МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«ВОЛГОГРАДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ»

Создание приложений для мобильной операционной системы Android с использованием Android SDK

А.Е. Андреев, В.В. Камнев

Методические указания
к лабораторным работам по дисциплине
«Мобильные и встраиваемые операционные системы»

Волгоград 2015

УДК 004.42 (075) + 004.451 (075)

Р е ц е н з е н т
доцент кафедры «САПР и ПК»
канд.техн.наук
Матохина Анна Владимировна

Издаётся по решению редакционно-издательского совета
Волгоградского государственного технического университета.

Создание приложений для мобильной операционной системы Android с использованием Android SDK: метод. указания к лабораторным работам по дисциплине «Мобильные и встраиваемые операционные системы». / сост. А.Е. Андреев, В.В. Камнев – Волгоград : ИУНЛ ВолгГТУ, 2015. - 21 с.

Предназначены для студентов третьего курса, обучающихся по направлению 09.03.01 «Информатика и вычислительная техника»

Создание приложений для мобильной операционной системы Android с использованием Android SDK

Цели работы: 1. Рассмотреть необходимые средства разработки для создания приложений под операционную систему (ОС) Android
2. Научиться создавать простые приложения для ОС Android.
3. Научиться использовать эмулятор ОС Android.

1 Общие сведения об операционной системе Android

Мобильная операционная система (ОС) Google Android появилась в 2008 году. Основана на ядре Linux 2.6. Является достаточно открытой ОС, в том числе позволяет пользовательским приложениям публиковать свои сервисы для других приложений и пользоваться такими сервисами. Так же, как и iOS (и Windows Phone 8 – WP8) использует концепцию распространения приложений через он-лайн магазин Google Play, но позволяет устанавливать и приложения с носителей.

По состоянию на конец первого полугодия 2013 года доля смартфонов с Google Android составила 79% по количеству продаваемых новых устройств, в 3 квартале 2015 – уже 84,7%.

2 Программное обеспечение для разработки под Android

Для разработки программного обеспечения (ПО) под Android к компьютеру не предъявляется каких-то особых требований, хотя эмулятор Android достаточно требователен и работает заметно медленнее эмулятора WP8.

В настоящее время (на конец 2015 года) для разработки приложений под ОС Android доступны множество средств. Основное средство – Android Studio от Google Inc. [6.1] позволяет создавать приложения на Java. Пакет Android Studio версии 1.4 включает в себя интегрированную среду разработки (IDE) Android Studio, основанную на IntelliJ IDEA, набор инструментов Android SDK, платформу Android 6.0 (Marshmallow), эмулятор с образом операционной системы Android 6.0 и набором Google API. Android Studio с IDE IntelliJ IDEA была впервые представлена 15 мая 2013 года на конференции Google I/O. До этого для разработки Android приложений был доступен пакет (bundle) Android SDK + ADT (Android Development Tools) с IDE Eclipse.

Intel XDK [6.2] – бесплатный инструмент кросс-платформенной разработки веб-приложений с помощью HTML5 и JavaScript. При помощи единого инструмента позволяет создавать, тестировать и развертывать гибридные приложения и веб-приложения. Пакет такого приложения можно опубликовать в Android, iOS, Amazon Appstore, Windows Store и т.д.

Кроме того, существуют многочисленные каркасы [6.3], в том числе коммерческие, облегчающие создание приложений под Android.

Поскольку чаще всего разработку под Android ведут на языке Java, основное необходимое для разработки ПО в настоящее время включает Java Development Kit и Android SDK. Также необходима среда разработки с набором плагинов и шаблонов для Android приложений, и все это уже включено в Android Studio. Опционально можно установить по ссылке [6.4] ускоритель эмулятора от Intel (гипервизор Intel® Hardware Accelerated Execution Manager (HAXM)) при наличии на машине разработчика процессора Intel® с поддержкой технологий Intel® VT-x, Intel® EM64T (Intel® 64), и Execute Disable (XD) Bit. В данной работе использовался JDK версии 8 update 60 для Windows 64, а также пакет (bundle) Android Studio Bundle версии 141.2288178. JDK можно скачать с сайта www.oracle.com, а Android Studio Bundle — с сайта developer.android.com. При установке необходимого ПО обычно не возникает сложностей, и единственное, на что следует обратить

внимание, это HAXM: при установке запрашивает максимальный объем оперативной памяти, доступный гипервизору. Следует указать не менее 1600Мб, так как эмулятор Andriod 6.0 требует чуть более 1550Мб. Если заданный лимит в установщике HAXM менее требуемого эмулятором, эмулятор не задействует режим аппаратного ускорения и будет работать крайне медленно.

Для размещения разработанного приложения в смартфоне разработчику достаточно перенести на телефон пакет *.apk с приложением, а также соответствующий файл AndroidManifest.xml.

Для разработки на C/C++ необходимо установить пакет Native Development Kit (NDK) [6.5]. Использование NDK, как правило, требуется при разработке высокопроизводительных приложений, таких, как игровые движки, обработчики сигналов, физические симуляторы с оптимизацией кода (в том числе, возможно использование интринсиков) под архитектуру мобильного CPU (ARM, NEON, x86 (32- и 64-бит), MIPS). Кроме того, NDK позволяет использовать ваши или сторонние C/C++ библиотеки.

В целом разработка для Android выполняется аналогично разработке для Windows Phone, размещение приложений в телефоне выполняется даже проще, но требуется изучение Java и несколько более сложного (на первый взгляд) SDK, что может стать некоторым препятствием. В данной работе рассматривается введение в разработку с использованием SDK.

3 Примеры простых приложений для ОС Android

3.1 Создание простого приложения по шагам

Для примера рассмотрим создание того же небольшого приложения для конвертации десятичных чисел в двоичное представление, которое рассмотрено в лабораторной работе для Windows Phone 8.

Установив все необходимое перечисленное в пункте 2 программное обеспечение, приступим к созданию программы.

Запустим Android Studio (рис. 1).

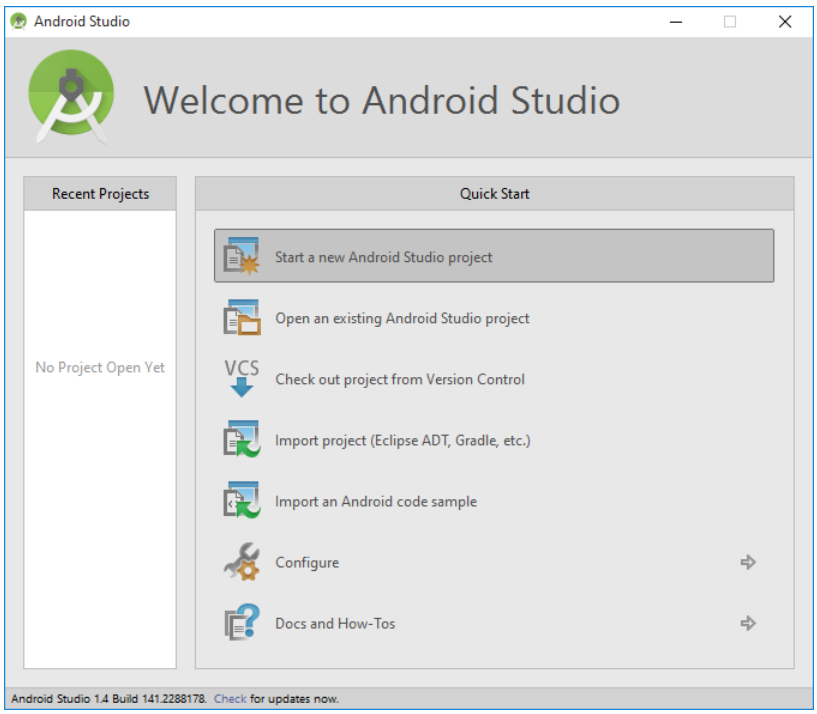

Рис. 1. Начальный экран IDE IntelliJ из Android Studio

При первом запуске IDE IntelliJ запросит действие пользователя: начать новый проект, открыть существующий проект Android Studio, получить проект из системы контроля версий, импортировать проект других средств разработки (Eclipse ADT, Grandle, и т.д.), импортировать пример кода Android, сконфигурировать среду разработки, запустить документацию и руководства. При последующих запусках может сразу открываться последний рабочий проект. Вернуться к главному меню со списком проектов можно через меню IDE IntelliJ File->Close Project.

Выберем «Начать новый проект Android Studio» (Start a new Android Studio project) или через меню File -> New -> New Project. Запустится мастер создания нового проекта (рис. 2), в котором на первом шаге необходимо указать: имя приложения (Application name), например, Dec2Bin, сайт разработчика приложения (Company Domain), например, lab2.example.com. Следующее поле – имя Java-пакета (Package Name), формируется на основе

значения поля Company Domain и должно быть уникальным для идентификации вашего приложения в случае его распространения в Google Play. По умолчанию, поле Company Domain предлагает в демонстрационных целях значение “com.example”, так как в документации рассматриваются примеры с пакетом com.example. Имейте ввиду, что это название зарезервировано в Google Play и, если вы планируете распространять ваше приложение, отредактируйте имя кнопкой Edit. Кроме того, необходимо указать директорию для размещения файлов проекта.

На следующем шаге будет предложено указать типы устройств и минимальные требования к SDK платформ, для которых создается приложение (рис. 3), проще говоря - минимальные требования к операционной системе устройств, на которых будет запускаться приложение.

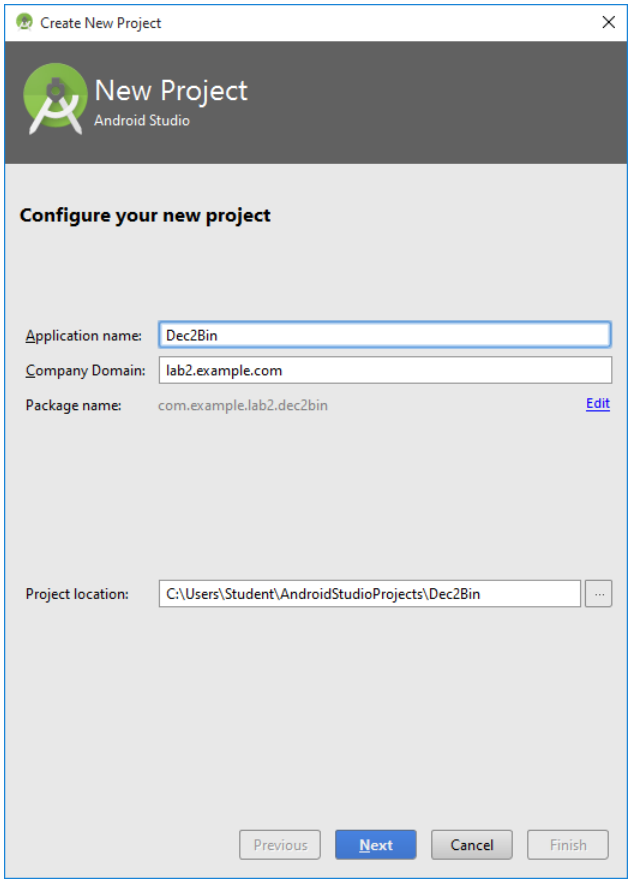

Рис. 2. Создание проекта Android Application

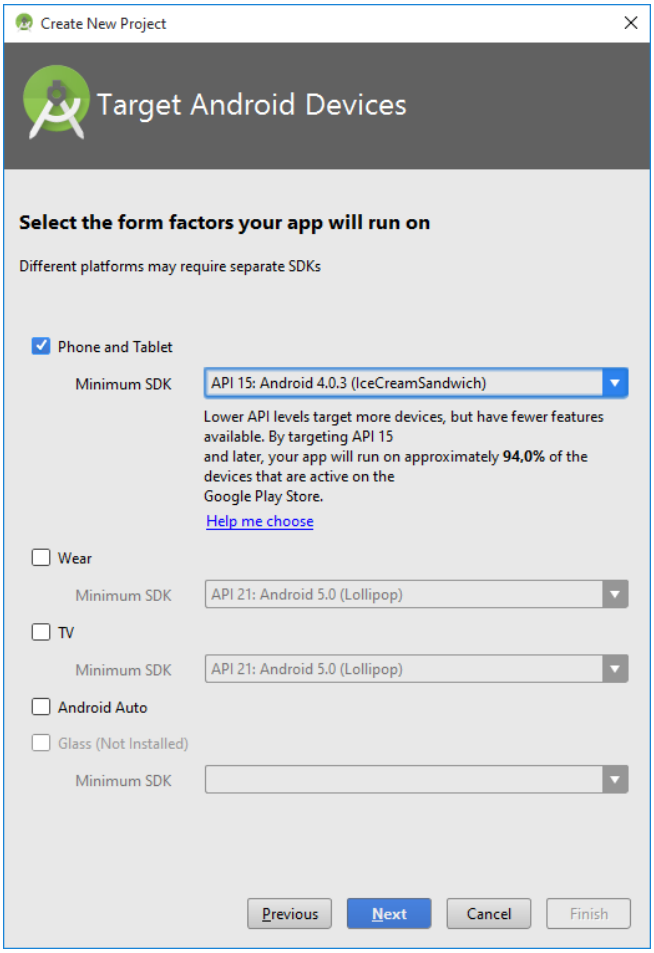

Рис. 3. Выбор требований к SDK целевых платформ

Затем будет предложено выбрать из списка (с предпросмотром приблизительного внешнего вида приложения) один из шаблонов-заготовок Activity для вашего приложения (рис. 4), можно отказаться от их использования в проекте, выбрав Add No Activity.

В настоящее время доступны следующие шаблоны:

Blank Activity,

Empty Activity,

Fullscreen Activity,

Google Maps Activity,

Google AdMod Ads Activity,

Login Activity,

Master/Detail Flow,

Navigation Drawer Activity,

Scrolling Activity,

Settings Activity,

Tabbed Activity.

Шаблон Empty Activity предназначен для обычных смартфонов и подойдет для большинства создаваемых в данной работе учебных приложений.

Шаблон Master/Detail Flow предназначен для планшетных ПК с реализацией двухпанельного режима. Шаблон Fullscreen Activity можно использовать для создания игровых приложений, когда требуется вся область экрана, в том числе занимаемая информационной панелью. Другие шаблоны нужны для создания приложений с API картографического сервиса Google Maps или сервисами Google Play.

Выбираем вариант Empty Activity и переходим к следующему шагу. Если не требуется переименование Activity и Layout, то завершаем работу мастера кнопкой Finish (рис. 5).

Если проект приложения уже существует, но Вы еще не работали с ним в данной среде (например, для рассмотрения новых примеров), Вы можете его

открыть через пункт «Open an existing Android Studio project» быстрого меню начального экрана (рис. 1).

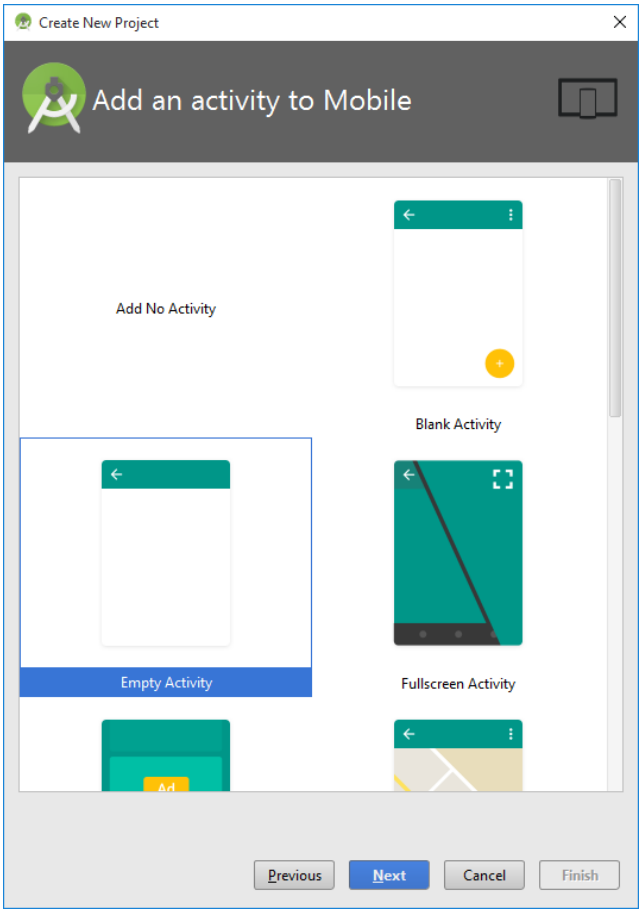

Рис. 4. Выбор шаблона приложения

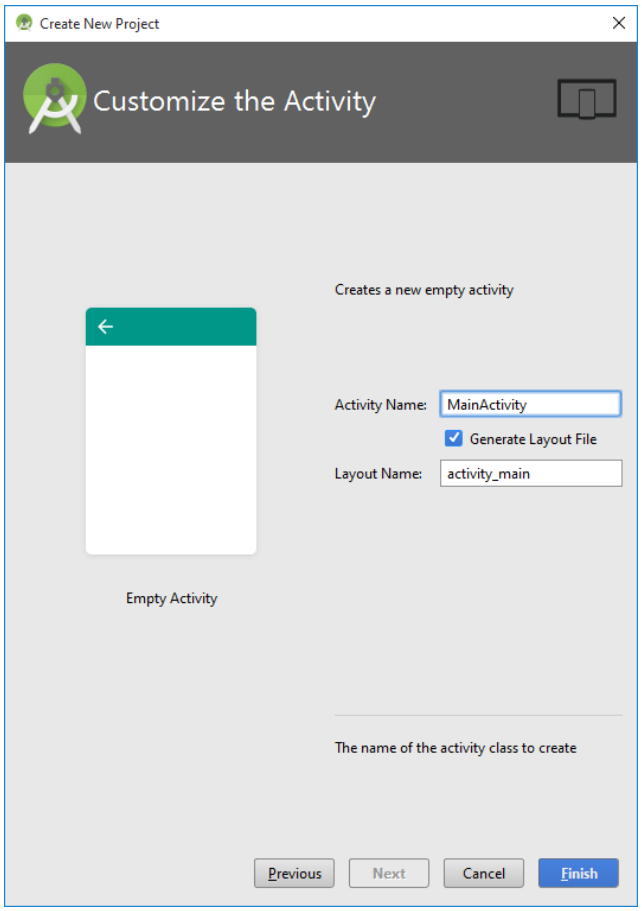

Рис. 5. Завершение работы мастера создания проекта приложения

Итак, при выборе шаблона Empty Activity создается приложение с одним окном и сообщением “Hello, world”.

Сбоку слева окна IDE имеется несколько вертикальных вкладок. Скорее всего будет активна первая вкладка 1: Project. Вкладки Structure и Captures используются гораздо реже. Если вкладка не активна, кликните на нее. Тогда слева на горизонтальной вкладке Android появится иерархический список из папок, относящихся к проекту. В некоторых случаях желательно переключиться на режим Project, который показывает истинное расположение файлов. Но на первых порах удобнее использовать именно вид Android, который скрывает служебные файлы.

Вкладка Android содержит две основные папки: **app** и **Gradle Scripts**. Папка **app** содержит все необходимые файлы приложения – код, ресурсы

(картинки и т.п.). Вторая папка служит для различных настроек, управления проектом и др.

Просмотрите содержимое папки **app**. В ней находятся три папки: **manifest**, **java**, **res**. Папка **manifest** содержит единственный файл манифеста **AndroidManifest.xml**. В этом файле должны быть объявлены все активности, службы, приёмники и контент-провайдеры приложения. Также он должен содержать требуемые приложению разрешения. Например, если приложению требуется доступ к сети, это должно быть определено здесь. «**AndroidManifest.xml**» можно рассматривать как описание для развертывания Android-приложения.

Папка **java** содержит две подпапки - рабочую и для тестов. Рабочая папка имеет название вашего пакета и содержит файлы классов. Сейчас там один класс **MainActivity**. Папка для тестов в рамках лабораторной работы не потребуется.

Папка **res** содержит файлы ресурсов, разбитых на отдельные подпапки.

Подпапка **drawable** – в этих папках хранят графические ресурсы (картинки и xml-файлы, описывающие цвет и фигуры).

Подпапка **layout** – в данной папке содержатся xml-файлы, описывающие внешний вид форм и различных элементов форм. После создания проекта там уже имеется файл **activity_main.xml**, который отвечает за внешний вид главного окна приложения.

Подпапка **mipmap** – здесь хранятся значки приложения под разные разрешения экрана.

Подпапка **values** – тут располагаются какие-либо строковые ресурсы, ресурсы цветов, тем, стилей и измерений, которые можно использовать в нашем проекте. Здесь вы можете видеть папки **colors.xml**, **dimens.xml**, **strings.xml**, **styles.xml**.

С помощью редактора разметки необходимо создать разметку, представленную на рисунках 9-10. Прежде всего, необходимо открыть сам файл разметки **activity_main.xml**, который находится в папке **\res\layout**

(рис. 6). Обратите внимание, что XML-файлы разметки можно просматривать в двух режимах: текстовом и визуальном. Для этого предназначены две вкладки в нижней части окна редактора: **Design** и **Text**. При создании разметки понадобятся элементы управления **TextView**, **EditText** и **Button**.

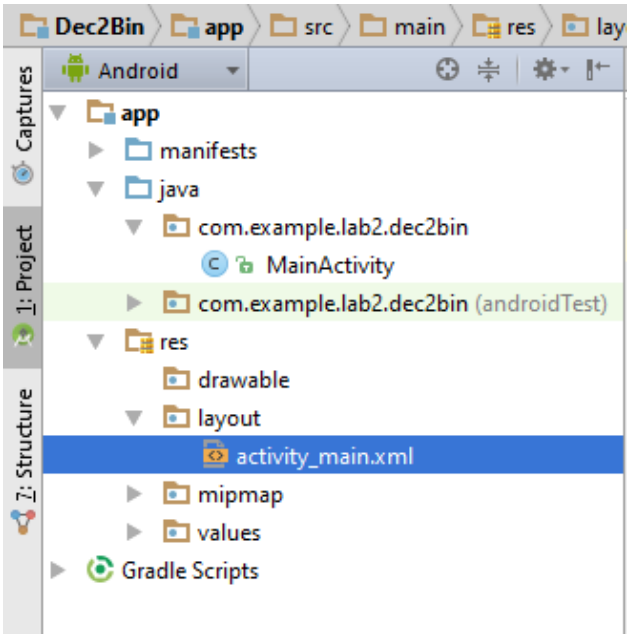

Рис. 6. Расположение файла разметки в каталогах проекта

Необходимо обратить внимание на специфику задания строковых констант — их сначала нужно создавать в xml-документе **\res\values\strings.xml** или с помощью специального диалогового окна (рис. 7 и 8) Translate Editor, а потом ссылаться на их по именам в редакторе разметки или свойствах элементов управления.

После создания разметки переходим к созданию обработчика события нажатия кнопки **btnConvert** (ее необходимо так назвать в редакторе формы / разметки). Обработчик события будет находиться в единственной активности (Activity) нашего приложения **MainActivity**, код которой находится в файле **MainActivity.java**.

Код активности на Java приведен на рисунках 11-12. На рисунке 11 приведен раздел импорта необходимых пакетов, а также заголовок и метод **OnCreate** класса активности. Все добавленные фрагменты кода обведены :

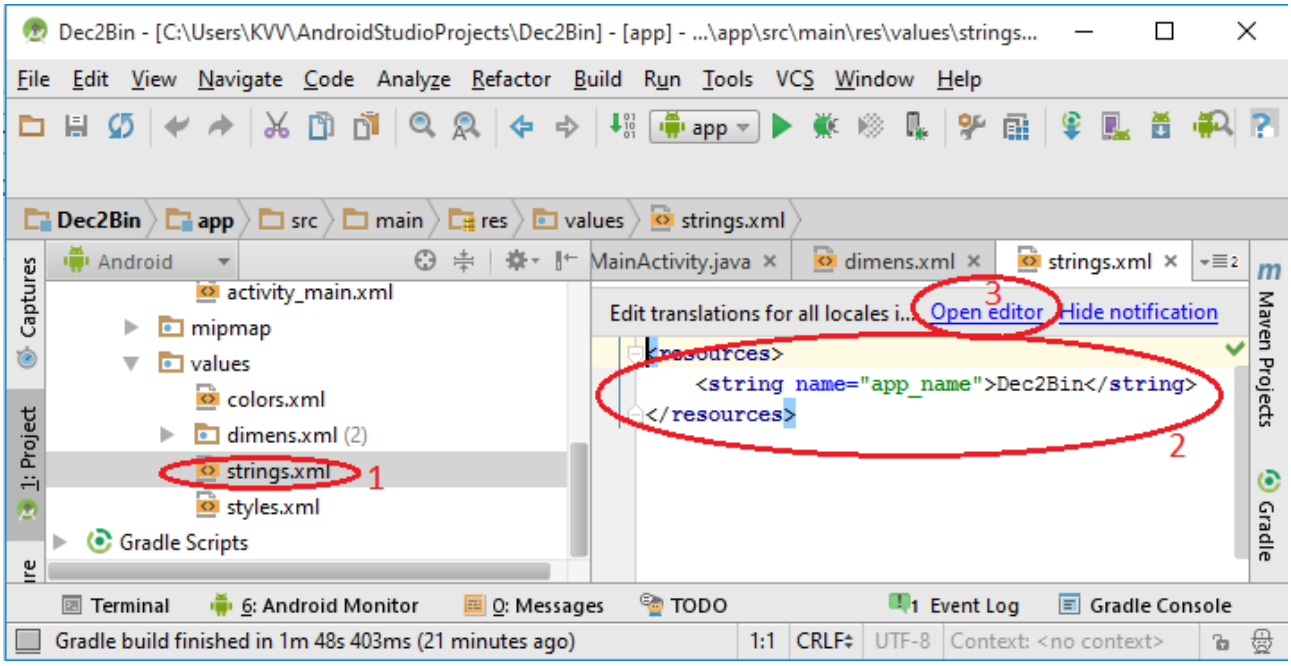

Рис. 7. Вызов редактора списка строк

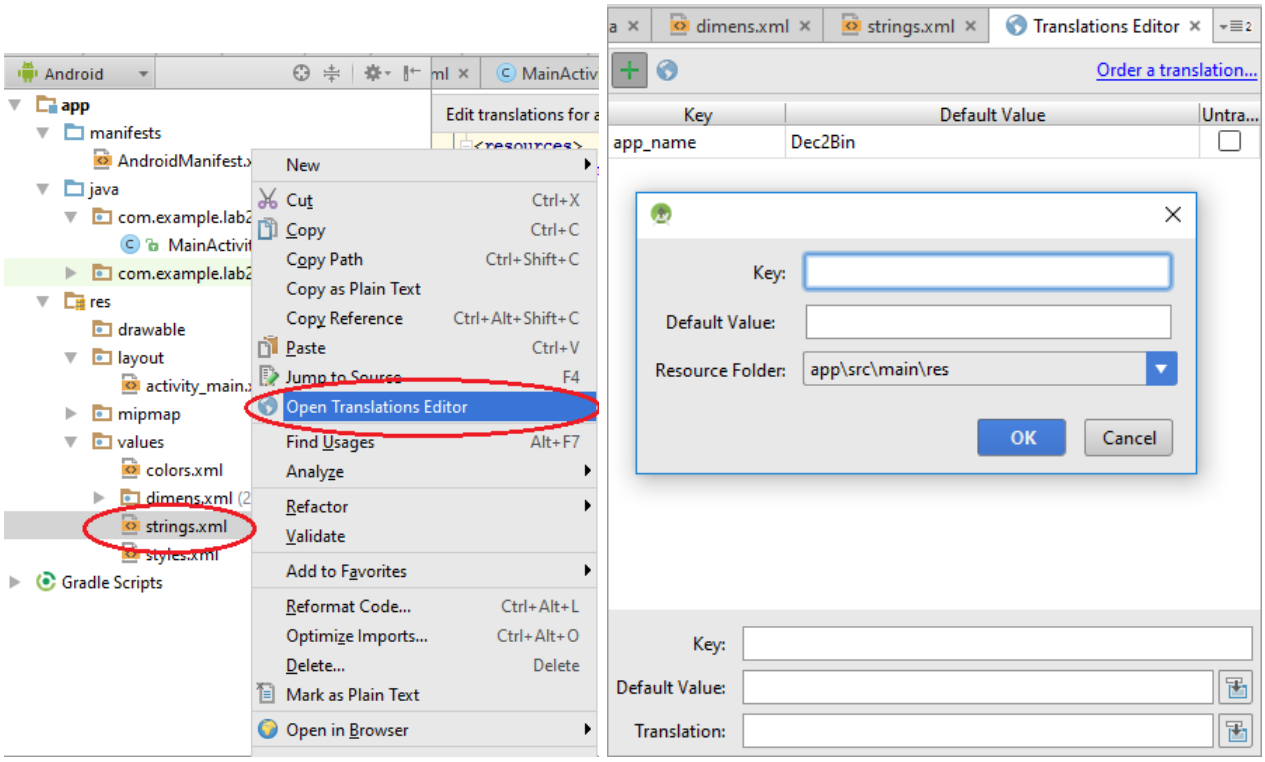

Рис. 8. Редактор списка строк (в файле strings.xml) с возможностью многоязыковой локализации

– добавлен пакет виджетов view;

– ссылка на класс (интерфейс) **OnClickListener**;

– ссылки на виджеты **Button** и **EditText**;

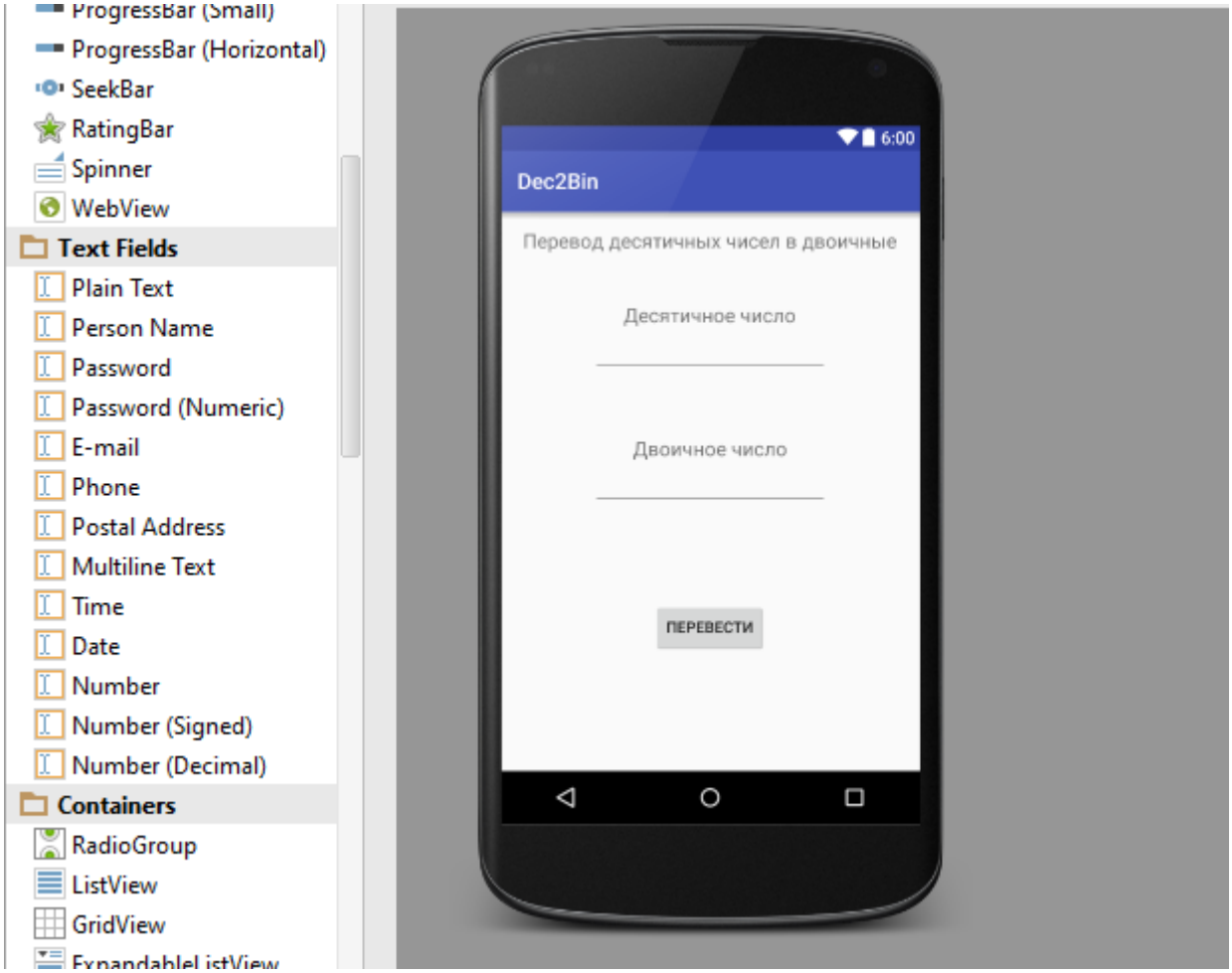

Рис. 9. Экранная форма в редакторе

activity_main.xml × MainActivity.java ×

```
<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Перевести"
    android:id="@+id/btnConvert"
    android:layout_alignParentBottom="true"
    android:layout_centerHorizontal="true"
    android:layout_marginBottom="91dp" />

<EditText
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:inputType="numberDecimal"
    android:ems="10"
    android:id="@+id/decText"
    android:layout_below="@+id/textView3"
    android:layout_centerHorizontal="true" />

<EditText
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:inputType="numberDecimal"
    android:ems="10"
    android:id="@+id/binText"
    android:layout_below="@+id/textView4"
    android:layout_centerHorizontal="true" />

</RelativeLayout>
```

Рис. 10. XML-код разметки (фрагмент)

– класс активности реализует (**implements**) интерфейс **OnClickListener**;

– добавлен как атрибут класса объект кнопки; этот объект создается путем извлечения виджета по его **id** (поле статического класса **R**, сгенерированного средой), с последующим приведением к типу **Button**;

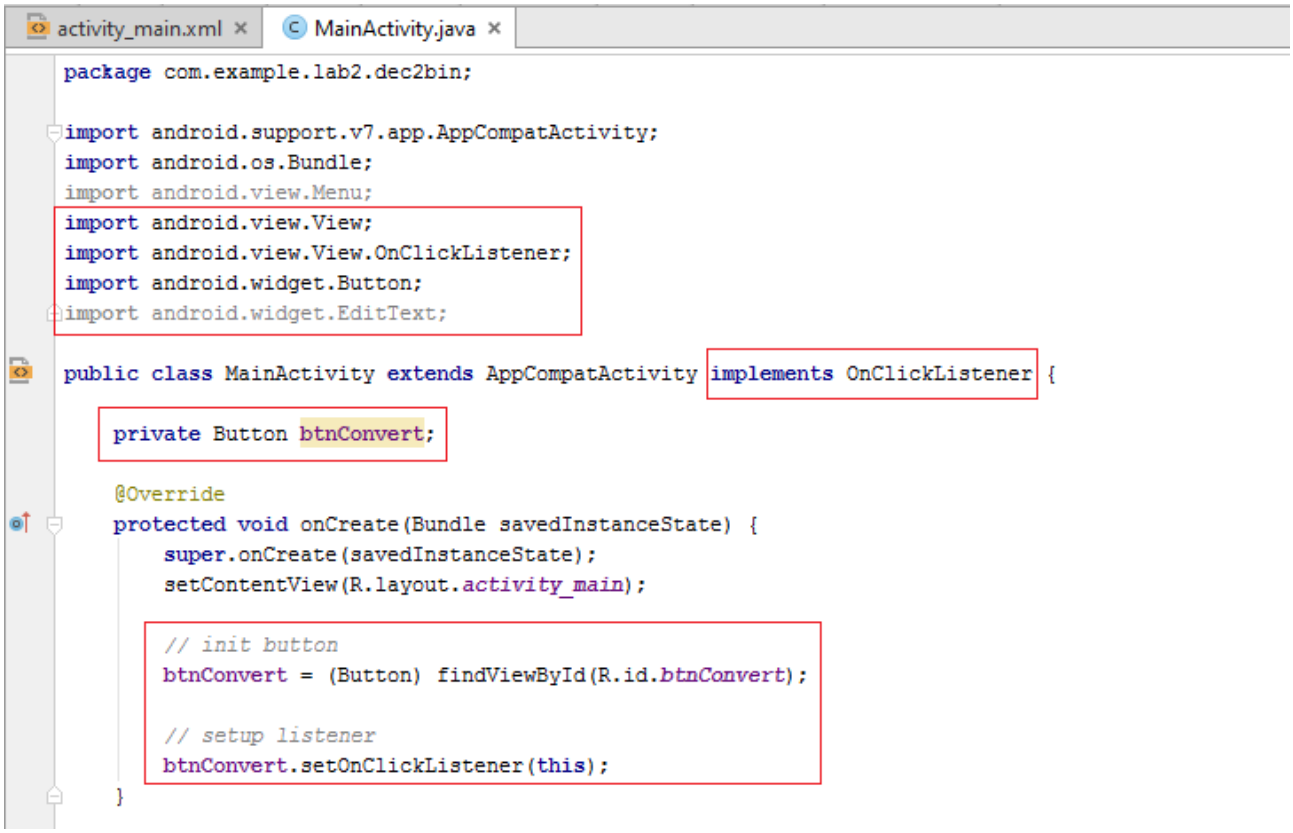

Рис. 11. Раздел импорта пакетов, заголовок и метод Create класса активности

– обработчиком нажатия кнопки устанавливается сам класс активности (поэтому они реализует интерфейс **OnClickListener**).

На рис. 12 приводится собственно код метода обработки события нажатия кнопки **onClick**, который добавлен нами в класс активности для реализации им интерфейса **OnClickListener**.

В этом методе сначала проверяется, что виджет, вызвавший это событие — это нужная нам кнопка **btnConvert**, затем по аналогии с созданием объекта кнопки создается объект **EditText** путем извлечения виджета по его **id** (поле статического класса **R**, сгенерированного средой), с последующим приведением к типу **EditText**. Затем из него извлекается

введенная строка. Во избежание ошибок при преобразовании строки в целое число сначала проверяется, что длина строки — от 1 до 8 символов, затем выполняется преобразование строки в число.

Сам алгоритм получения строки с двоичным представлением — тот же, что использован в работе по WP8, даже код практически такой же (т.к. и C#, и Java – C-подобные языки). После обработки строки по аналогии с получением объекта кнопки и текстового поля получаем объект **EditText** для вывода строки и выводим в него полученную строку с двоичным представлением.

activity_main.xml × MainActivity.java ×

```
@Override
public void onClick(View view) {
    if (btnConvert.equals(view)) {
        int dec = -1;
        EditText decText = (EditText) findViewById(R.id.decText);
        String decStr = decText.getText().toString();
        if (decStr.length() > 0 && decStr.length() < 9)
            dec = Integer.parseInt(decStr);
        if (dec >= 0) {
            String bin ="";
            while (dec != 0) {
                if (dec % 2 == 0)
                    bin = "0" + bin;
                else
                    bin = "1" + bin;
                dec = dec / 2;
            }
            EditText binText = (EditText) findViewById(R.id.binText);
            binText.setText(bin);
        }
    }
}
```

Рис. 12. Метод обработки нажатия на кнопку

В принципе, приложение готово. Оно строится автоматически после сохранения файлов. Далее необходимо запустить его в эмуляторе. Для этого сначала нужно запустить из меню Tools -> Android -> AVD Manager (Android Virtual Device Manager) и, при необходимости, в нем добавить новое виртуальное устройство (Android Virtual Device – AVD) или изменить параметры текущего – рис. 13. Например, можно создать виртуальные

устройства с различными версиями Android, размером и разрешением экрана, архитектурой CPU, объемом оперативной памяти и хранилища, наличием клавиатуры, и другими (рис. 14).

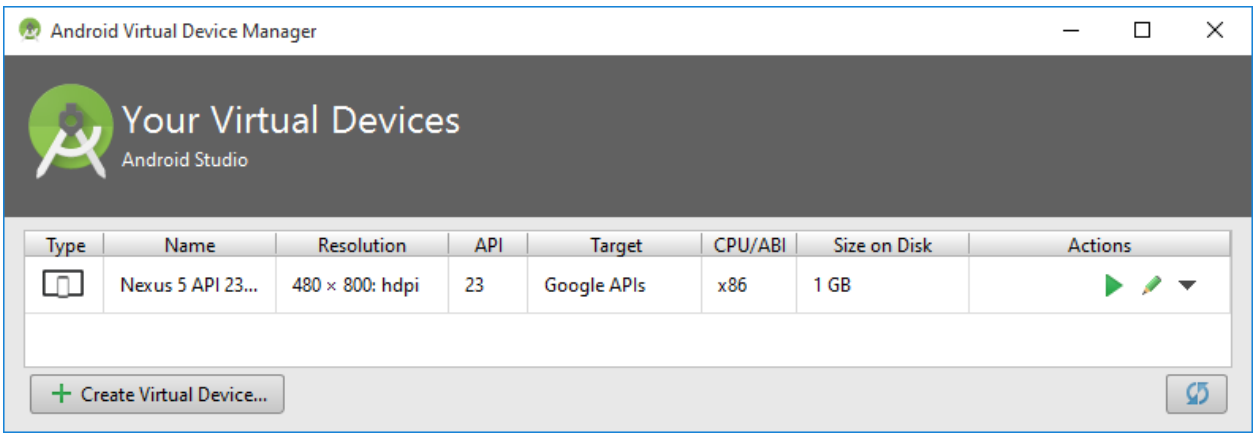

Рис. 13. AVD Manager

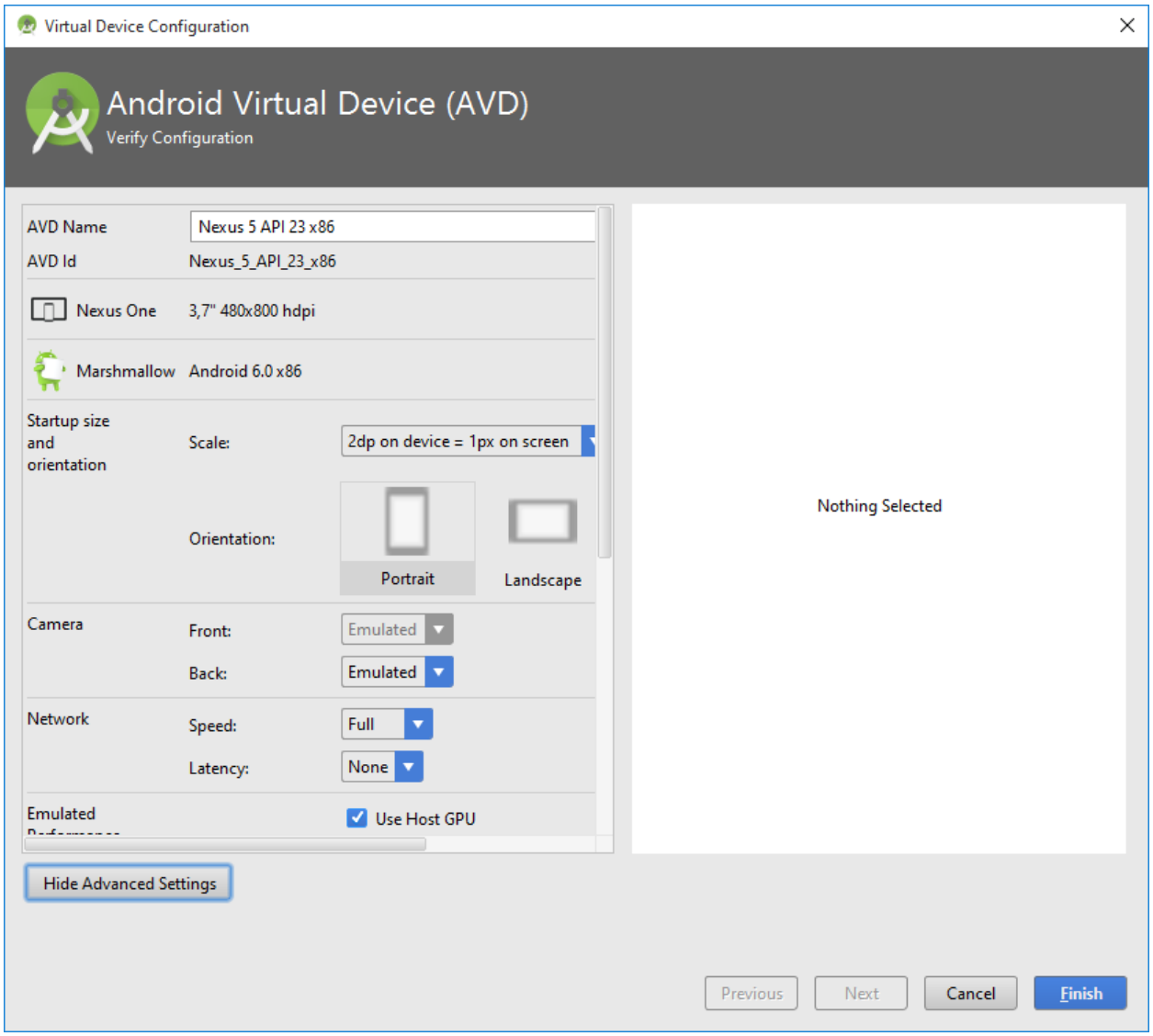

Рис. 14. Добавление/редактирование виртуального устройства

После создания устройства можно запустить программу на эмуляторе из меню Run, горячей клавишей Shift-F10, нажав на зеленый треугольник в панели инструментов, либо — из контекстного меню проекта.

Вид приложения в окне эмулятора приведен на рис. 15 (вид может отличаться в зависимости от выбранных Вами настроек AVD, для ускорения работы эмулятора не стоит выбирать большой размер и разрешение экрана устройства).

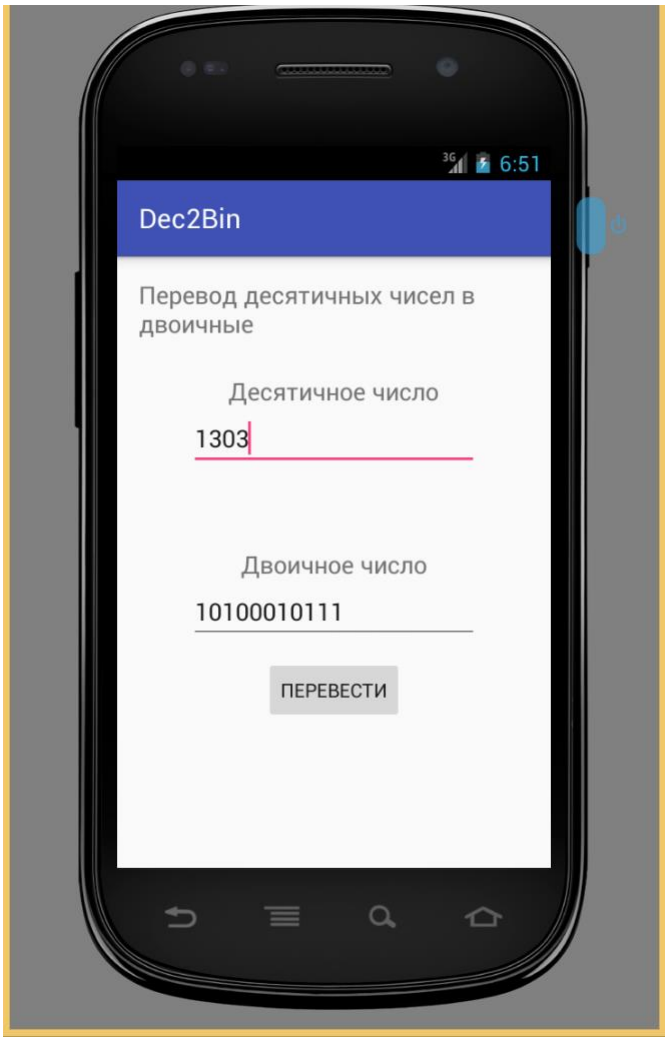

Рис. 15. Приложение, запущенное в эмуляторе

3.2 Дополнительные примеры приложений

В рамках данной работы рассматриваются еще три примера простых приложений, которые дают представление о работе с экраном и кнопками, о

работе с анимацией и с GPS. Примеры рассмотрены в лабораторных работах курса А. Семаковой (СПбГУ-Intel) [6.6, 6.7].

3.3 Развертывание приложений на смартфоне

Несмотря на наличие эмулятора, конечно, интересно понаблюдать за работой программы непосредственно на смартфоне, для которого она и создается. Никаких регистраций, как в случае с WP8, не требуется. Можно подключить ваш Android-телефон к компьютеру, переслать туда пакет Dec2Bin.apk и файл AndroidManifest.xml, затем обратиться к apk – файлу через Проводник (Explorer), подтвердить установку (для этого должна быть включена установка приложений не из Маркета) и затем открыть приложение.

4 Порядок выполнения работы

4.1 Изучить пункт 3.1, рассмотреть пошаговое создание примера.

4.2 Выполнить создание примера по пункту 3.1

4.3 Выполнить небольшую модификацию примера из п. 3.1, например, предусмотреть обработку исключений при неверном вводе данных с выдачей информации об ошибке (отключить ввод только чисел)

4.4 Изучить пример из п. 3.2

4.5 При наличии телефона с Android развернуть на нем приложение из п. 3.1 или 3.2

4.6 Выполнить модификацию второго примера из п. 3.2.

4.7 Выполнить индивидуальные задания из п. 5

5 Варианты индивидуальных заданий

5.1 Дополнить пример 3.1 обратным преобразованием (из двоичного – в десятичный код).

5.2 Дополнить пример 3.1 преобразованием в шестнадцатеричный код и обратно.

5.3 Рассмотреть пример работы с кнопками и выполнить свои модификации для него.

5.4 Модифицировать пример 3.1 для учета поворота экрана.

5.5 Модифицировать пример 3.2 для учета поворота экрана.

5.6 Рассмотреть пример работы с анимацией и выполнить свои модификации для него.

5.7 Рассмотреть пример работы с GPS и выполнить свои модификации для него.

5.8 Написать небольшое приложение для просмотра фото со смартфона.

5.9 Написать небольшое приложение для записи и воспроизведения записанных фрагментов (диктофон)

5.10 Написать небольшое приложение для просмотра контактов из телефонной книги смартфона

5.11 Написать небольшое приложение с использованием Grid.

5.12 Написать небольшое приложение для работы с TouchScreen (масштабирование текста щипком).

6 Использованные и рекомендуемые источники

6.1 Download Android Studio and SDK Tools [Электронный ресурс] Режим доступа : https://developer.android.com/sdk/index.html

6.2 Intel® XDK | Intel® Developer Zone [Электронный ресурс] Режим доступа: https://software.intel.com/ru-ru/intel-xdk

6.3 12 лучших фреймворков для разработки приложений под Android [Электронный ресурс] Режим доступа : http://habrahabr.ru/post/265261/

6.4 Intel® HAXM Download [Электронный ресурс] Режим доступа : https://software.intel.com/sites/default/files/haxm-windows_r05.zip

6.5 Android NDK [Электронный ресурс] Режим доступа: https://developer.android.com/tools/sdk/ndk/index.html

6.6 Материалы Робошколы-2012 [Электронный ресурс] Режим доступа: http://roboschool.org/materials/roboschool2012/Android/05%20Practice-Android.pdf

6.7 А. Семакова НОУ ИНТУИТ | Введение в разработку приложений для смартфонов на ОС Android [Электронный ресурс] Режим доступа: http://www.intuit.ru/studies/courses/4462/988/info

6.8 А. Климов Освой Android играючи [Электронный ресурс] Режим доступа: http://developer.alexanderklimov.ru/android/

7 Контрольные вопросы

7.1 Особенности ОС Android, ее место на рынке мобильных ОС и перспективы

7.2 Перечислите необходимое программное обеспечение для разработки под Android.

7.3 Является ли описанный способ создания приложений для Android единственно возможным ?

7.4 Какие основные шаблоны приложений для Android предлагает Android Studio ?

7.5 Для чего служит и как используется при разработке эмулятор Android?

7.6 Особенности разработки интерфейса пользователя для Android приложений.

7.7 Что необходимо для развертывания приложения на самом телефоне ?

7.8 Какие функции для работы с телефоном Вы использовали в данной работе ?

7.9 Какие отличия от процесса разработки под WP8 Вы можете выделить ? Есть ли что-то общее в процессе разработки ?

У ч е б н о е и з д а н и е

Андрей Евгеньевич Андреев
Виталий Владимирович Камнев

Создание приложений для мобильной операционной системы Android с использованием Android SDK

Методические указания к
лабораторным работам по дисциплине
«Мобильные и встраиваемые операционные системы»

Темплан заказных изданий 2015 г. Поз. №_____
Подписано в печать _____ 2015. Формат 60x84 1/16. Бумага офсетная
Гарнитура Times. Печать офсетная. Усл. печ. л. ____

Волгоградский государственный технический университет
400005, Волгоград, пр. им. В.И. Ленина 28, корп. 1

Отпечатано в типографии УИНЛ ВолгГТУ
400005, Волгоград, пр. им. В.И. Ленина 28, корп. 7