режиме.

Примеры сервисов:

- ▶ проверка электронной почты
- ▶ получение гео-информации



Эмуляторы

Эмуляция



Эмуляция (англ. *emulation*) в вычислительной технике — комплекс программных, аппаратных средств или их сочетание, предназначенное для копирования (или *эмулирования*) функций одной вычислительной системы (*гостя*) на другой, отличной от первой, вычислительной системе (*хосте*) таким образом, чтобы эмулированное поведение как можно ближе соответствовало поведению оригинальной системы (*гостя*). Целью является максимально точное воспроизведение поведения [...]

Стандартный эмулятор

- ▶ Эмулятор – виртуальное мобильное устройство, которое запускается на компьютере
- ▶ Позволяет разрабатывать и тестировать приложения без использования реальных устройств

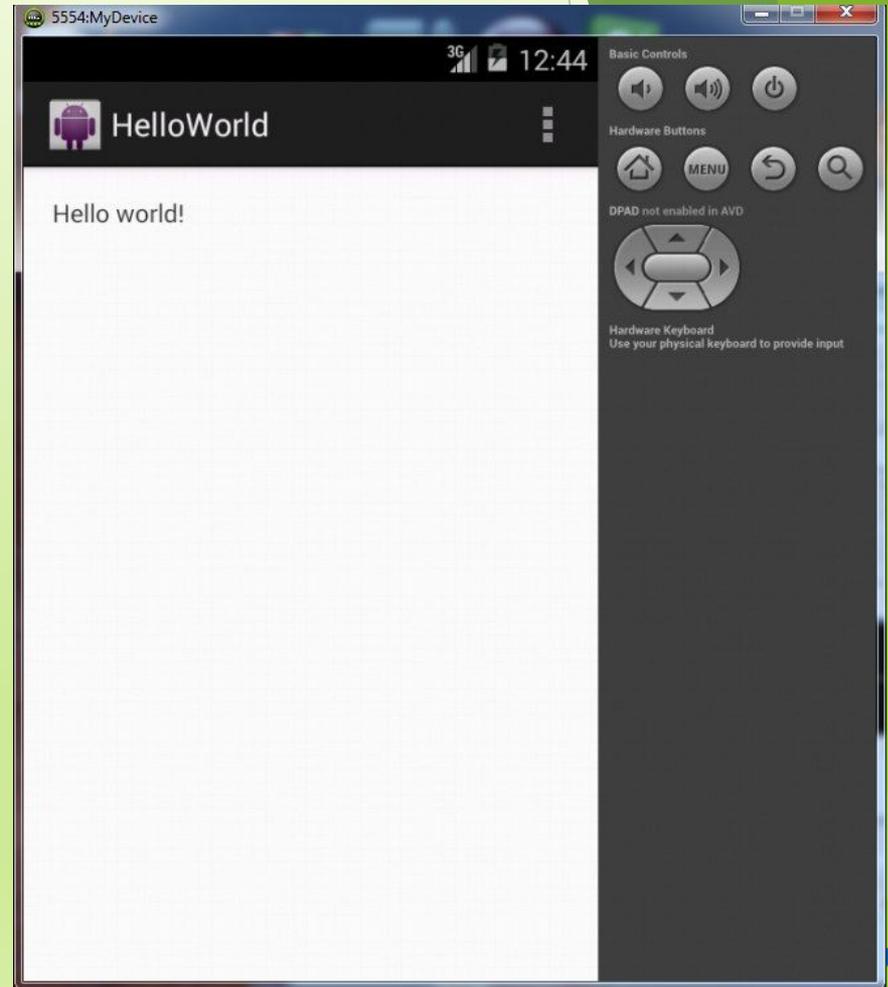


Недостатки эмуляторов

- ▶ Требует много системных ресурсов
- ▶ Из-за различий в архитектуре процессоров компьютера и смартфона медленно запускается
- ▶ В некоторых случаях эмулятора недостаточно



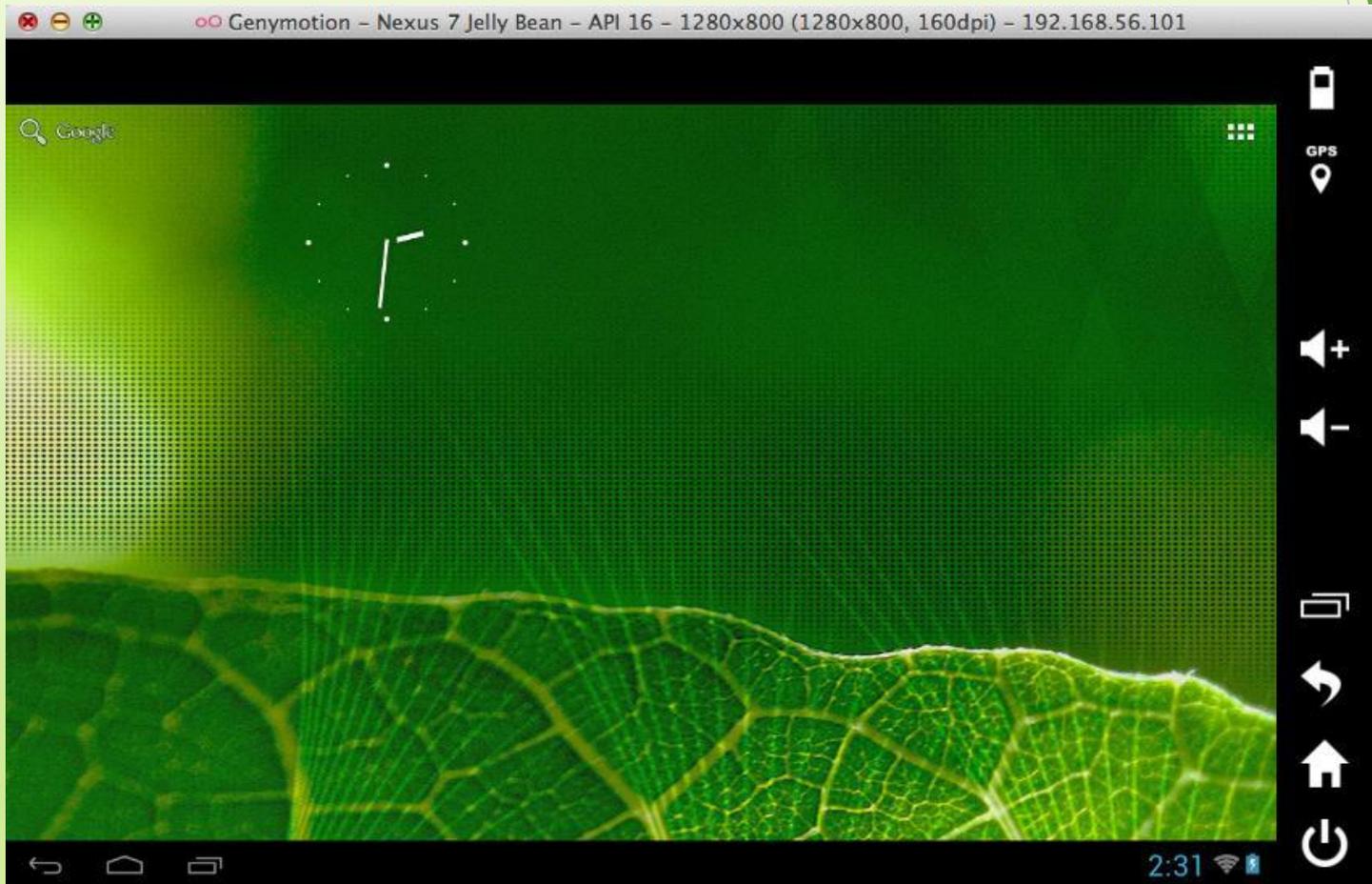
Запуск приложения на эмуляторе



Альтернативные эмуляторы

- ▶ Genymotion – быстрый эмулятор Android
- ▶ Содержит предварительно настроенные образы Android (x86 с аппаратным ускорением OpenGL)
- ▶ Genymotion доступен для Linux, Windows и Mac OS X и требует VirtualBox
- ▶ Активно развивается 😊

Genymotion



Возможности отладки на реальных устройствах

Отладка на устройствах

Настройка устройства

Включить режим отладки по USB

Разрешить установку приложений из альтернативных источников

Настройка компьютера

Установить драйвер вручную

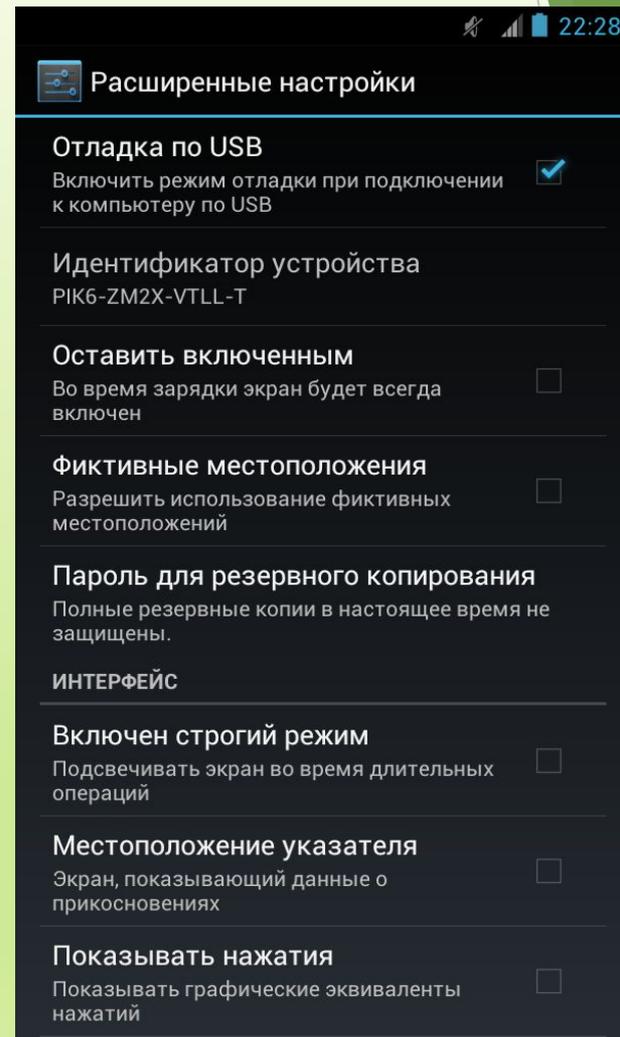
Настройка среды

Включить возможность работы с устройством

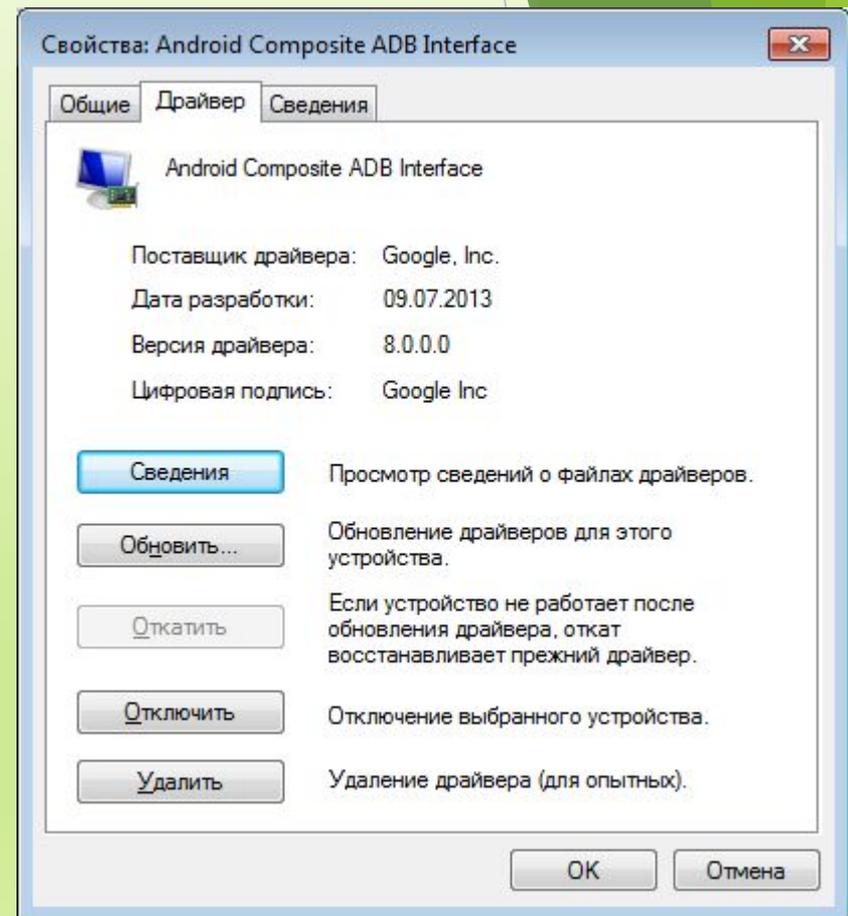
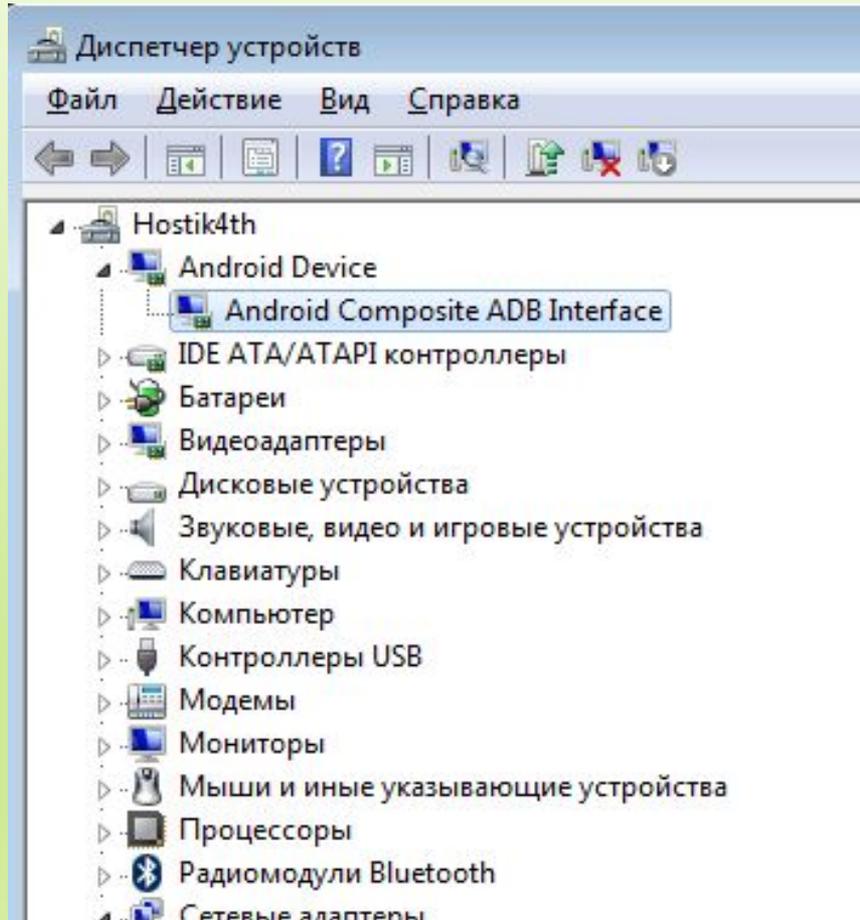
Запустить проект на устройстве

Настройка устройства

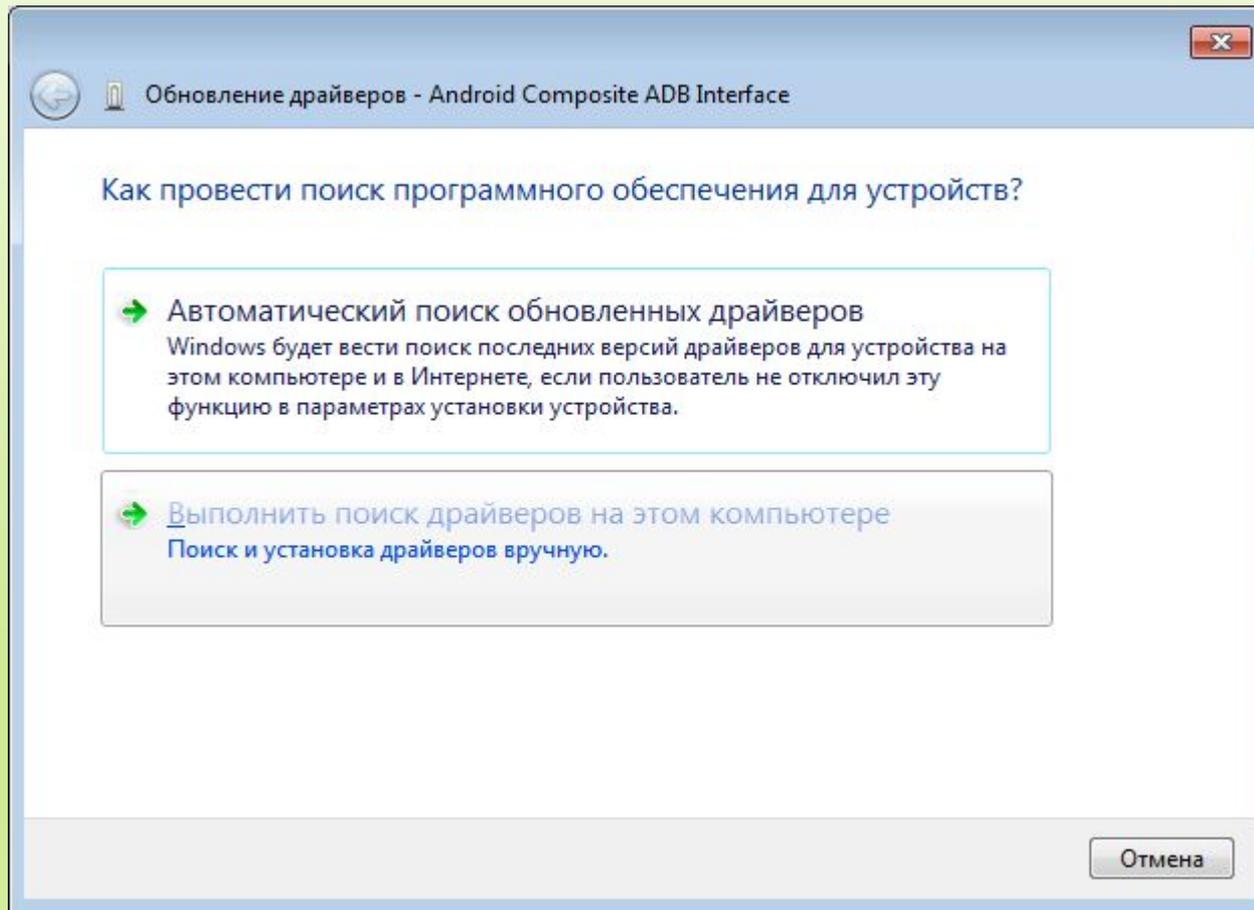
- ▶ Необходимо включить режим отладки по USB
- ▶ Для запуска *.apk, полученных не из магазина приложений Google Play, необходимо разрешить установку приложений из альтернативных источников



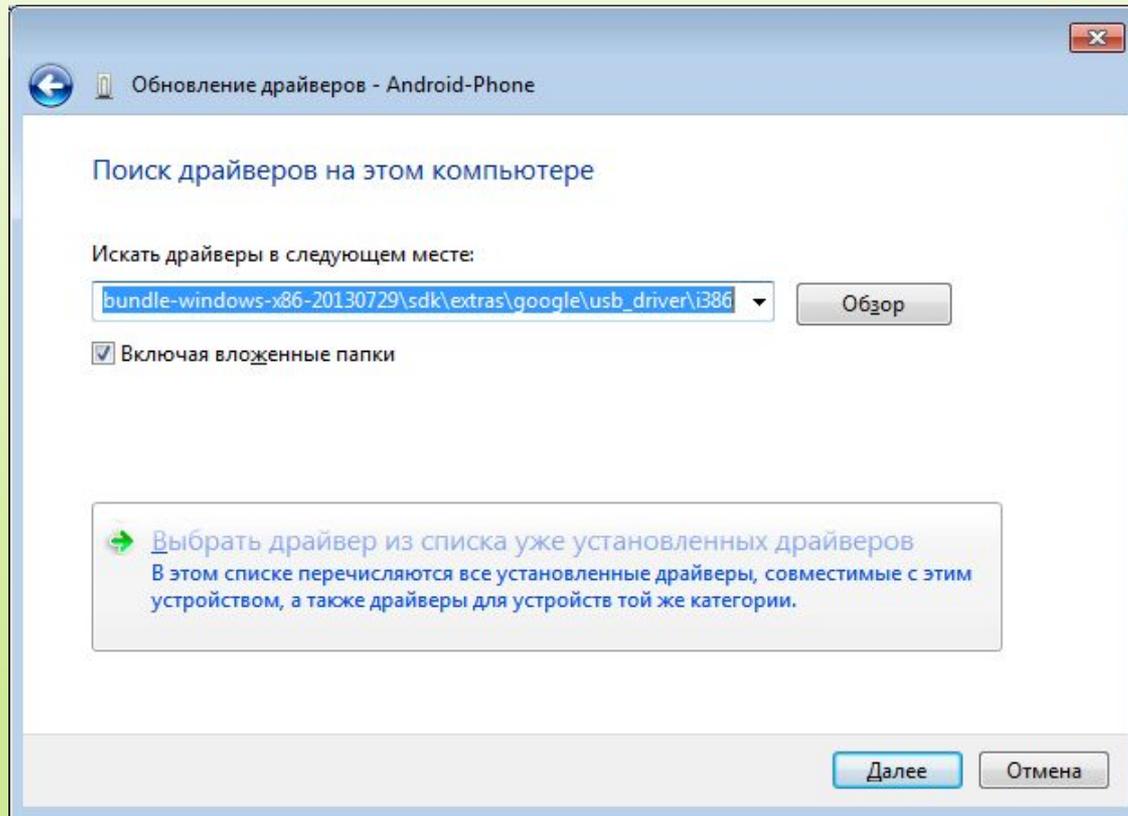
Настройка компьютера



Настройка компьютера

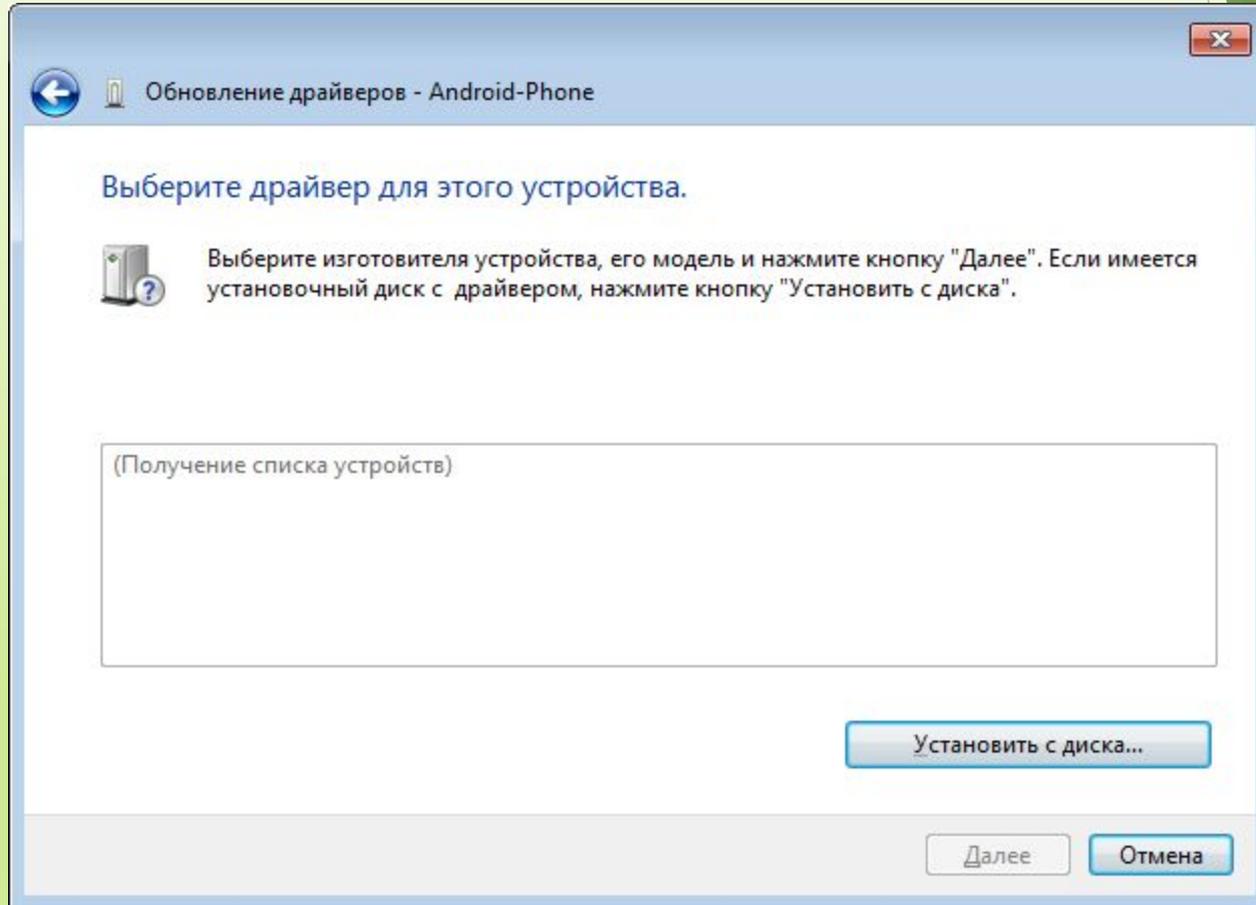


Настройка компьютера

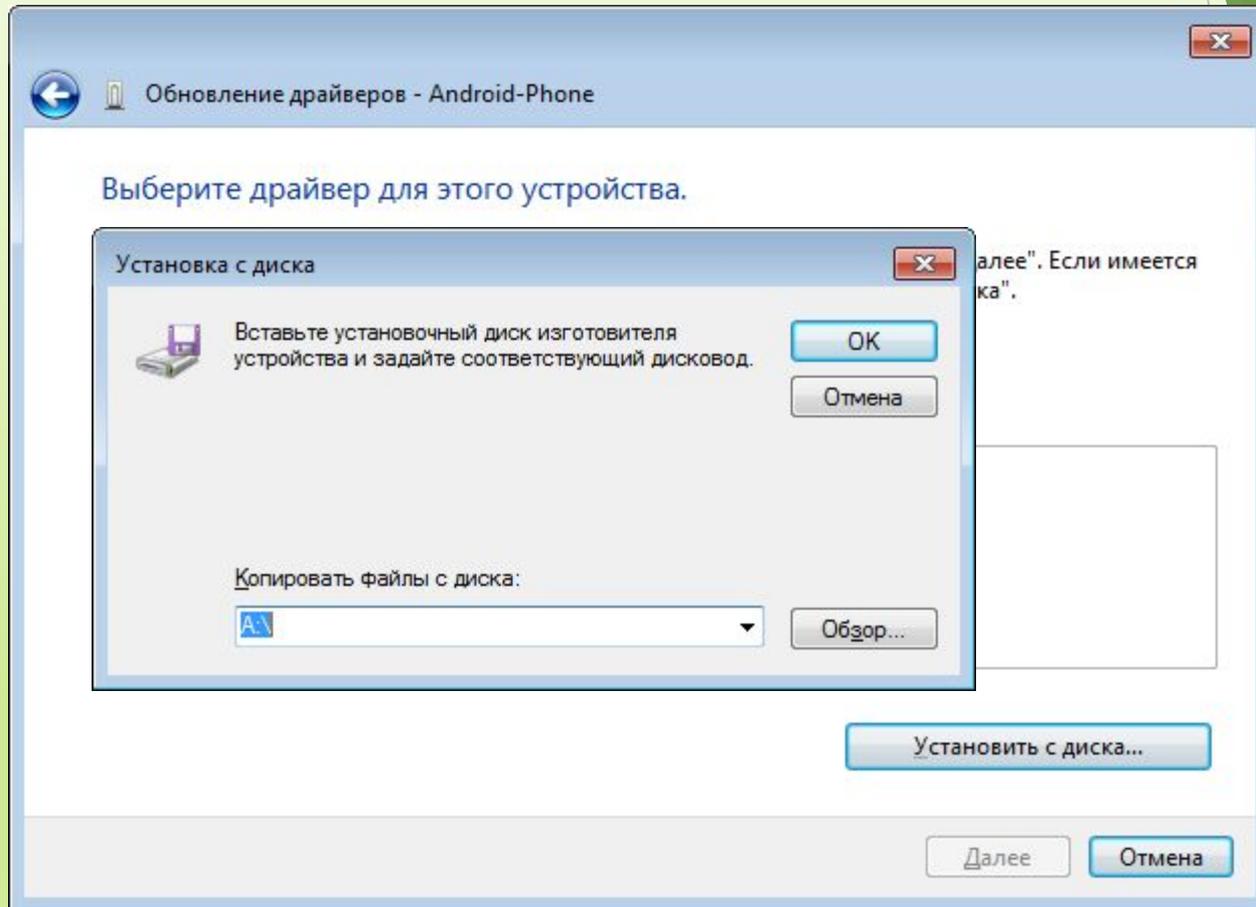


Нажать далее в этом и следующем окне!

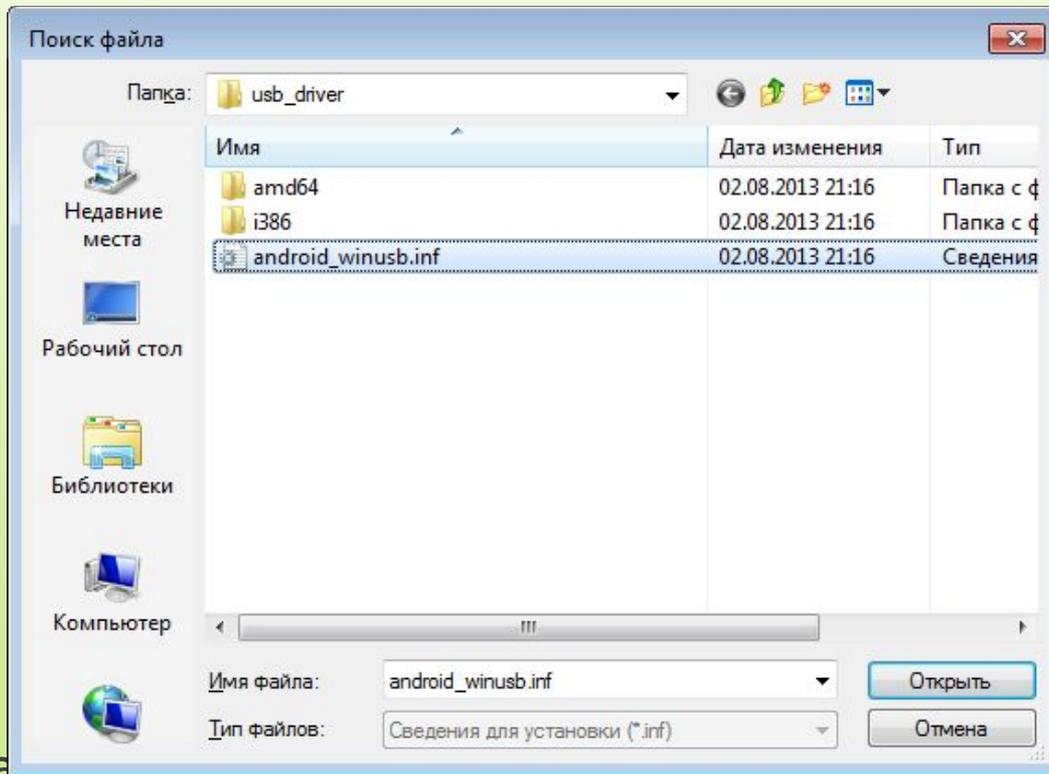
Настройка компьютера



Настройка компьютера



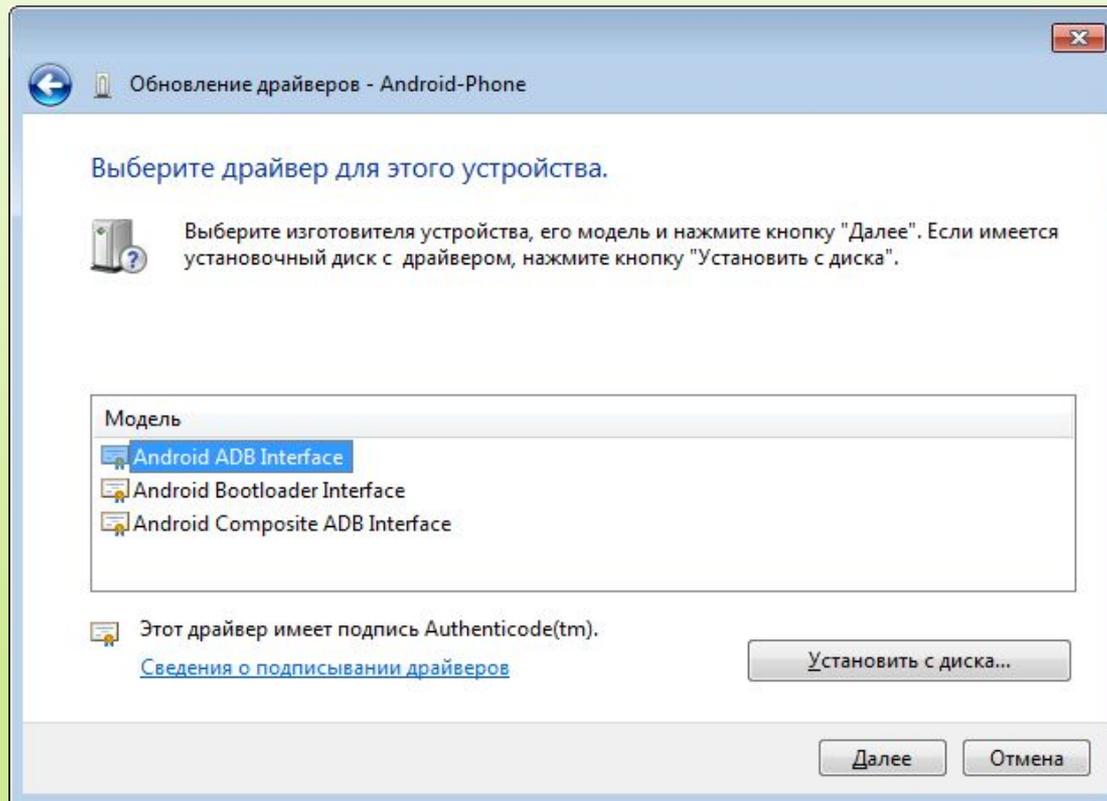
Настройка компьютера



► Папка с драйвером:

`\adt-bundle-windows-x86-20130729\sdk\extras\google\usb_driver`

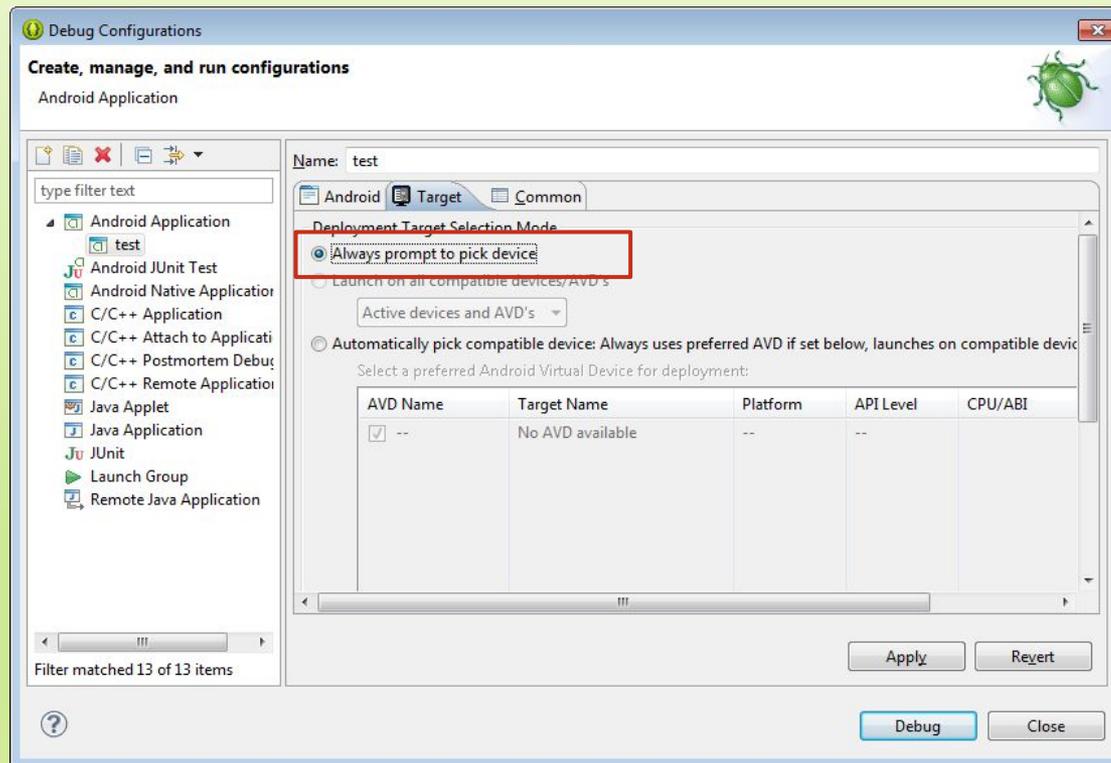
Настройка компьютера



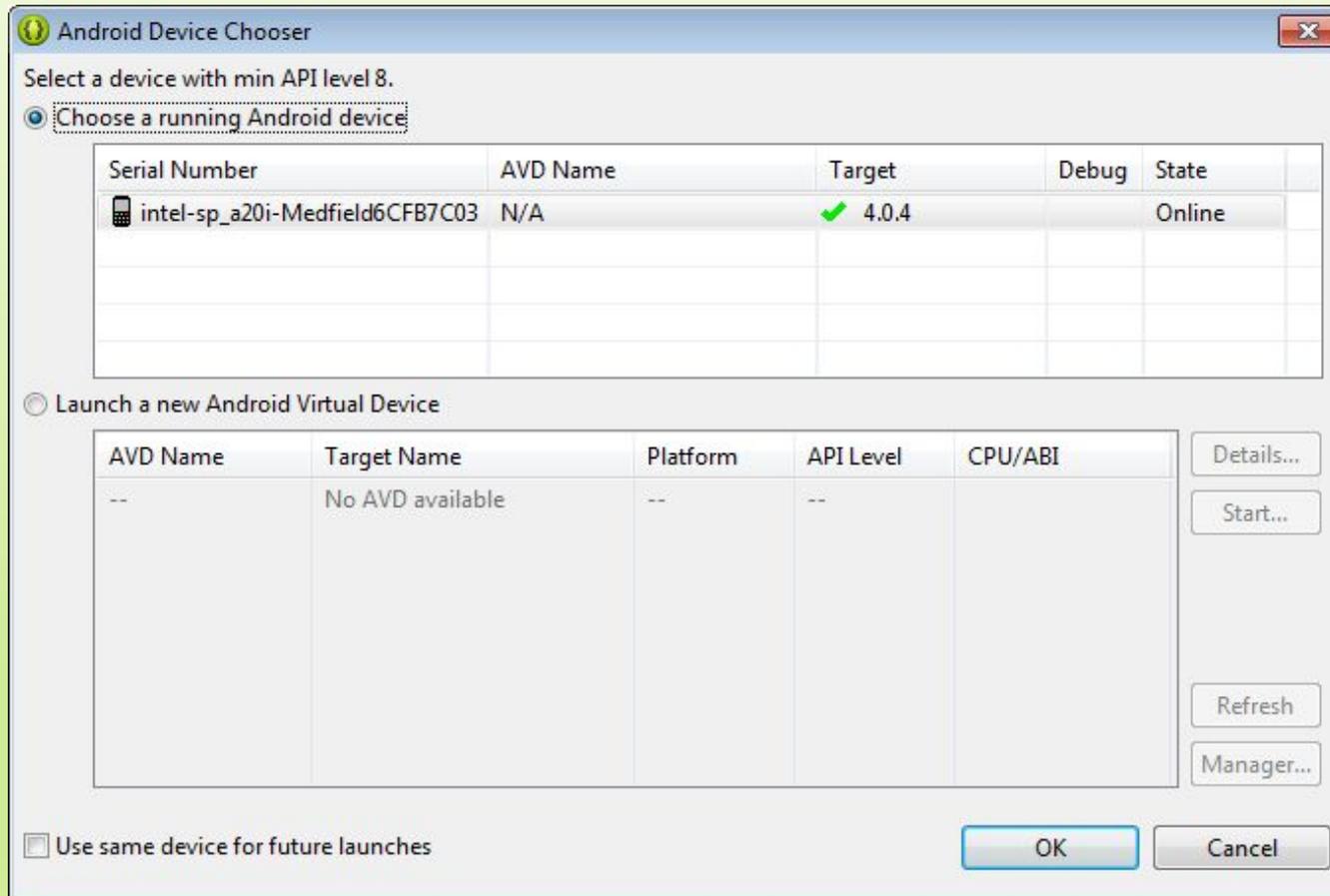
Нажать Далее и Да!

Настройка среды

- ▶ Меню Run\Debug Configurations



Настройка среды



Примеры приложений

Магазин приложений Google Play



- ▶ Магазин приложений от Google, позволяющий владельцам устройств с операционной системой Android устанавливать и приобретать различные приложения
- ▶ Учётная запись разработчика, которая даёт возможность публиковать приложения, стоит \$25

Примеры прило



Новое напоминание

Нажмите на кнопку и скажите,
что и когда вам напомнить, например,
"Разбуди меня завтра в девять"



Все напоминания Сохранить

Скажи ему по-русски!

Билеты РЖД

05:25

Петербург

Найти

Пасс) → Москва

Пасс) → Санкт-Петербур

→ Санкт-Петербур

→ Москва

Список дополнительных источников

- ▶ <http://developer.android.com>
- ▶ <http://confluence.jetbrains.com/display/IntelliJ+DEA/Android+Development>

