**Лабораторная работа №1**

**Исследование процесса создания мобильных приложений в среде Android Studio**

Цель работы: Ознакомиться с процессом написания программы в Android Studio для операционной системы Android. Ознакомиться с принципом добавления в программу элементов интерфейса (кнопки, текстовые поля ввода и т.п.) и обработкой событий этих элементов в Android.

**Ход работы**

1. Т.к. разработка приложения будет на языке Java, чтобы начать работать с Android Studio нам в первую очередь нужно загрузить и установить пакет JDK 10, который необходим для разработки на языке Java. JDK 10 можно найти на сайте компании Oracle.

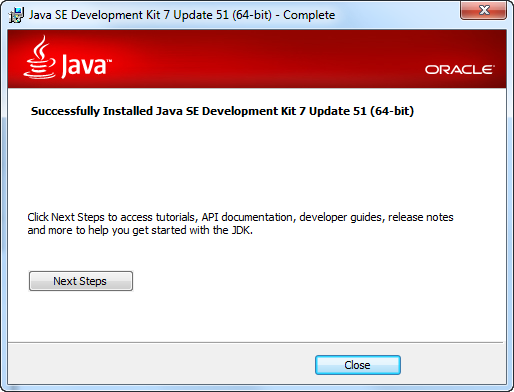


Рисунок 1 – Установка JDK

2. Далее нам нужно поставить среду разработки под Android, в которой мы будем работать. Выберем Android Studio, т.к. это нативное средство разработки приложений для данной платформы и разработчики мобильных в основном используют его. Скачать Android Studio можно с помощью официального сайта: <http://developer.android.com/sdk/index.html>

Скорее всего вы не сможете зайти на данный сайт без Прокси/VPN, поэтому если такое произошло – лучше использовать VPN-сервисы при любых манипуляция с Android Studio, т.к. в противном случае у вас будут выпадать ошибки. Можете использовать бесплатные CyberGhost, Windscribe, либо любой другой VPN-сервис.

3. На следующем шаге нужно установить Android SDK.

Android SDK - Software Development Kit, это набор всех необходимых для создания Андроид-приложений библиотек и файлов. Этот набор также включает в себя сэмпл-коды и Android Virtual Device Manager - менеджер виртуальных устройств, который позволяет протестировать разрабатываемое приложение на любой версии ОС Android

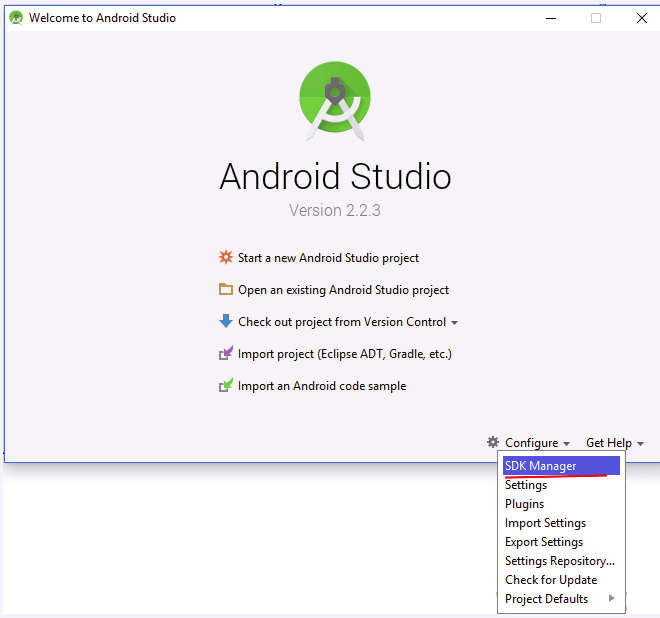


Рисунок 2 – Установка Android SDK

Провести установку можно из Android Studio через SDK Manage (рис. 3), либо также можно скачать через сторонний сайт, но тогда нужно будет в SDK Manager указать путь к Android SDK

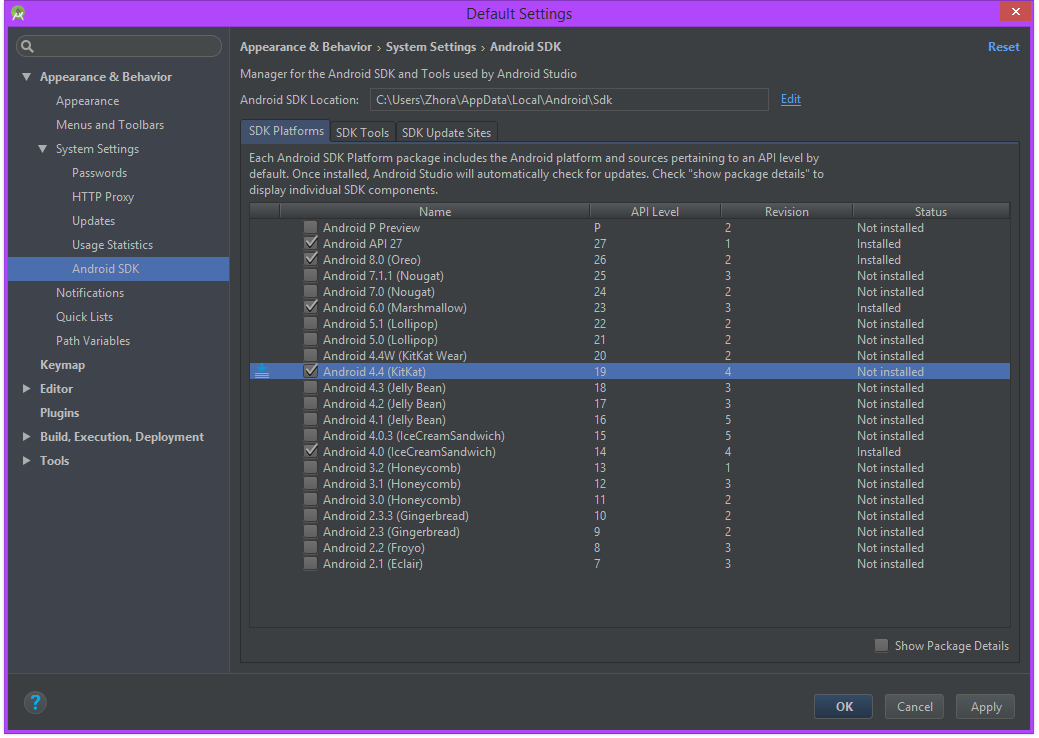


Рисунок 3 – Окно выбора Android SDK

4. Теперь создадим первое приложение в среде Android Studio для операционной системы Android. Откроем Android Studio и на начальном экране выберем пункт Start new Android Project:

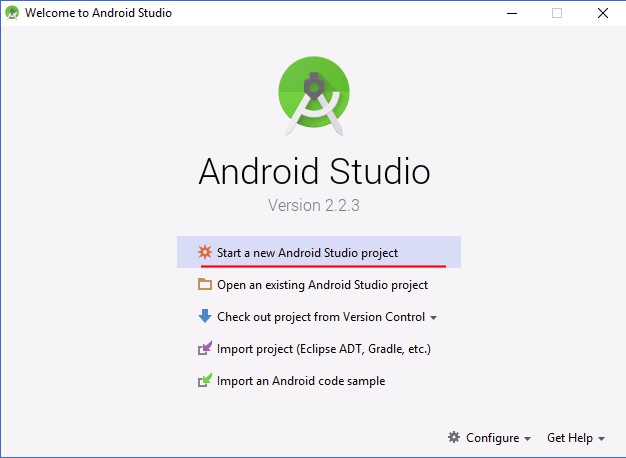


Рисунок 4 – Создание проекта

Появится окно (рис. 5) конфигурации проекта, выберите название и минимальную версию SDK. На момент 2018 года использование API 16 покрывало 98% потребителей, поэтому выбирайте его и нажмите Next.

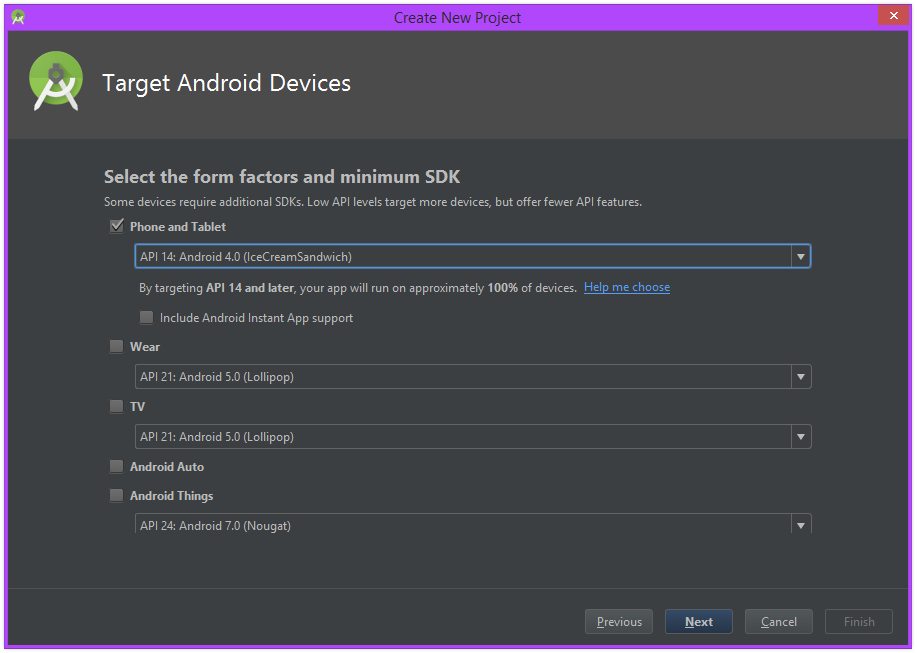
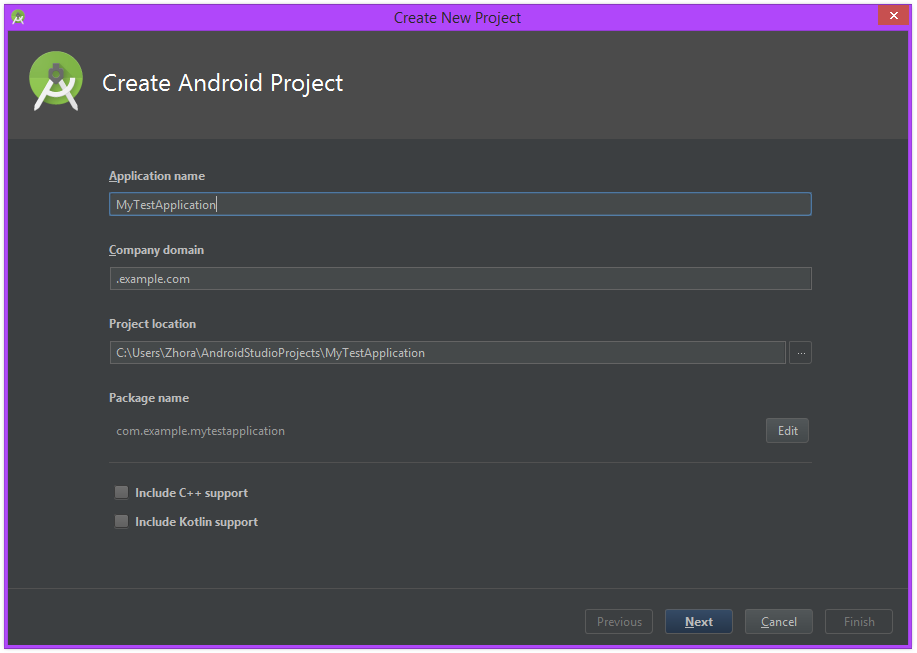


Рисунок 5 – Окно конфигурации проекта

Далее можно выбрать шаблон, на основе которого можно написать собственное приложение. В данной работе лучше всего выбрать пустой шаблон.

5. После всего проделанного проект (программу) уже можно откомпилировать, установить на телефон или планшет, и запустить.

Чтобы запустить программу на компьютере, нужно вначале создать и запустить виртуальный смартфон – в эмуляторе Android. Он встроен в Android Studio и создается достаточно просто в несколько шагов

**Создание эмулятора Android**

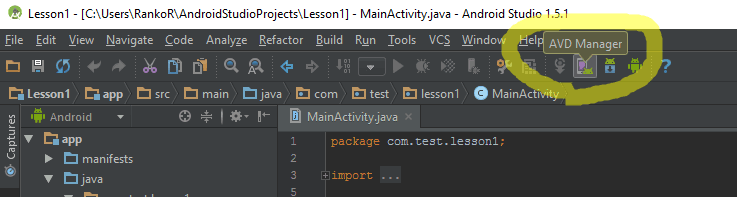
Для создания эмулятора Android нам понадобится **Android AVD Manager** (AVD = Android Virtual Device). Не беспокойтесь, ставить больше ничего не потребуется. Просто нажмите на кнопку – рис. 6:

Рисунок 6 – AVD Manager

Потом на следующую кнопку:

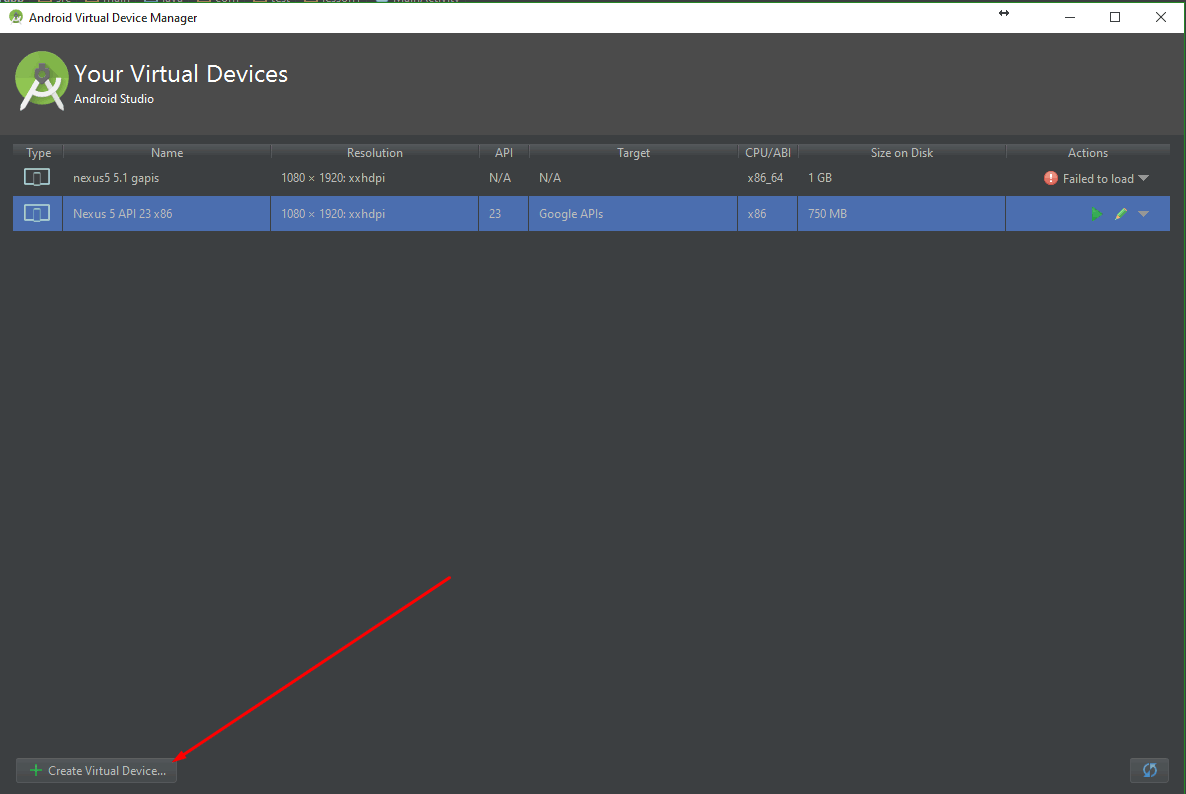


Рисунок 7 – Create Virtual Device

А потом несколько раз нажмите «Next» и, наконец, «Finish»

**Модификация первого приложения**

6. Откомпилированный и запущенный только что созданный пустой проект, выводит пустую форму activity\_main.xml. Для того чтобы её модифицировать выберем в Project Explorer выберите ваш проект и нажмите res\layout\activity\_main.xml (рис. 8).

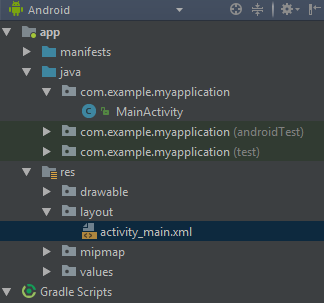


Рисунок 8 – Окно Project Explorer

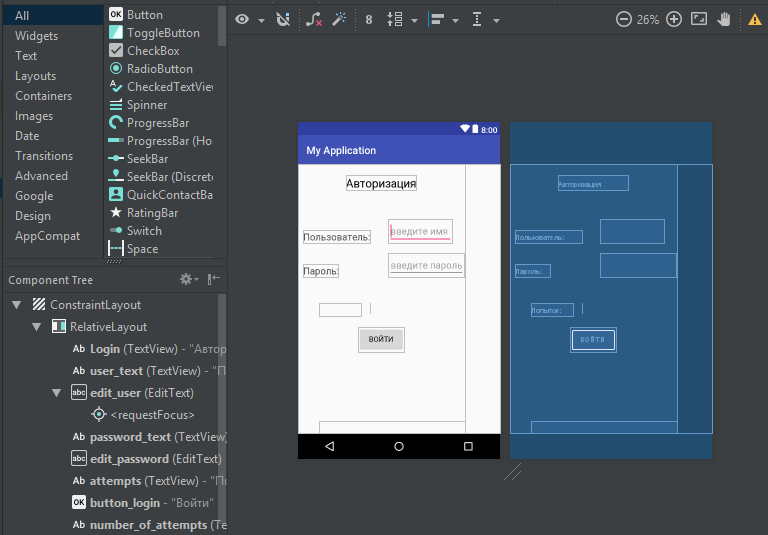
Откроется окно визуальною редактирования содержимою экрана нашей программы (приложения) – рисунок 9. С расположенною в левой части окна Palette сначала добавьте RelativeLayout - он нужен для более удобного размещения элементов на форме. Далее перетащите на окно текстовое поле “TextView”. Нажмите на это поле и в окне Properties в правой части экрана в ноле текст введите новую надпись для него “Авторизация” – рисунок 9.

Рисунок 9 – Окно визуального редактирования содержимого экрана программы

Во вкладке «Text» мы можем отредактировать нашу форму вручную. Например так выглядит наше TextView «Авторизация», расположенное по центру:

<TextView

android:id="@+id/Login"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_alignParentTop="true"

android:layout\_centerHorizontal="true"

android:layout\_marginTop="20dp"

android:text="Авторизация"

android:textAppearance="?android:attr/textAppearanceLarge" />

Таким же образом добавить в форму TextView «Пользователь» и «Пароль». Далее нам нужно добавить в форму поле для ввода имени и пароля, для этого выберем поле с вводом текста «EditText» и добавим на нашу форму. Для того чтобы пароль скрывался – нужно либо выбрать в Palette “Password”, либо в аргументе inputType выбрать тип textPassword. Должно получится примерно вот так:

<EditText

android:id="@+id/edit\_user"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_alignBottom="@+id/user\_text"

android:layout\_marginLeft="35dp"

android:layout\_toRightOf="@+id/user\_text"

android:hint="введите имя " >

<requestFocus />

</EditText>

<EditText

android:id="@+id/edit\_password"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_alignBottom="@+id/password\_text"

android:layout\_alignLeft="@+id/edit\_user"

android:hint="введите пароль"

android:inputType="textPassword" />

По аналогии добавим кнопку «Войти», должно в итоге получиться примерно так:

<Button

android:id="@+id/button\_login"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_below="@+id/edit\_password"

android:layout\_centerHorizontal="true"

android:layout\_marginTop="94dp"

android:onClick="Login"

android:text="Войти" />

Можно попробовать запустить приложение и понажимать на эту кнопку, но ничего не произойдет. Почему? Мы пока не написали никакого текста программы, который бы реагировала это нажатие.

В редакторе нажмите на кнопку и посмотрите в окне Properties значение поля Id. Допустим, там написано “@+id/button 1”. Это значит, что Id кнопки - “button 1”. Пo этому Id к ней можно обращаться из нашей программы. Лучше поменять идентификатор на что-то более логичное, чтобы в будущем не перепутать кнопки и сделать код при обращении к этой кнопке более читаемым.

Для того чтобы по нажатию на кнопку происходили нужные нам действия – нужный сценарий нужно написать на языке Java/Kotlin. Для этого в Project Explorer для своего проекта выберите app\java\ myapplication\MainActivity.java

Чтобы работать с текстовой надписью и кнопкой, для начала нужно создать поля (переменные) этих типов. В Java для Android тин данных для кнопок называется Button, а тип данных для надписей - TextView.

Добавьте в класс MainActivity поле кнопки и иоле текстовой строки, придумав для них логичные названия, например:

private EditText username;

private EditText password;

private Button login;

private TextView loginLocked;

private TextView attempts;

private TextView numberOfAttempts;

Android Studio подчеркнет TextView и Button как ошибки. Чтобы устранить ошибку, добавьте в программу import всех нужных типов данных.

Теперь в методе onCreate (он будет запускаться при инициализации окна программы) добавьте следующие две строки:

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

// Связываемся с элементами нашего интерфейса:

username = (EditText) findViewById(R.id.edit\_user);

password = (EditText) findViewById(R.id.edit\_password);

login = (Button) findViewById(R.id.button\_login);

loginLocked = (TextView) findViewById(R.id.login\_locked);

attempts = (TextView) findViewById(R.id.attempts);

numberOfAttempts = (TextView) findViewById(R.id.number\_of\_attempts);

numberOfAttempts.setText(Integer.toString(numberOfRemainingLoginAttempts))

Последнюю строку мы добавили «на будущее», для того чтобы данный текст работал – нужно добавить переменную numberOfRemainingLoginAttempts.

// Число для подсчета попыток залогиниться:

int numberOfRemainingLoginAttempts = 3;

Теперь остается только назначить обработчик для события нажатия на кнопку. Это можно сделать как в данном файле, так и в xml, прописав в редакторе:

android:onClick="Login"

Также есть вариант в этом же методе onCreate добавые строку:

login.setOnClickListener(*someAction*);

В класс добавые новый экземпляр класса OnClickListener следующим образом:

*OnClickListener someAction = new OnClickListener() {*

*public void onClick (Viev v) {*

*//действия*

*}*

Здесь действия будет выполняться при нажатии кнопки.

Теперь нужно написать функция Login() – обработка события кнопки.

// Обрабатываем нажатие кнопки "Войти":

public void Login(View view) {

// Если введенные логин и пароль будут словом "admin",

// показываем Toast сообщение об успешном входе:

if (username.getText().toString().equals("admin") &&

password.getText().toString().equals("admin")) {

Toast.makeText(getApplicationContext(), "Вход выполнен!",Toast.LENGTH\_SHORT).show();

// Выполняем переход на другой экран:

Toast.makeText(getApplicationContext(), "Переход успешен!",Toast.LENGTH\_SHORT).show();

}

// В другом случае выдаем сообщение с ошибкой:

else {

Toast.makeText(getApplicationContext(), "Неправильные данные!",Toast.LENGTH\_SHORT).show();

numberOfRemainingLoginAttempts--;

// Делаем видимыми текстовые поля, указывающие на количество оставшихся попыток:

attempts.setVisibility(View.VISIBLE);

numberOfAttempts.setVisibility(View.VISIBLE);

numberOfAttempts.setText(Integer.toString(numberOfRemainingLoginAttempts));

// Когда выполнено 3 безуспешных попытки залогиниться,

// делаем видимым текстовое поле с надписью, что все пропало и выставляем

// кнопке настройку невозможности нажатия setEnabled(false):

if (numberOfRemainingLoginAttempts == 0) {

login.setEnabled(false);

loginLocked.setVisibility(View.VISIBLE);

loginLocked.setBackgroundColor(Color.RED);

loginLocked.setText("Вход заблокирован!!!");

}

}

}

В данной пункте от студента требуется разобраться в данном программном коде и найти что делают неизвестные ему функции, используя документацию либо находя ответы в сети Интернет.

*Примечание*. Для отладки программ можно использовать Android-устройство, в котором в настройках для разработчиков выбран пункт «Откладка по USB» либо создать виртуальное устройство внутри Android Studio с помощью AVD Manager (Android Virtual Device). Также есть сторонние программы для создания виртуальных машин, которые могут работать быстрее, например Genymotion (требует регистрации)

**Задания к лабораторной работе**

Нам нужно создать форму авторизации

1. Модифицировать activity\_main.xml(который находится res/layout) таким образом, чтобы пользователь видел поля ввода «введите имя» и «введите пароль». Также добавить кнопку «Войти».

2. Модифицировать java файл MainActivity (находится в java/com.example.myapplication) таким образом, чтобы при введении имени и пароля «admin» выводилось сообщение с надписью «Вход выполнен», а если имя(пароль) неверны – выводить сообщение «Неправильные данные».

3. Добавить возможность блокировки кнопки «Войти» при введении неправильного имени(пароля) трижды

**Контрольные вопросы**

1. Какие языки программирования поддерживаются для разработки приложений Android?
2. Зачем создавать AVD для платформы Android версии 1.1 (или любой более старой версии прошивки), если существуют новые версии Android SDK?
3. Редакторы ресурсов Android могут быть неудобны для обработки большого объема данных, таких как строковые ресурсы. Есть ли какой-либо способ решения этой проблемы?
4. Кто входит в состав Альянса разработчиков открытых стандартов мобильных устройств (Open Handset Alliance)?
5. Можно ли использовать Eclipse для отладки приложений на мобильном устройстве?
6. Программный стек Android.
7. Виртуальная машина Dalvik.

**Литература**

1. Google Android. Программирование для мобильных устройств / Алексей Голощапов. - БХВ-Петербург, 2012. ISBN 978-5-9775-0729-5.

2. Сайт разработчика Google Android. URL: http://developer.android.com/index.html