



Принципы переноса ранее разработанных приложений на смартфон

Лекция 4

Авторы: Березовская Ю.В., Латухина Е.А., Носов К.А., Юфрякова О.А.

Содержание

- ▶ Общие принципы портирования приложений
- ▶ Особенности разработки интерфейсов с учетом возможностей смартфона
- ▶ Работа с файловой системой и сетью при портировании на Android
- ▶ Обзор программных средств. Инструменты Intel



Общие принципы портирования приложений

Портирование приложений

- ▶ **Портирование** - адаптация некоторой программы или её части, чтобы она работала в другой среде, отличающейся от той среды, под которую она была изначально написана с максимальным сохранением её пользовательских свойств
- ▶ **Результат — порт**
- ▶ Главная задача сохранить привычные пользователю функционал и интерфейс



Причины портирования

- ▶ Различия в системе команд процессора
- ▶ Различия между способами взаимодействия операционной системы и программ (API — Application Program Interface)
- ▶ Принципиальные различия в архитектуре вычислительных систем
- ▶ Несовместимость языка программирования с целевым окружением



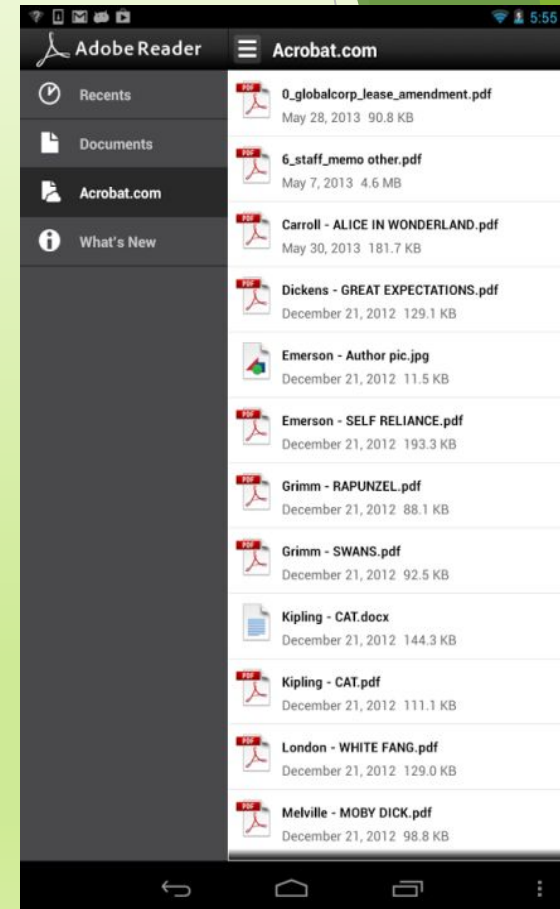
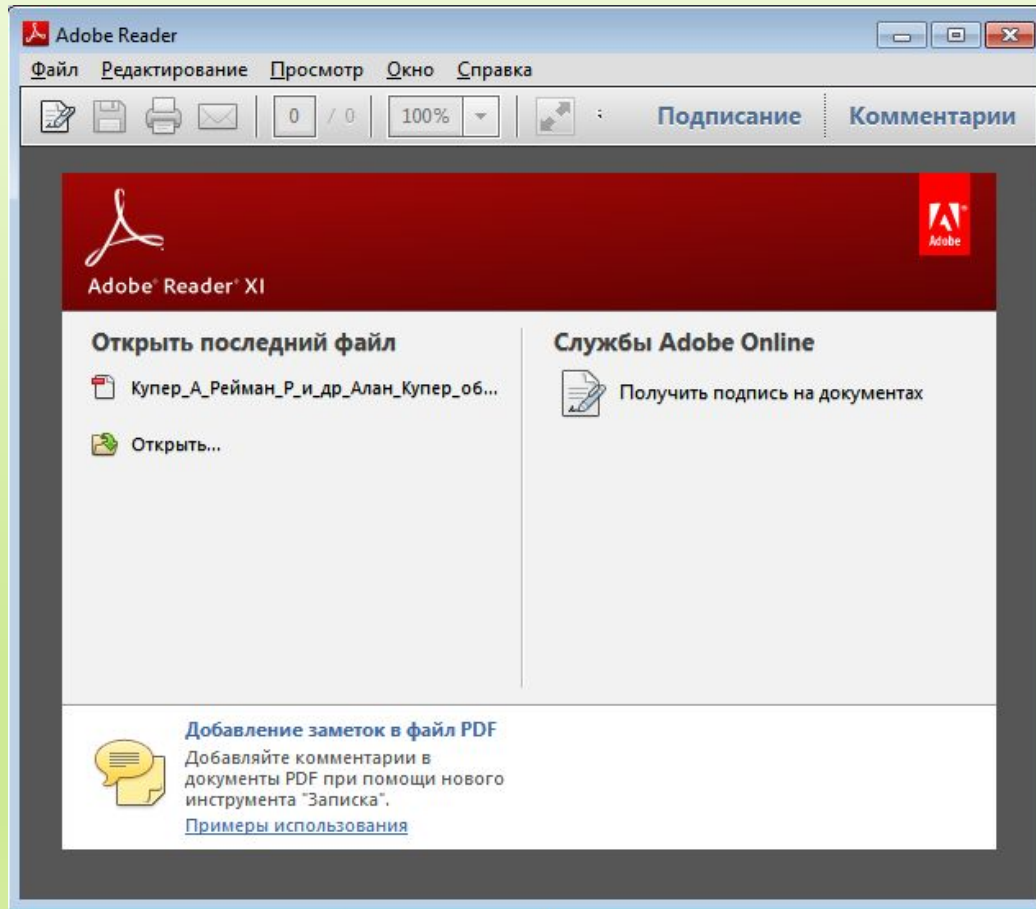
Роль стандартов в портировании

- ▶ Международные стандарты значительно упрощают портирование



- ▶ Часто портирование между платформами, реализующими один и тот же стандарт, сводится к перекомпиляции на новой платформе

Портирование приложений



Разработка приложений для смартфонов на ОС Android. Лекция 4. Тема "Принципы переноса ранее разработанных приложений на смартфон"



Технические особенности смартфонов

- ▶ Ограниченный объем памяти (оперативной и используемой для хранения)
- ▶ Сравнительно невысокая скорость обработки информации
- ▶ Слабый графический процессор
- ▶ Ограниченный срок работы устройства от батареи
- ▶ Небольшой экран



Технические возможности смартфонов

- ▶ Сенсорный экран
- ▶ Наличие датчиков gps, гироскопа, акселерометра и других
- ▶ Легкое переключение между горизонтальным и вертикальным расположением экрана



Этапы разработки мобильного приложения

- ▶ Постановка задачи
- ▶ Создание прототипа
- ▶ Определение системных требований и выбор инструмента
- ▶ Работа над графикой и дизайном приложения
- ▶ Разработка приложения
- ▶ Тестирование и оптимизация
- ▶ Подготовка и публикация приложения
- ▶ Сбор аналитики и получение обратной связи
- ▶ Исправление недочетов и выпуск обновлений



Этапы разработки мобильного приложения

- ▶ Постановка задачи
- ▶ **Создание прототипа**
- ▶ **Определение системных требований и выбор инструмента**
- ▶ **Работа над графикой и дизайном приложения**
- ▶ **Разработка приложения**
- ▶ **Тестирование и оптимизация**
- ▶ Подготовка и публикация приложения
- ▶ Сбор аналитики и получение обратной связи
- ▶ Исправление недочетов и выпуск обновлений



Отличия портирования от разработки «с нуля»

- ▶ Переработка интерфейса с учетом новых возможностей
- ▶ Выбор подходящего инструмента разработки
- ▶ Нужно подключить работающий код портируемого приложения
- ▶ В процессе тестирования и оптимизации возможны значительные изменения в коде исходного приложения





Родные (нативные) платформы позволяют создавать приложения, которые замечательно выглядят и воспринимаются, а также обеспечивают сравнительно высокую производительность. Для каждой платформы нужно писать свой код.

Портирование нативных приложений наиболее сложное.



Гибридные инструменты и среды сочетают в себе простоту программирования и универсальность web-приложений с блеском родных приложений.

Портирование гибридных приложений зависит от многих факторов, которые могут как упрощать, так и усложнять процесс перехода на другую платформу.



Мобильные Web-приложения обычно пишут на HTML5, CSS и JavaScript.

Можно написать один раз и запускать на разных платформах.

Часто уступают родным по внешнему виду, восприятию и производительности.

Портирование не представляет особых сложностей.

Платформы разработки

Разделение кода

- ▶ Перед портированием необходимо произвести разделение платформенно-зависимых и платформенно-независимых частей программы



Особенности разработки интерфейсов с учетом возможностей смартфона

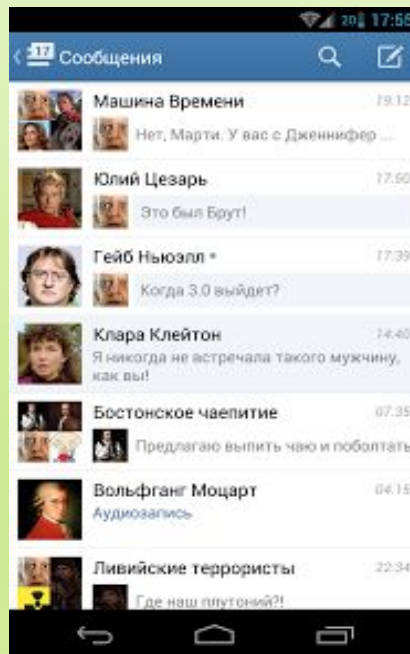
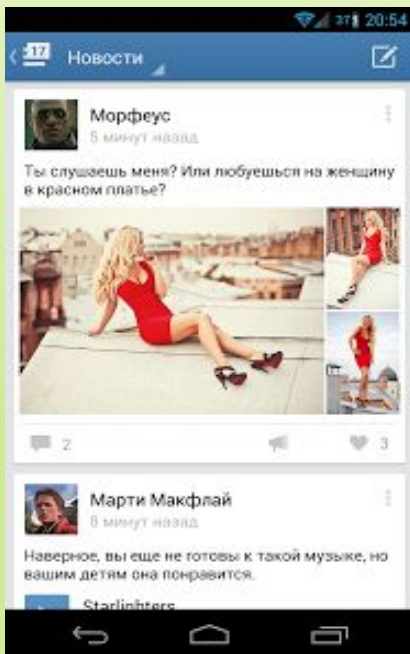
Рекомендации по разработке интерфейсов

- ▶ Элементы управления на экране должны бросаться в глаза
- ▶ Располагайте наиболее важное в центре экрана
- ▶ Используйте минимальное количество визуальных элементов
- ▶ Дизайн приложения должен быть максимально простым и понятным



Рекомендации по разработке интерфейсов

- ▶ Создавайте визуальные якоря, помогающие пользователям ориентироваться



Рекомендации по разработке интерфейсов

- ▶ Элементы управления следует делать достаточно крупными, чтобы их можно было активировать пальцами. Минимальный размер - 7-10 мм
- ▶ Используйте крупные шрифты без засечек
- ▶ Четко указывайте наличие дополнительной информации за пределами экрана. Многие люди не привыкли к идее маленького экрана, требующего прокрутки информации



Рекомендации по разработке интерфейсов

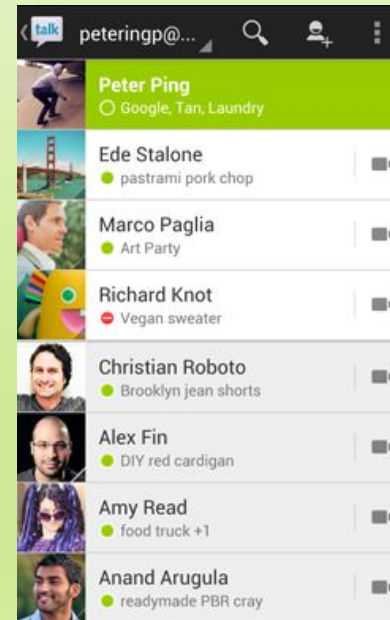
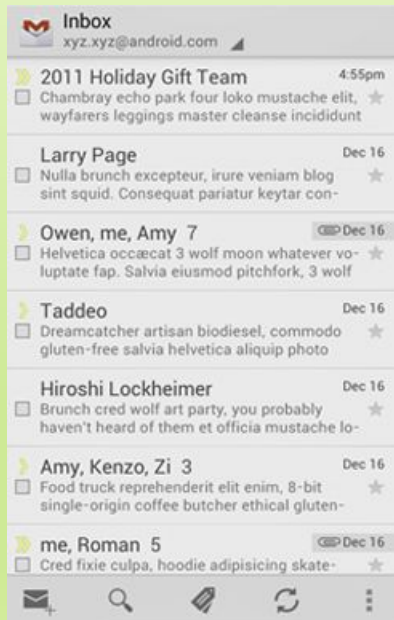
- ▶ Если предполагается перевод интерфейса на другие языки, необходимо учитывать, что в некоторых языках слова длиннее, чем в других. Необходимо выделять элементам, содержащим текст, достаточное количество драгоценного места на экране
- ▶ Учитывайте мобильный контекст использования. Люди имеют привычку пользоваться смартфонами на ходу, стоя, в общественном транспорте и т.д.
- ▶ Обязательно учитывайте официальные рекомендации разработчика платформы



Рекомендации по дизайну приложений для Android

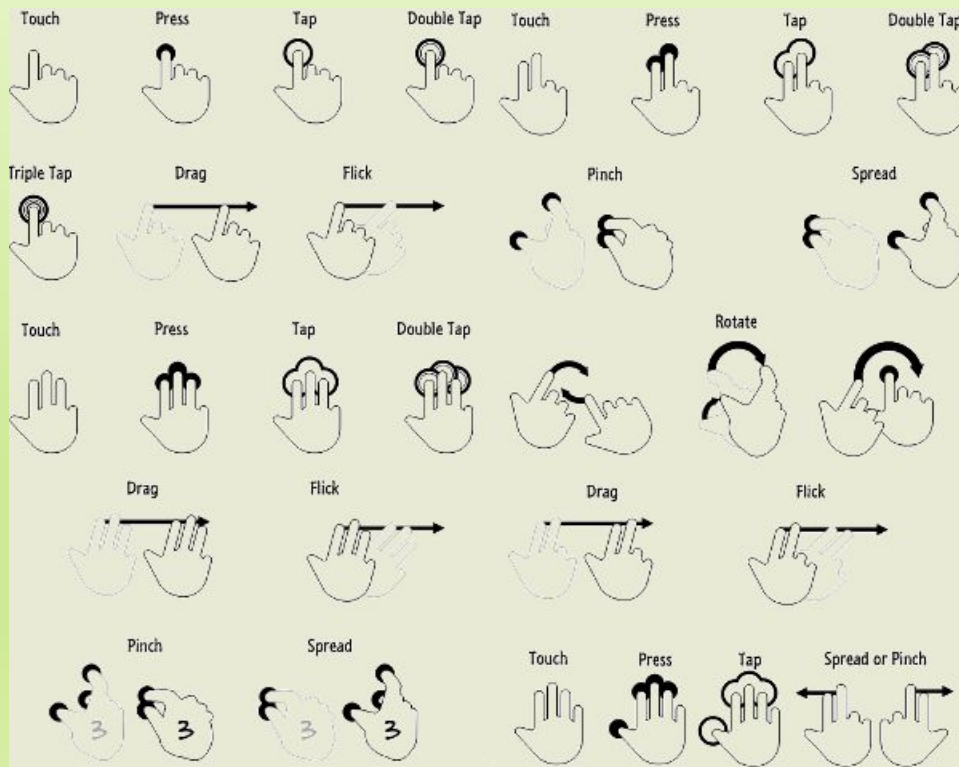
- ▶ Официальные рекомендации от Google

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



Использование технических возможностей смартфонов при проектировании интерфейсов

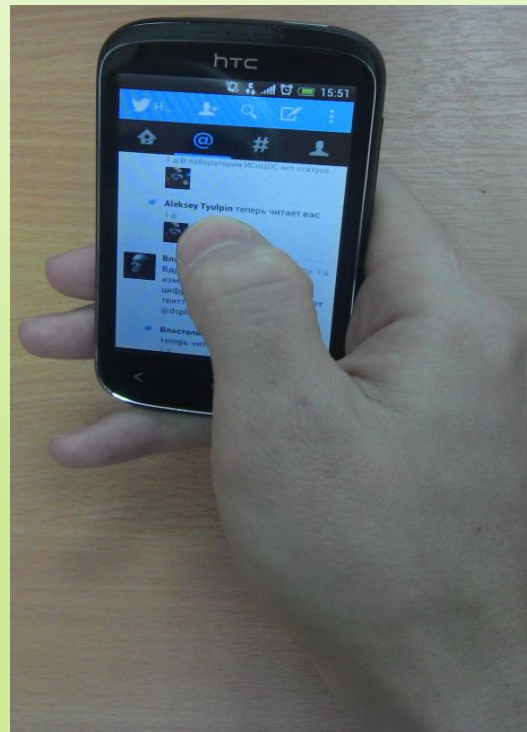
- ▶ **Мультитач (мультикасание)** - технология, позволяющая одновременно определить координаты двух и более точек касания



Разработка приложений для смартфонов на ОС Android. Лекция 4. Тема "Принципы переноса ранее разработанных приложений на смартфон"

Расположение смартфона в пространстве

- ▶ Левой или правой рукой



Расположение смартфона в пространстве

- ▶ Двумя руками



Особенности сенсорного управления

- ▶ В первую очередь опирается на визуальную обратную связь
- ▶ Отсутствие физического ощущения нажатия клавиши (тактильной обратной связи)
- ▶ Можно использовать вибрацию
- ▶ Механика длительного касания. Можно использовать для вызова контекстного меню или дополнительных параметров



Влияние размера экрана

- ▶ Разрешение - общее число физических пикселей на экране
- ▶ Экранная плотность - количество пикселей в физической области экрана, как правило, называют **DPI** (точек на дюйм)



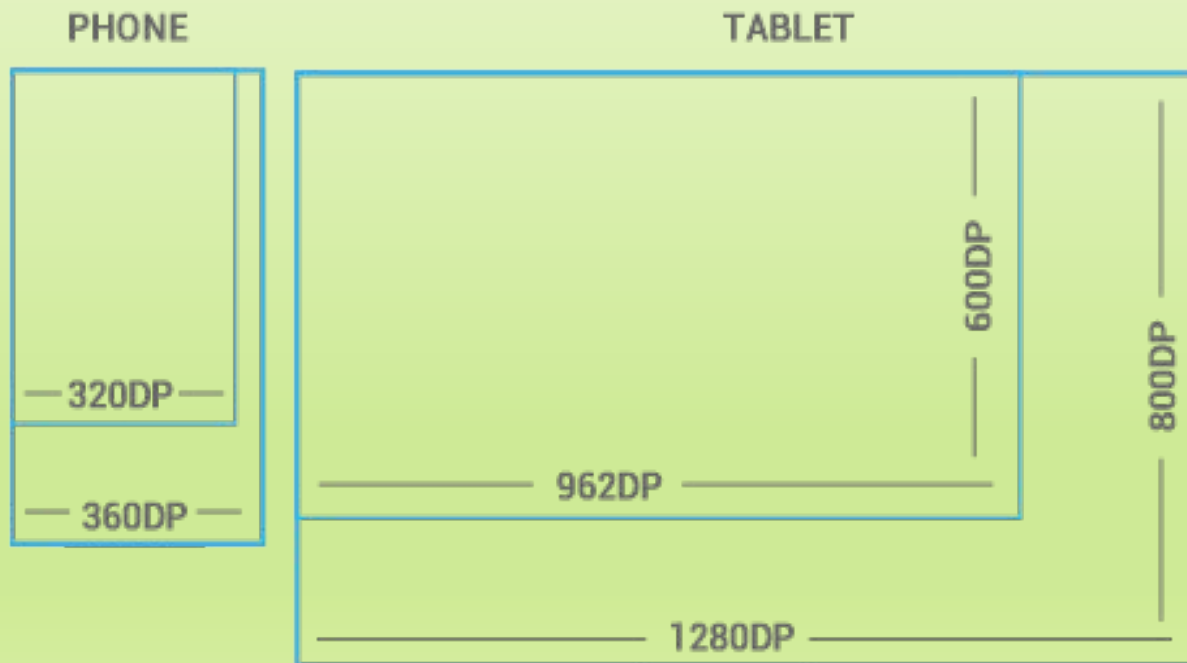
Влияние размера экрана

- ▶ Плотностно-независимая точка (**DP**) - виртуальная единица-пиксел, которая используется при определении интерфейса макета, чтобы сделать размеры макета или его положения независимым от плотности образом
- ▶ Эквивалентна одному физическому пикселю на 160 **DPI** (базовая плотность)
- ▶ Формула преобразования единиц **DP** в пиксели:
пиксель = **DP** * (**DPI** / 160)



Телефоны и планшеты

- ▶ 2 типа устройств: телефонный (до 600 DP) и планшетный (больше или равно 600 DP)



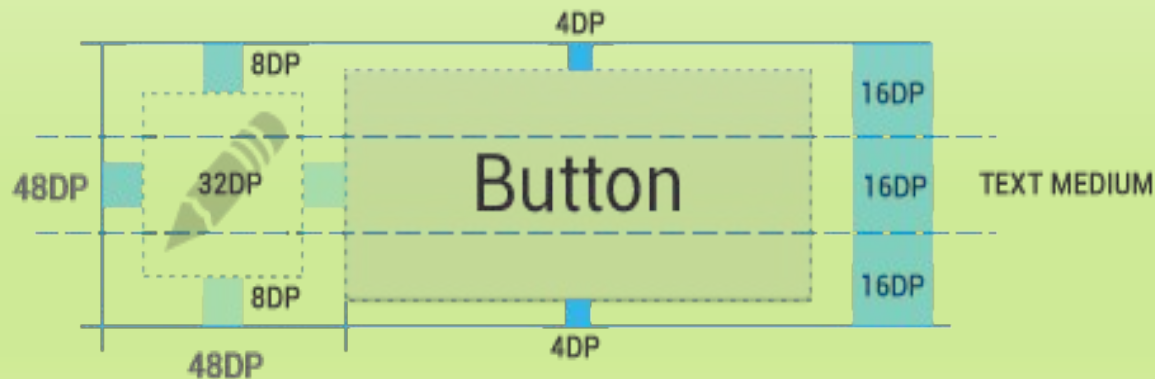
5 основных плотностей экрана

- ▶ **LDPI** (низкий)
- ▶ **MDPI** (средний)
- ▶ **HDPI** (высокий)
- ▶ **XHDPI** (очень высокий)
- ▶ **XXHDPI** (очень-очень высокий)



Минимальный размер элементов управления

- ▶ Элементы управления должны быть размером по крайней мере 48 DP
- ▶ Расстояние между элементами управления не должно быть меньше 8 DP



Портирование графики, работа с файловой системой и сетью

Разработка интерфейсов для разных экранов

- ▶ Необходимо подключить различные варианты компоновки интерфейсов для различных плотностей экранов



Указание альтернативных конфигураций

- ▶ Создайте новый каталог в `res/` с именем в виде `<resources_name>-<config_qualifier>`, где
- ▶ `<resources_name>` - это имя каталога соответствующих ресурсов по умолчанию
- ▶ `<qualifier>` - имя, определяющее индивидуальную конфигурацию устройств
- ▶ Сохраните соответствующие файлы в этот новый каталог



Указание альтернативных конфигураций

Настройка выравнивания (layout)

res/

layout/

main.xml (по умолчанию)

layout-ar/

main.xml (для арабских стран)

layout-ldrtl/

main.xml (для любого языка, предусматривающего чтение справа налево, кроме арабского, потому что соответствующий квалификатор имеет более высокий приоритет)



Указание альтернативных конфигураций

Настройка графики (drawable)

```
res/  
  drawable/  
    icon.png  
    background.png  
  drawable-hdpi/  
    icon.png  
    background.png
```

hdpi указывает, что ресурсы в этой директории рассчитаны для устройств с экраном высокой плотности. Изображения в каждом из этих каталогов рассчитаны на различные типы экранов, однако имена файлов совпадают



Работа с файловой системой и сетью

- ▶ В целях безопасности одно приложение не может получить доступ к файлам другого. Однако подобное ограничение не распространяется на файлы, расположенные на SD-карте
- ▶ При разработке приложений следует придерживаться следующего правила: критичные для работы приложения файлы записываются в память устройства, а дополнительные, особенно имеющие большой объем, лучше выносить на карту памяти



Работа с файловой системой и сетью

- ▶ Для периодической синхронизации можно сделать в настройках опцию «Синхронизировать только через wi-fi»
- ▶ Следует жестко контролировать подключения и отключения от сети, время работы сервисов
- ▶ Следует использовать защищенные сетевые протоколы (например, HTTPS) для передачи пользовательских паролей и другой конфиденциальной информации



Обзор программных средств. Инструменты Intel. Intel XDK

Marmalade Juice

- ▶ **Marmalade SDK** – кроссплатформенное SDK от Ideaworks3D Limited
- ▶ Представляет собой набор библиотек, образцов, инструментов и документации, необходимых для разработки, тестирования и развертывания приложений для мобильных устройств
- ▶ Однократное написание программы и компилирование ее на все поддерживаемые платформы



Marmalade Juice



Поддерживает различные платформы:

- ▶ iOS версии 4.3 и выше
- ▶ Android версии 2.1 и выше (в том числе x86)
- ▶ Windows Phone 8
- ▶ Tizen 2.2 и др
- ▶ Windows XP и выше
- ▶ OSX 10.6 и выше



Marmalade Juice

- ▶ Код пишется на C++ или с использованием HTML5, может содержать скриптовые вставки на Lua
- ▶ Многие игры, прочно обосновавшиеся в топах магазинов приложений, от казуальных «Cut the Rope» и «Plants vs. Zombies» до крутых «Call of Duty» и «Need for Speed» были разработаны с использованием Marmalade
- ▶ Самая дешевая лицензия - 15\$ за месяц использования (бесплатный пробный период 30 дней)



Android NDK

- ▶ Набор инструментов, позволяющих реализовать части приложения с использованием языков C и C++
- ▶ Можно использовать для повышения производительности некоторых приложений, но необходимо соблюдать меры предосторожности, поскольку желаемого результата можно добиться не всегда
- ▶ Не стоит использовать Android NDK только потому, что вы хорошо знаете C/C++



Android NDK

Предоставлении разработчикам следующие
ВОЗМОЖНОСТИ:

- ▶ Компиляция встроенной библиотеки C/C++ для использования (при вызове кодом Java) в пакете Android
- ▶ Перекомпиляция встроенных библиотек ARM для x86 (микроархитектура Intel® Atom™) с возможностью портирования при необходимости



Intel® C++ Compiler для Android

- ▶ Компилятор для создания приложений C и C++, ориентированные на устройства Android на базе Intel Atom
- ▶ Можно подключить инструменты Intel® C++ Compiler к Android NDK
- ▶ Можно использовать в качестве автономного компилятора



Intel® C++ Compiler для Android

- ▶ Разработка приложений для Android мобильных устройств на базе процессоров Intel
- ▶ Совместимость с Android NDK
- ▶ Совместимость с GNU C++
- ▶ Разработка на Windows, OS X или Linux.
- ▶ Поддержка Eclipse и командной строки
- ▶ Простая и быстрая загрузка и установка
- ▶ Поддержка Android Jelly Bean и Android KitKat



Intel® Integrated Native Developer Experience beta

- ▶ Бета-версия кросс-платформенного комплекта разработчика от Intel для быстрого и легкого создания приложений для Android и Windows
- ▶ Предоставляет полный набор C++/Java инструментов, библиотек и примеров для настройки среды, создания кода, компиляции, отладки и анализа на устройствах на базе процессоров Intel® и некоторых возможностей Android-устройств на базе ARM
- ▶ Ранее была известна под именем Beacon Mountain

Возможности Intel® INDE: медиа

Легко добавить видео и аудио расширения, которые работают на актуальных версиях популярных смартфонов и планшетов. Intel® INDE Media Pack содержит примеры приложений, позволяющих:

- Работать с камерой и осуществлять захват содержимого экрана
- Редактировать видео
- Работать с потоковым видео
- Поддерживать устройства Android на базе архитектур Intel и ARM

Возможности Intel® INDE: потоки

- ▶ Эффективная реализация параллелизма с помощью Intel® Threading Building Blocks (Intel® TBB)
- ▶ Intel® TBB предназначена для разработки высокопроизводительных и масштабируемых приложений
- ▶ Может работать с устройствами Android с версией не меньше 4.3 и клиентскими версиями Microsoft Windows 7-8.1

Возможности Intel® INDE: компиляция и построение кода

- ▶ Используется Intel® C++ Compiler для Android
- ▶ Максимальная производительность графики достигается посредством использования Compute Code Builder
- ▶ Работает с Google Renderscript и OpenCL™
- ▶ Можно использовать в автономном режиме или интегрировать с Microsoft Visual Studio или Eclipse

Возможности Intel® INDE: анализ и отладка

- ▶ В режиме реального времени для анализа исполнения кода, использования CPU/GPU, выполнения различных задач
- ▶ Для анализа и отладки захваченных кадров
- ▶ Содержит Intel® Graphics Performance Analyzer (Intel® GPA) System Analyzer, Intel® GPA Platform Analyzer, Intel® GPA Frame Analyzer и Intel® Frame Debugger



Возможности Intel[®] INDE

- ▶ Можно быстро настроить среду под себя, потратив на это не часы, а минуты
- ▶ Можно выбрать часть инструментов и установить только их
- ▶ Может работать с Google Android SDK (включая Eclipse), плагином Microsoft Visual Studio, Android NDK и другими средствами



Intel XDK

- ▶ Поддерживает разработку кроссплатформенных мобильных приложений
- ▶ Позволяет разрабатывать приложения, используя следующие HTML5, CSS3 и JavaScript
- ▶ Более полный обзор среды приведен в [первой части данного курса](#)



Дополнительные источники

- ▶ Статья Джона Бергмана «Реализация сенсорного интерфейса в новых и существующих играх» (<http://software.intel.com/ru-ru/node/394259>)
- ▶ Официальные рекомендации Google по разработке дизайна приложений для платформы Android (<http://developer.android.com/design/index.html>)
- ▶ Портирование OpenGL на Android на Intel® Atom (<http://software.intel.com/ru-ru/articles/opengl-android-intel-atom-1>)
- ▶ Учебный курс «Введение в разработку приложений для ОС Android» <http://www.intuit.ru/studies/courses/12643/1191/info>
- ▶ Справочник по работе с файлами в Android <http://forum.startandroid.ru/viewtopic.php?f=26&t=860>
- ▶ Работа с сетью в Android: трафик, безопасность и батарейка <http://habrahabr.ru/company/mailru/blog/185696/>
- ▶ Портирование кода на мобильные устройства iOS/Android <http://habrahabr.ru/company/mailru/blog/182506/>
- ▶ Методика портирования приложений NDK Android <http://software.intel.com/ru-ru/articles/ndk-android-application-porting-methodologies>
- ▶ Intel® C++ Compiler 14.0 for Android <https://software.intel.com/ru-ru/c-compiler-android/>

