

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ	4
ЦЕЛЬ И ЗАДАЧИ РАБОТЫ, ТРЕБОВАНИЯ К РЕЗУЛЬТАТАМ ЕЕ ВЫПОЛНЕНИЯ.....	5
КРАТКАЯ ХАРАКТЕРИСТИКА ОБЪЕКТА ИЗУЧЕНИЯ, ИССЛЕДОВАНИЯ	6
УСТАНОВКА И НАСТРОЙКА ANDROID STUDIO	9
СОЗДАНИЕ ПЕРВОГО ПРИЛОЖЕНИЯ.....	15
ЗАДАНИЕ НА ЛАБОРАТОРНУЮ РАБОТУ	25
КОНТРОЛЬНЫЕ ВОПРОСЫ И ЗАДАНИЯ	31
ФОРМА ОТЧЕТА ПО ЛАБОРАТОРНОЙ РАБОТЕ	31
ОСНОВНАЯ ЛИТЕРАТУРА.....	32
ДОПОЛНИТЕЛЬНАЯ ЛИТЕРАТУРА	32

ВВЕДЕНИЕ

Настоящие методические указания составлены в соответствии с программой проведения лабораторных работ по курсу «Разработка мобильного программного обеспечения» на кафедре «Программное обеспечение ЭВМ, информационные технологии и прикладная математика» факультета фундаментальных наук Калужского филиала МГТУ им. Н.Э. Баумана.

Методические указания, ориентированные на студентов 3-го курса направления подготовки 09.03.04 «Программная инженерия», содержат краткое описание среды Android Studio, руководство по настройке среды, характеристика структуры Android-проекта, руководство по запуску приложения с заданными настройками и задание на выполнение лабораторной работы.

Для выполнения лабораторной работы студенту необходимы минимальные знания по программированию на высокоуровневом языке программирования Kotlin.

Требования к программному обеспечению:

- ОС Windows 7/8/10 / Ubuntu 16.04 (или старше)
- Android Studio

ЦЕЛЬ И ЗАДАЧИ РАБОТЫ, ТРЕБОВАНИЯ К РЕЗУЛЬТАТАМ ЕЕ ВЫПОЛНЕНИЯ

Целью выполнения лабораторной работы является приобретение практических навыков по настройке и запуску Android-приложений.

Основными задачами выполнения лабораторной работы являются:

1. Ознакомиться с инструментами разработки Android-приложений.
2. Получить навыки настройки среды Android Studio.
3. Изучить структуру Android-проекта.
4. Научиться запускать приложение с заданными настройками.

Результатами работы являются:

- Настроенная для разработки среда Android Studio
- Простейшее android-приложение, выполненное согласно варианту задания
- Подготовленный отчет

КРАТКАЯ ХАРАКТЕРИСТИКА ОБЪЕКТА ИЗУЧЕНИЯ, ИССЛЕДОВАНИЯ

Android – открытая операционная система для мобильных телефонов, смартфонов, коммуникаторов, планшетных компьютеров, электронных книг, цифровых проигрывателей, наручных часов, нетбуков и смартбуков, основанная на ядре Linux и поддерживающая различные аппаратные платформы, такие как ARM, MIPS, POWER, x86.

Android-устройства набирают популярность быстрее, чем любая другая мобильная платформа, что делает их отличным выбором для первого знакомства с разработкой мобильных приложений, особенно для Java-программистов. Навыки в разработке android-приложений могут принести много пользы своему владельцу. Например, можно разрабатывать для себя полезные, интересные, занимательные приложения, а можно, разведав обстановку и осмотревшись, сделать разработку мобильных приложений своей профессиональной деятельностью, основной или дополнительной.

Изначально ОС Android разрабатывалась компанией Android Inc., которую затем (июль, 2005) купила Google. Впоследствии (ноябрь, 2007) Google инициировала создание бизнес-альянса Open Handset Alliance, который и занимается сейчас поддержкой и дальнейшим развитием платформы.

С момента выхода первой версии (сентябрь, 2008) произошло несколько обновлений системы. Эти обновления, как правило, касаются исправления обнаруженных ошибок и добавления нового функционала в систему. Каждая версия системы получает собственное кодовое имя на тему десерта.

Android поддерживает фоновое выполнение задач; предоставляет богатую библиотеку элементов пользовательского интерфейса; поддерживает 2D и 3D графику, используя OpenGL стандарт; поддерживает доступ к файловой системе и встроенной базе данных SQLite.

С точки зрения архитектуры, система Android представляет собой полный программный стек, в котором можно выделить следующие уровни:

- **Базовый уровень (Linux Kernel)** уровень абстракции между аппаратным уровнем и программным стеком;
- **Набор библиотек и среда исполнения (Libraries & Android Runtime)** обеспечивает важнейший базовый функционал для приложений, содержит виртуальную машину Dalvik и базовые библиотеки Java необходимые для запуска Android приложений;
- **Уровень каркаса приложений (Application Framework)** обеспечивает разработчикам доступ к API, предоставляемым компонентами системы уровня библиотек;
- **Уровень приложений (Applications)** набор предустановленных базовых приложений.

Разработчик обычно взаимодействует с двумя верхними уровнями архитектуры Android для создания новых приложений. Библиотеки, система исполнения и ядро Linux скрыты за каркасом приложений.

Инструментарий для разработки Android-приложений

Перед тем, как приступить к созданию Android-приложений, необходимо установить подходящий инструментарий.

Как правило, разработка Android-приложений осуществляется на языке Java. Поэтому, в первую очередь, необходимо установить Java Development Kit (JDK).

Java SDK (JDK) – это бесплатно распространяемый комплект разработчика приложений на языке Java, включающий в себя компилятор Java, стандартные библиотеки классов Java, примеры, документацию, различные утилиты и исполнительную систему Java Runtime Environment (JRE). В состав JDK не входит интегрированная среда разработки (Integrated Development Environment). Поэтому после того, как будет установлен JDK, следует установить IDE.

В качестве среды разработки будем использовать [Android Studio](#).

Android Studio – интегрированная среда разработки для Android-программирования, созданная на базе популярной среды IntelliJ IDEA.

Установка Android Studio включает в себя:

- Android SDK - последняя версия Android SDK.
- Инструменты и платформенные средства Android SDK - средства отладки и тестирования приложений.
- Образ системы для [эмулятора Android](#) - позволяет создавать и тестировать приложения на различных виртуальных устройствах.

УСТАНОВКА И НАСТРОЙКА ANDROID STUDIO

Скачать установочный пакет для студии можно со страницы <https://developer.android.com/studio/index.html>

Сама установка примитивна - обычный диалог инсталлятора. Установочный пакет уже включает в себя необходимый минимум.

После загрузки установочного exe-файла запускаем его. Жмем Next, до появления следующего окна:

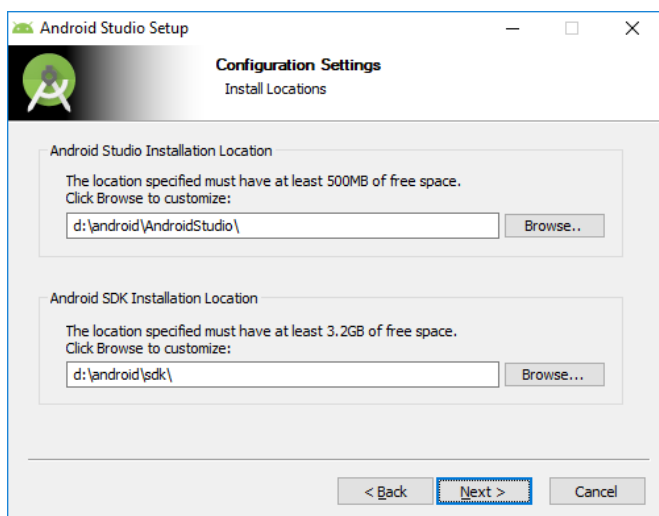


Рис.1. Указание установочного пути

Здесь, установщик спрашивает, в какую директорию ставить студию, и в какую ставить [SDK](#). Если с самой студией все понятно, то с SDK нужно быть внимательным. Как опять же следует из скриншота, для установки SDK нужно минимум 3.2 GB места на диске. Это минимум, на самом деле, места нужно больше, поскольку через какое-то время вам нужно будет докачивать обновленный SDK. так что, если вы не уверены, что места в будущем хватит — лучше измените местоположение на более вместительный диск.

После этого понадобится стандартно несколько раз нажать на кнопку «Next», и на этом установка Android Studio завершена.

При первом запуске Android Studio задаст вам стандартный вопрос об импорте конфигурации:

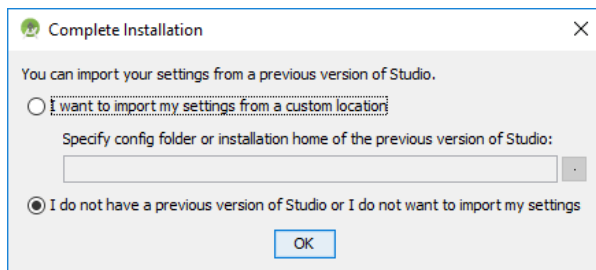


Рис.2. Запуск Android Studio. Окно импорта конфигурации

Поскольку пока у нас нет старых настроек, то оставляем выбранным нижний пункт и жмем «Ок».

Далее появится визард установки (рисунок 3).

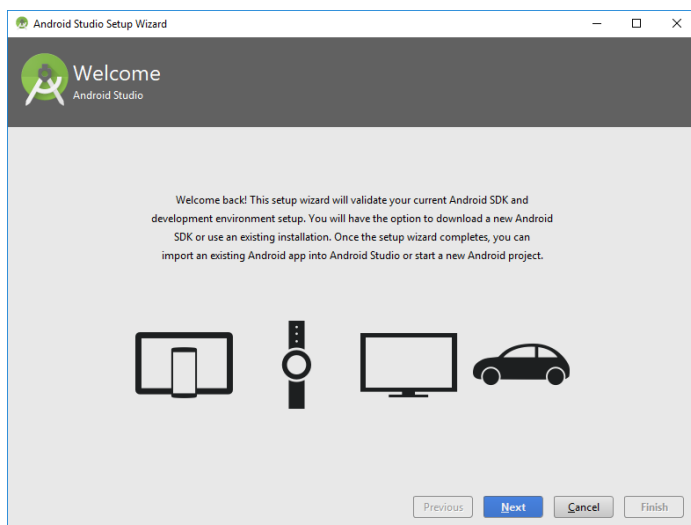


Рис.3. Запуск Android Studio. Окно приветствия

Жмем «Next».

В следующем окне выбираем стандартную установку и жмем «Next» (рисунок 4).

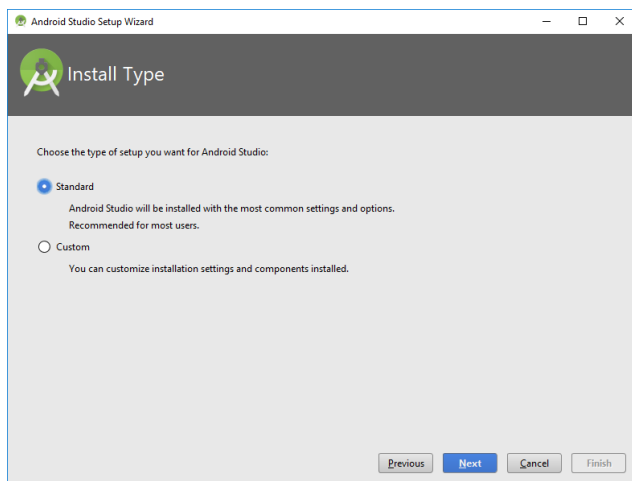


Рис.4. Выбор типа установки

Визард сообщит, что ему необходимо загрузить несколько компонентов для завершения установки, после чего начнется процесс загрузки и распаковки (рисунок 5).

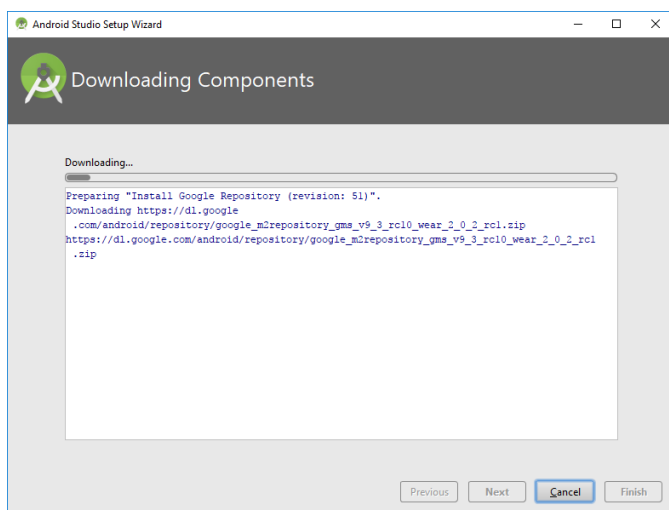


Рис.5. Процесс загрузки

По окончании жмем Finish. Появится стартовое окно (рисунок 6).

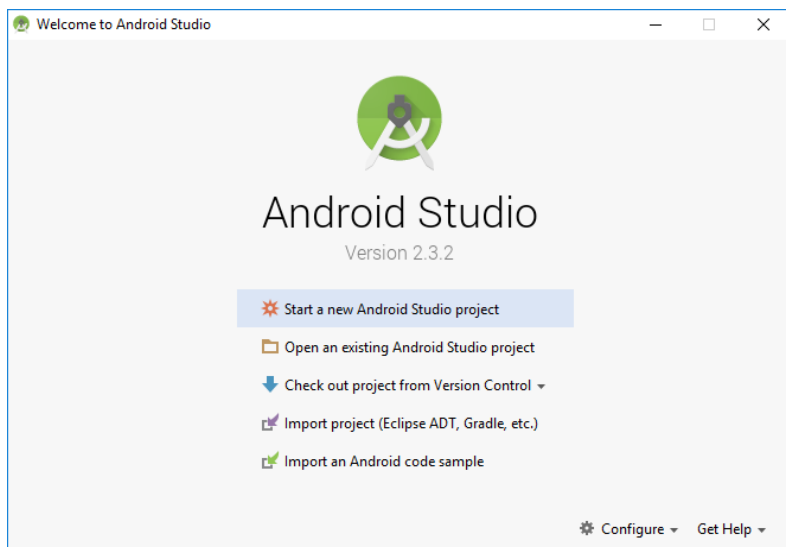


Рис.6. Стартовое окно

После выполнения этих шагов первоначальная настройка среды завершена. Однако для удобства разработки рекомендуется произвести также следующие настройки.

Перейдите в настройки (Configure | Settings) и поставьте галочки напротив «show line numbers» и «show method separators» (рисунок 6). Первая будет отображать номера строк слева от текста, вторая — будет рисовать разделители между методами в коде.

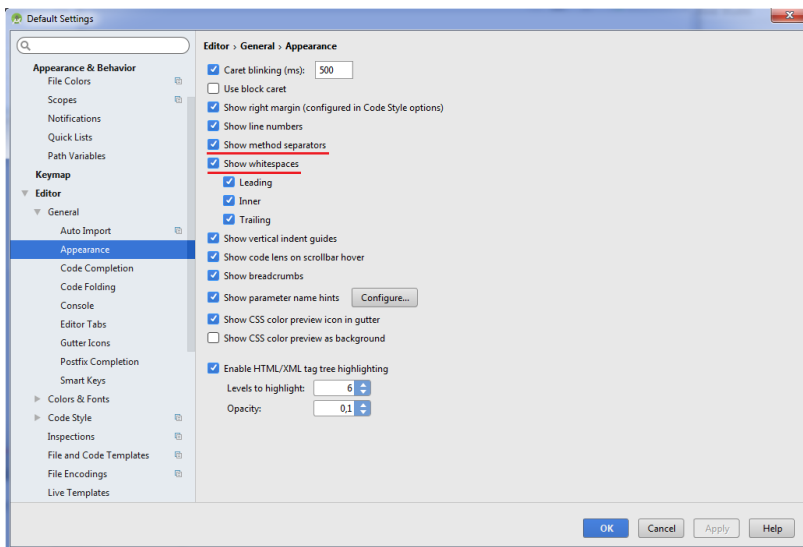


Рис.6. Окно настроек

Иногда вам необходимо будет запускать Android SDK Manager и проверять наличие новых версий SDK (Configure | SDK Manager) (рисунок 7).

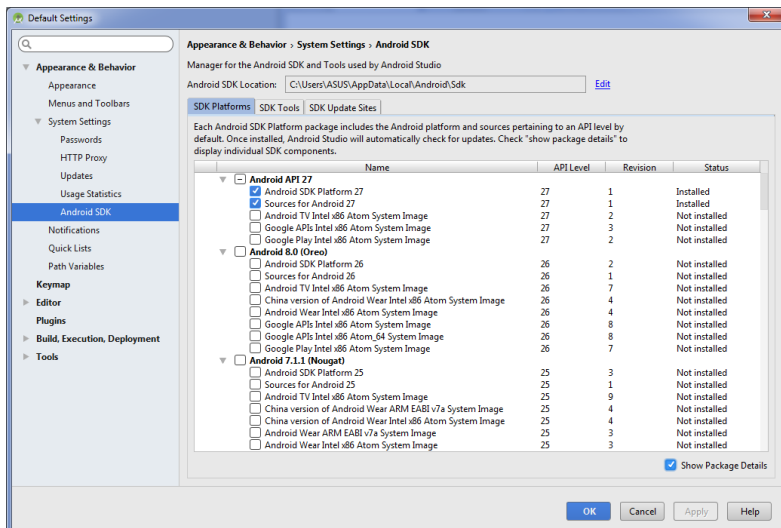


Рис.7. SDK Manager.

Здесь мы детально можем посмотреть, какие пакеты для каждой платформы установлены. Все компоненты сгруппированы по определенной версии SDK, например, Android API 27, Android 8.0 (Oreo), Android 7.1.1 (Nougat) и так далее. Каждая версия SDK фактически представляет определенную версию или подверсию ОС Android. Но каждая версия SDK предполагает широкий круг компонентов, это в том числе и инструменты для разработки под телевизоры, под смарт-часы и т.д. Не все эти инструменты могут понадобиться, поэтому нет смысла все версии SDK абсолютно полностью устанавливать.

В данном случае нас прежде всего будет интересовать пункт Android SDK Platform. Android SDK Platform содержит весь основной функционал, который используется при разработке.

Кроме SDK Platform каждая платформа, как правило, содержит еще ряд компонентов, которые предназначены для тестирования приложения на эмуляторе.

СОЗДАНИЕ ПЕРВОГО ПРИЛОЖЕНИЯ

Для запуска и тестирования приложений, нам понадобится Android Virtual Device (AVD). Это эмулятор Android-смартфона, на который мы сможем устанавливать созданные нами приложения, и запускать их там.

Создание проекта

Для начала создадим проект. В стартовом окне выбираем пункт «Start a new Android Studio project». Появится диалоговое окно создания проекта.

Необходимо поставить галочку «**Include Kotlin Support**»

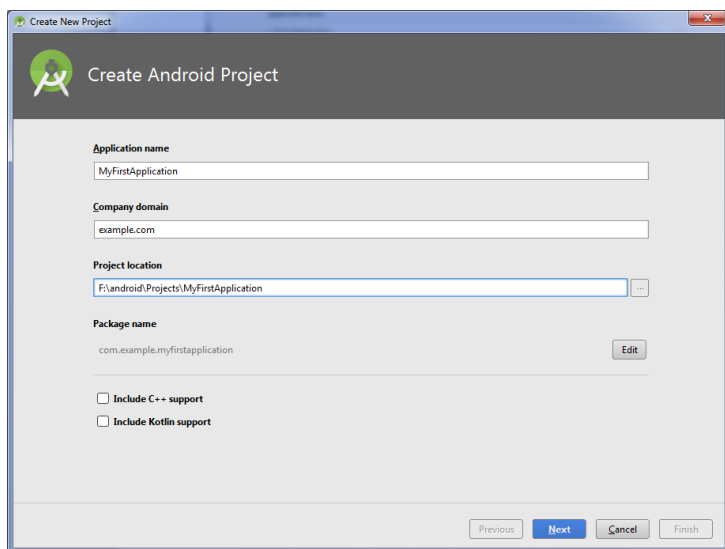


Рис.8. Окно создания проекта

- **Application Name** – здесь вводится название приложения.
- **Company Domain** – здесь указывается домен приложения или тот пакет классов, где будет размещаться главный класс приложения. Лучше также сразу задать какое-нибудь свое значение в поле Company Domain. Дело в том, что если вы будете размещать приложение в магазине Google Play, то значение для этого поля

должно быть уникальным для всего магазина. Хотя для тестовых проектов, как в данном случае, можно оставить в этом поле значение по умолчанию.

— **Project Location** – расположение файлов проекта на жестком диске.

На следующем шаге будет предложено установить минимальную поддерживаемую версию проекта (рисунок 9). По умолчанию устанавливается версия Android 4.1, что покрывает более 95% устройств Android.

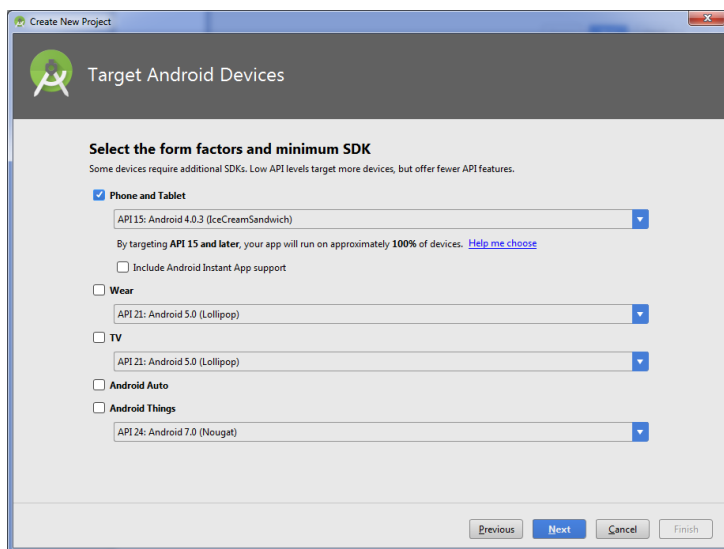


Рис.9. Выбор минимальной версии проекта

На следующем шаге необходимо выбрать шаблон проекта (рисунок 10).

Android Studio предоставляет ряд шаблонов для различных ситуаций, но самыми распространенными являются Basic Activity и Empty Activity. Это самые удобные шаблоны для старта для создания большинства приложений. В данном случае мы выберем шаблон Empty Activity.

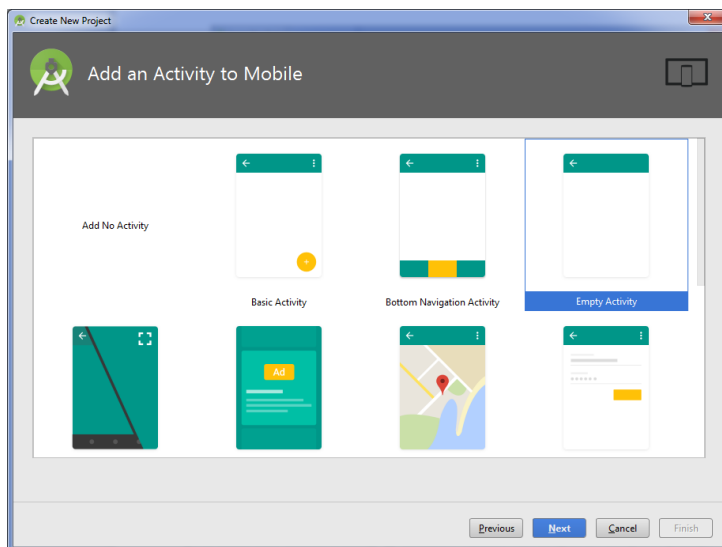


Рис.10. Выбор шаблона проекта

Далее будет необходимо установить ряд настроек проекта:

- **Activity Name** название главного класса приложения.
- **Layout Name** название файла xml, в котором будет храниться определение визуального интерфейса.
- **Generate Layout File** надо ли генерировать файл xml с определением визуального интерфейса.
- **Backwards Compatibility (AppCompat)** в отмеченном состоянии позволяет установить обратную зависимость между различными версиями Android.

Оставим все настройки по умолчанию и нажмем на кнопку Finish. И через некоторое время Android Studio создаст и откроет проект (рисунок 11).

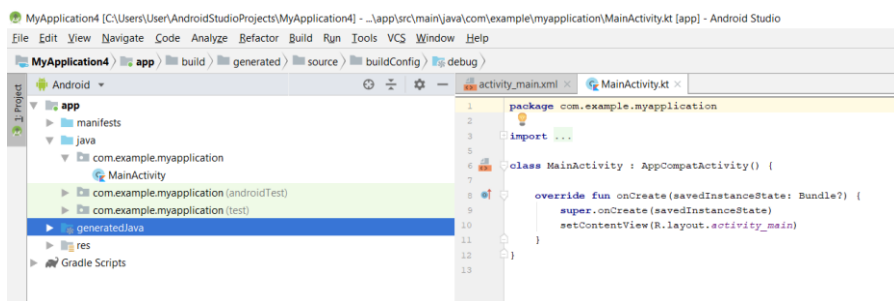


Рис.11. Созданный проект.

Структура проекта

Проект Android может состоять из различных модулей. По умолчанию, когда мы создаем проект, создается один модуль - **app**. Модуль имеет три подпапки:

- **manifests** хранит файл манифеста `AndroidManifest.xml`, который определяет конфигурацию
- **java** хранит файлы кода на языке `java`, которые структурированы по отдельным пакетам
- **res** содержит используемые в приложении ресурсы
- **Gradle Scripts** содержит ряд скриптов `gradle` (как для модуля `app` или других возможных модулей, так и для всего проекта), которые используются при построении приложения

Рассмотрим полную структуру проекта приложения под ОС Android, которая создается по умолчанию. Для этого перейдем к полной структуре проекта. Сделать это можно двойным щелчком по названию проекта в левом верхнем углу экрана. Здесь мы увидим единственный модуль проекта - модуль `app`. Собственно весь код, с которым мы будем работать, располагается внутри этого модуля.

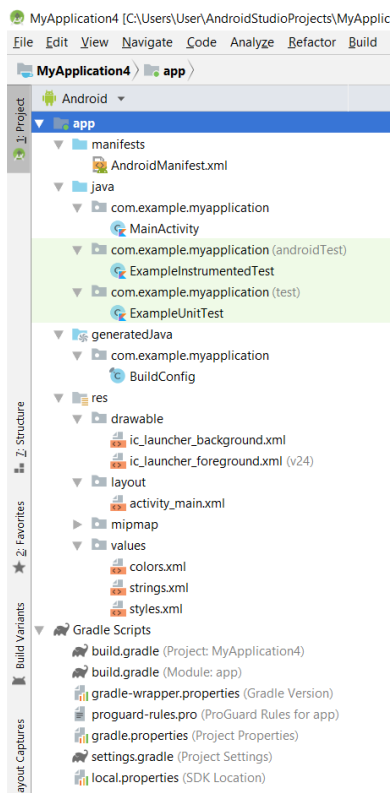


Рис.12. Полная структура проекта

Все модули в проекте описываются файлом `setting.gradle`. По умолчанию он имеет следующее содержимое:

```
include ':app'
```

Файл `build.gradle` содержит информацию, которая используется при построении проекта.

Каждый модуль имеет свой файл `build.gradle`, который определяет конфигурацию построения проекта, специфичную для данного модуля. Так, если мы посмотрим на содержимое папки `app`, то как раз найдем в ней такой файл. На начальном этапе данные файлы не столь важны, достаточно лишь понимать, для чего они нужны.

В модуле `app` мы можем увидеть несколько папок и файлов, из которых для нас важнейшими являются:

- каталог **libs** - предназначен для хранения библиотек, используемых приложением

- каталог **src** - предназначен для хранения исходного кода. Он содержит ряд подкаталогов. Каталоги **androidTest** и **test** предназначены для хранения файлов тестов приложения. А собственно исходные коды располагаются в папке **main**

Папка `main` имеет сложную структуру:

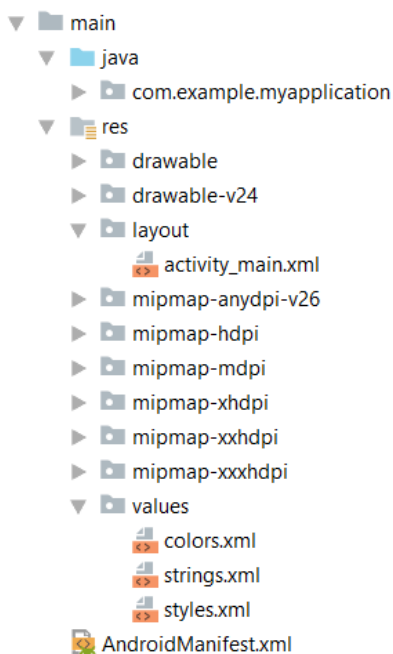


Рис.13. Структура папки `main`

1. **AndroidManifest.xml** представляет файл манифеста, который описывает фундаментальные характеристики приложения, его конфигурацию и определяет каждый из компонентов данного приложения.

2. Папка **java** содержит исходные файлы приложения. По умолчанию в нем находится файл класса MainActivity, который запускается по умолчанию при старте приложения.
3. Папка **res** содержит каталоги с ресурсами, в частности она содержит следующие каталоги:
 - папка **drawable** предназначена для хранения изображений, используемых в приложении;
 - папка **layout** предназначена для хранения файлов, определяющих графический интерфейс. По умолчанию здесь есть файл activity_main.xml, который определяют интерфейс для единственной в проекте activity – MainActivity;
 - папки **mipmap-xxxx** содержат файлы изображений, которые предназначены для создания иконки приложения при различных разрешениях экрана. Соответственно для каждого вида разрешения здесь имеется свой каталог;
 - папка **values** хранит различные xml-файлы, содержащие коллекции ресурсов - различных данных, которые применяются в приложении.

Стоит заметить, что один модуль в проекте это тривиальный случай.

При создании проекта, в нем создается модуль. В этом модуле мы рисуем экраны приложения и пишем код. И при запуске этого модуля мы получаем готовое приложение. Поэтому модуль по сути и является приложением. А проект - контейнер для модуля. Есть проект, и в нем есть модуль. При запуске проекта запускается модуль и мы получаем Android-приложение, которое создано в этом модуле. Но в одном проекте может быть несколько модулей. Да и проектов можно создать несколько. В этом случае при запуске какого-либо проекта необходимо будет указать, какой именно модуль вы хотите запустить. И каждый модуль является отдельным Android-приложением.

Стандартный проект, уже содержит некоторый функционал. Это функционал представляет собой простейшее приложение с выводом на экран одной строки "Hello world!".

Открыв файл activity_main.xml вы увидите содержание интерфейса приложения. Обратите внимание, что XML-файлы можно просматривать в двух режимах: текстовом и визуальном. Для этого

предназначены две вкладки в нижней части окна редактора: Design и Text (рисунки 14, 15).

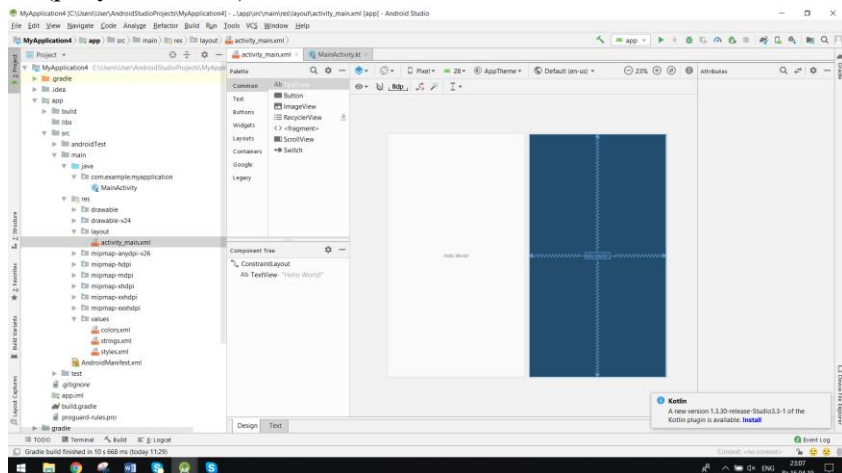


Рис.14. Просмотр файла activity_main.xml в визуальном режиме

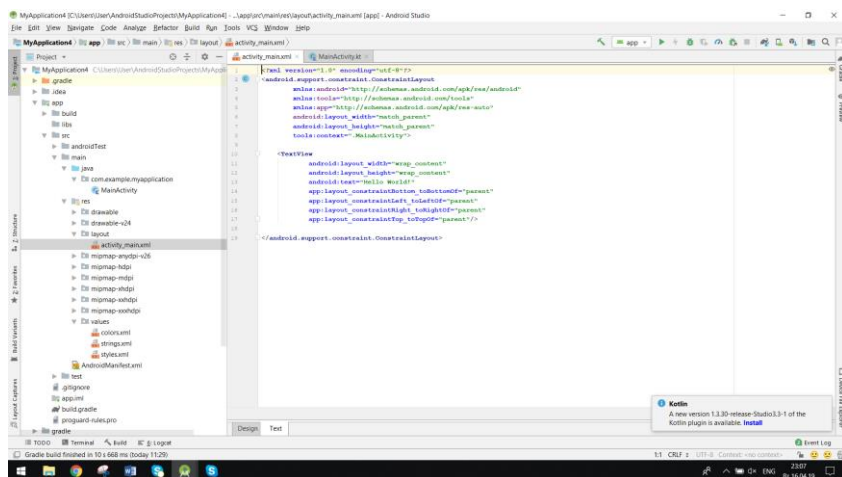


Рис.15. Просмотр файла activity_main.xml в текстовом режиме

Структура XML-файла достаточно проста - стандартное дерево XML-элементов, где каждый узел является именем класса View (TextView - один из элементов View). Вы можете создать интерфейс

программы, используя структуру и синтаксис XML. Подобный подход позволяет разделить код программы и визуальное представление.

Создание эмулятора

Для запуска приложения создадим [эмулятор](#), для этого вызовем AVD Manager при помощи соответствующего значка в верхней правой части экрана и нажмем Create New Virtual Device (рисунок 16).

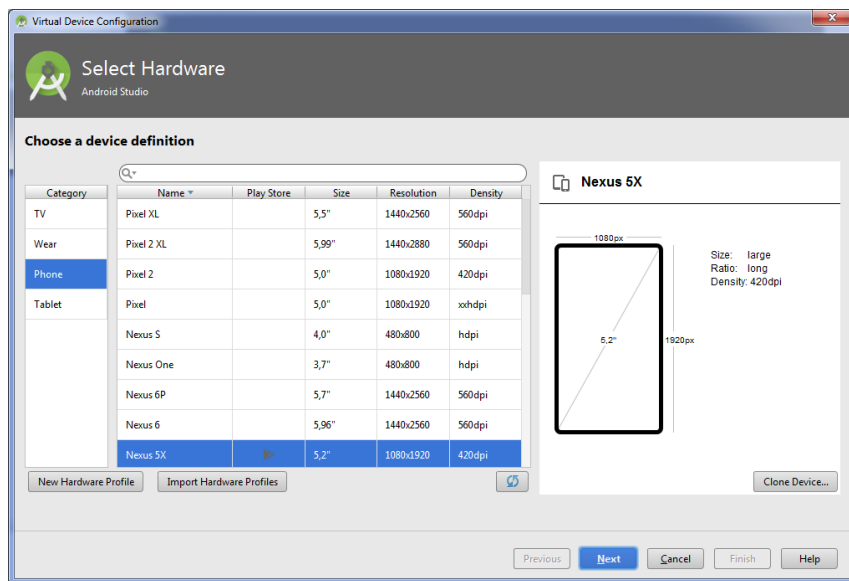


Рис.16. Выбор форм-фактора устройства

Здесь можно выбрать форм-фактор устройства. Оставим то, что выбрано по умолчанию и нажмем Next.

В следующем окне можно скачать необходимый эмулятор. Далее нам предложат указать название эмулятора и поменять его настройки. По окончании жмем Finish.

Теперь при запуске приложения (Shift + F10) мы можем выбрать созданный эмулятор и запустить его.

Через какое-то время появится эмулятор и запустится приложение (рисунок 17).

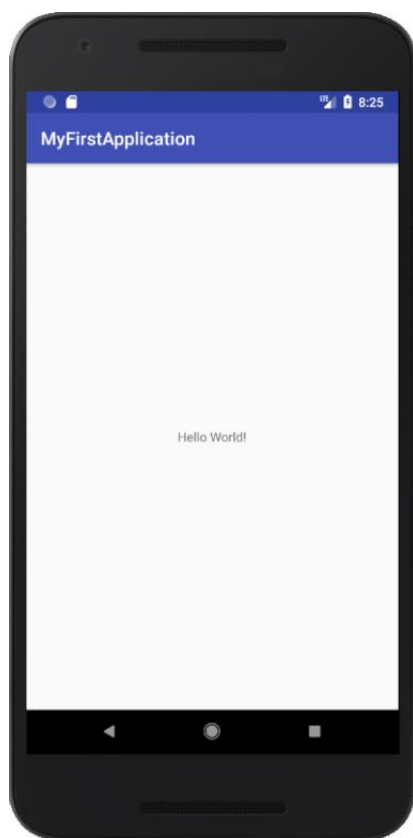


Рис. 17. Запущенное Android-приложение

ЗАДАНИЕ НА ЛАБОРАТОРНУЮ РАБОТУ

Для всех вариантов установить и настроить среду Android Studio и Android Virtual Device. Изучить принцип представления цвета в Android. Разобраться с величинами измерения расстояний и размеров в Android. Создать приложение согласно заданию указанному в варианте.

Вариант 1

Для активности выполнить следующие изменения:

1. Цвет фона активности изменить на красный
2. Задать название активности - фамилия студента
3. Цвет фона текстового поля изменить на зеленый
4. Расположить элемент TextView1 в левом верхнем углу
5. Для компонента TextView1 изменить начертание шрифта на TimesNewRoman
6. Расположить еще один элемент TextView2 по центру активности
7. Размер текста TextView2 изменить на 34

Изменить два произвольных параметра активности. Изменения выполнить как в файле ресурсов, так и в xml -коде.

Вариант 2

Для активности выполнить следующие изменения:

1. Цвет фона активности изменить на оранжевый
2. Задать название активности - название рабочей станции
3. Изменить иконку приложения на произвольную
4. Добавить два текстового поля TextView1 - по центру активности и TextView2 - в верхнем правом углу,
5. Для TextView1 цвет фона изменить на желтый, размер текста 20, тень для текста красная
6. Для TextView2 расположить текст по центру, изменить Начертание шрифта на Courier.

Изменить два произвольных параметра активности. Изменения выполнить как в файле ресурсов, так и в xml -коде.

Вариант 3

Для активности выполнить следующие изменения

1. Цвет фона активности изменить на зеленый
2. Название активности – группа студента
3. Расположить элемент TextView1 в нижнем правом углу
4. Цвет фона текстового поля заменить на желтый
5. Изменить размер элемента TextView1 – ширина 20, высота 30,
6. Изменить начертание текста в текстовом поле на жирный, подчеркнутый

Изменить два произвольные параметра активности. Изменения выполнить как в файле ресурсов, так и в xml -коде.

Вариант 4

Для активности выполнить следующие изменения

1. Цвет фона активности изменить на желтый
2. Название активности – столица Словении
3. Заменить иконку приложения на произвольную
4. Добавить два элемента – TextView1 и TextView2. TextView1 расположить в левом верхнем углу, TextView – в правом нижнем углу
5. Для TextView1 изменить цвет текста на красный, размер текста 30
6. Для TextView2 изменить цвет фона на фиолетовый, цвет тени текста – белый

Изменить два произвольные параметра активности. Изменения выполнить как в файле ресурсов, так и в xml-коде.

Вариант 5

Для активности выполнить следующие изменения

1. Цвет фона активности изменить на розовый
2. Название активности – автор романа «Искра жизни»
3. Цвет фона текстового поля – желтый
4. Изменить текст текстового поля на “Hello, Android”
5. Расположить текст по левому краю, изменить начертание – Calibri, курсив

6. Изменить цвет текста на синий, размер текста – 40

Изменить два произвольные параметра активности. Изменения выполнить как в файле ресурсов, так и в xml -коде.

Вариант 6

Для активности выполнить следующие изменения

1. Цвет фона активности изменить на синий
2. Название активности – название специальности студента
3. добавить два текстового поля TextView1 – внизу, по центру активности и TextView2 – в верхнем правом углу,
4. Для TextView1 изменить цвет фона на зеленый, задать тень для текста красного цвета
5. Для TextView2 изменить размер текста на 8, начертание текста – подчеркнутый, курсив
6. Изменить размер текстового поля – ширина 15, высота 15

Изменить два произвольные параметра активности. Изменения выполнить как в файле ресурсов, так и в xml-коде.

Вариант 7

Для активности выполнить следующие изменения

1. Цвет фона активности изменить на черный
2. Название активности – одно из чудес света
3. Заменить иконку приложения на произвольную
4. Расположить элемент TextView по центру, справа
5. Расположить текст по правому краю, изменить цвет текста на коричневый, задать тень для текста желтого цвета
6. Изменить размер текстового поля – ширина 40, высота 20

Изменить два произвольные параметра активности. Изменения выполнить как в файле ресурсов, так и в xml-коде.

Вариант 8

Для активности выполнить следующие изменения

1. Цвет фона активности изменить на бирюзовый
2. Название активности – фамилия первой учительницы
3. Заменить иконку приложения на произвольную

4. Добавить три элемента TextView1, TextView2, TextView3. Элемент TextView1 расположить в левом верхнем углу, TextView2 – по центру активности, TextView3 – в правом нижнем углу
 5. Для TextView1 цвет фона заменить на красный, для TextView2 – на синий, для TextView3 – на зеленый. Для TextView1, TextView2, TextView3 изменить текст на произвольный. Расположение текста для TextView1 – левый край, для TextView3 – правый край. Для TextView1 TextView2, TextView3 изменить начертание текста на жирный, курсив, Arial
 6. Изменить размер текста для TextView1 на 50
- Изменить два произвольные параметра активности. Изменения выполнить как в файле ресурсов, так и в xml-коде.

Вариант 9

Для активности выполнить следующие изменения

1. Цвет фона активности заменить на серый,
2. Название активности - тема первой курсовой работы,
3. Расположить элемент TextView по центру, внизу,
4. Цвет фона текстового поля изменить на фиолетовый,
5. Изменить цвет текста элемента TextView на оранжевый, задать тень для текста зеленого цвета, начертание текста – подчеркнутый,
6. Изменить размер текстового поля – ширина 30, высота 35

Изменить два произвольные параметра активности. Изменения выполнить как в файле ресурсов, так и в xml -коде.

Вариант 10

Для активности выполнить следующие изменения

1. Цвет фона активности изменить на коралловый
2. Название активности – столица Хорватии
3. Добавить два элемента TextView1, TextView2. Элемент TextView1 расположить по центру активности, TextView2 – по центру активности, сверху
4. Для TextView1 цвет фона заменить на медный, для TextView2 – на пурпурный. Для TextView1 изменить текст на произвольный

Расположение текста для TextView1 – левый край. Для TextView1, TextView2 изменить начертание текста на подчеркнутый.

5. Для TextView1, TextView2 задать тень для текста голубого цвета

6. Изменить размер TextView1 - сделать квадратным

Изменить два произвольные параметра активности. Изменения выполнить как в файле ресурсов, так и в xml-коде.

Вариант 11

Для активности выполнить следующие изменения

1. Цвет фона активности изменить на баклажанный

2. Название активности – название данного предмета

3. Заменить иконку приложения на произвольную

4. Цвет фона текстового поля изменить на салатовый

5. Расположить текстовое поле в левом нижнем углу

6. Цвет текста изменить на красный, цвет тени для текста – белый, начертание текста – подчеркнутый, жирный.

Изменить два произвольные параметра активности. Изменения выполнить как в файле ресурсов, так и в xml-коде.

Вариант 12

Для активности выполнить следующие изменения

1. Цвет фона активности изменить на хаки

2. Название активности – название версии Android 2.0

3. Заменить иконку приложения на произвольную

4. Добавить два текстового поля TextView 1 – по центру активности и TextView 2 – в верхнем левом углу

5. Для TextView 1 цвет фона изменить на бледно-синий, размер текста 30, тень для текста красная

6. Для TextView 2 расположить текст по правому краю, изменить начертание шрифта на Chiller начертание текста – подчеркнутый, жирный.

Изменить два произвольные параметра активности. Изменения выполнить как в файле ресурсов, так и в xml-коде.

Вариант 13

Для активности выполнить следующие изменения

1. Цвет фона активности изменить на чертополох,
 2. Название активности – фамилия разработчика ОС Android.
 3. Цвет фона текстового поля – сепия,
 4. Изменить текст текстового поля на “Hello-Hello”,
 5. Расположить текст по правому краю, изменить начертание – Times New Roman, курсив, жирный
 6. Изменить цвет текста на белый, размер текста – 15
- Изменить два произвольные параметра активности. Изменения выполнить как в файле ресурсов, так и в xml-коде.

КОНТРОЛЬНЫЕ ВОПРОСЫ И ЗАДАНИЯ

1. Дайте краткую характеристику платформы Android.
2. Перечислите уровни архитектуры системы Android. Дайте краткую характеристику каждого из уровней.
3. Перечислите основной инструментарий для разработки Android-приложений.
4. Опишите платформу Android Studio и основные компоненты, входящие в установку.
5. Перечислите шаги по установке платформы Android Studio.
6. Опишите порядок действий по настройке Android Studio.
7. Опишите роль AVD.
8. Приведите пример параметров, которые доступны при настройке AVD.
9. Опишите структуру проекта в Android Studio.
10. Опишите роль файла AndroidManifest.xml.
11. Опишите режимы, предоставляемые Android Studio для разработки интерфейса приложения.
12. Опишите структуру XML-файла.

ФОРМА ОТЧЕТА ПО ЛАБОРАТОРНОЙ РАБОТЕ

На выполнение лабораторной работы отводится 1 занятия (2 академических часа: 1 час на выполнение и сдачу лабораторной работы и 1 час на подготовку отчета).

Номер варианта студенту выдается преподавателем.

Отчет на защиту предоставляется в печатном виде.

Структура отчета (на отдельном листе(-ах)): титульный лист, формулировка задания (вариант), листинг (xml код разметки экрана, графическое представление, соответствующее разметке экрана, код программы и при необходимости наличие кода дополнительных классов), результаты выполнения работы (графические изображения примеров работы приложения), выводы).

ОСНОВНАЯ ЛИТЕРАТУРА

1. Семакова, А. Введение в разработку приложений для смартфонов на ОС Android / А. Семакова. - 2-е изд., испр. - М. : Национальный Открытый Университет «ИНТУИТ», 2016. - 103 с. : ил. ; То же [Электронный ресурс]. - URL: <http://biblioclub.ru/index.php?page=book&id=429181>
2. Введение в разработку приложений для ОС Android / Ю.В. Березовская, О.А. Юфрякова, В.Г. Вологодина и др. - 2-е изд., испр. - М. : Национальный Открытый Университет «ИНТУИТ», 2016. - 434 с. : ил. - Библиогр. в кн. ; То же [Электронный ресурс]. - URL: <http://biblioclub.ru/index.php?page=book&id=428937>
3. Разработка приложений для смартфонов на ОС Android / Е.А. Латухина, О.А. Юфрякова, Ю.В. Березовская, К.А. Носов. - 2-е изд., исправ. - М. : Национальный Открытый Университет «ИНТУИТ», 2016. - 252 с. : ил. - Библиогр. в кн. ; То же [Электронный ресурс]. - URL: <http://biblioclub.ru/index.php?page=book&id=428807>

ДОПОЛНИТЕЛЬНАЯ ЛИТЕРАТУРА

4. Дэвид, Х. Разработка приложений Java EE 6 в NetBeans 7. [Электронный ресурс] — Электрон. дан. — М. : ДМК Пресс, 2013. — 330 с. — Режим доступа: <http://e.lanbook.com/book/58693> — Загл. с экрана.
5. Сильвен, Р. Android NDK. Разработка приложений под Android на C/C++. [Электронный ресурс] — Электрон. дан. — М. : ДМК Пресс, 2012. — 496 с. — Режим доступа: <http://e.lanbook.com/book/9126> — Загл. с экрана.
6. Ретабоуил, С. Android NDK: руководство для начинающих. [Электронный ресурс] — Электрон. дан. — М. : ДМК Пресс, 2016. — 518 с. — Режим доступа: <http://e.lanbook.com/book/82810> — Загл. с экрана.

7. Соколова, В.В. Разработка мобильных приложений: учебное пособие / В.В. Соколова ; Федеральное государственное автономное образовательное учреждение высшего образования «Национальный исследовательский Томский государственный университет», Министерство образования и науки Российской Федерации. - Томск: Издательство Томского политехнического университета, 2015. - 176 с.: ил., табл., схем. - Библиогр. в кн.. - ISBN 978-5-4387-0369-3; То же [Электронный ресурс]. - URL: <http://biblioclub.ru/index.php?page=book&id=442808> (15.06.2017).
8. Баженова, И.Ю. Язык программирования Java / И.Ю. Баженова. - М.: Диалог-МИФИ, 2008. - 254 с. : табл., ил. - ISBN 5-86404-091-6 ; То же [Электронный ресурс]. - URL: <http://biblioclub.ru/index.php?page=book&id=54745> (15.06.2017).
9. Джошуа Блох Java. Эффективное программирование [Электронный ресурс]/ Джошуа Блох— Электрон. текстовые данные. — Саратов: Профобразование, 2017.— 310 с.— Режим доступа: <http://www.iprbookshop.ru/64057.html> .— ЭБС «IPRbooks»

Электронные ресурсы:

10. Научная электронная библиотека <http://eLIBRARY.RU>
11. Электронно-библиотечная система <http://e.lanbook.com>
12. Электронно-библиотечная система «Университетская библиотека онлайн» <http://biblioclub.ru>
13. Электронно-библиотечная система IPRBook <http://www.iprbookshop.ru>
14. Базовый сайт о системе Android - https://www.android.com/intl/ru_ru
15. Разработка приложения на базе Android - <https://developer.android.com/index.html>