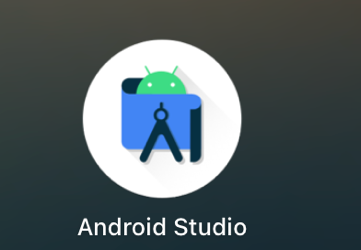
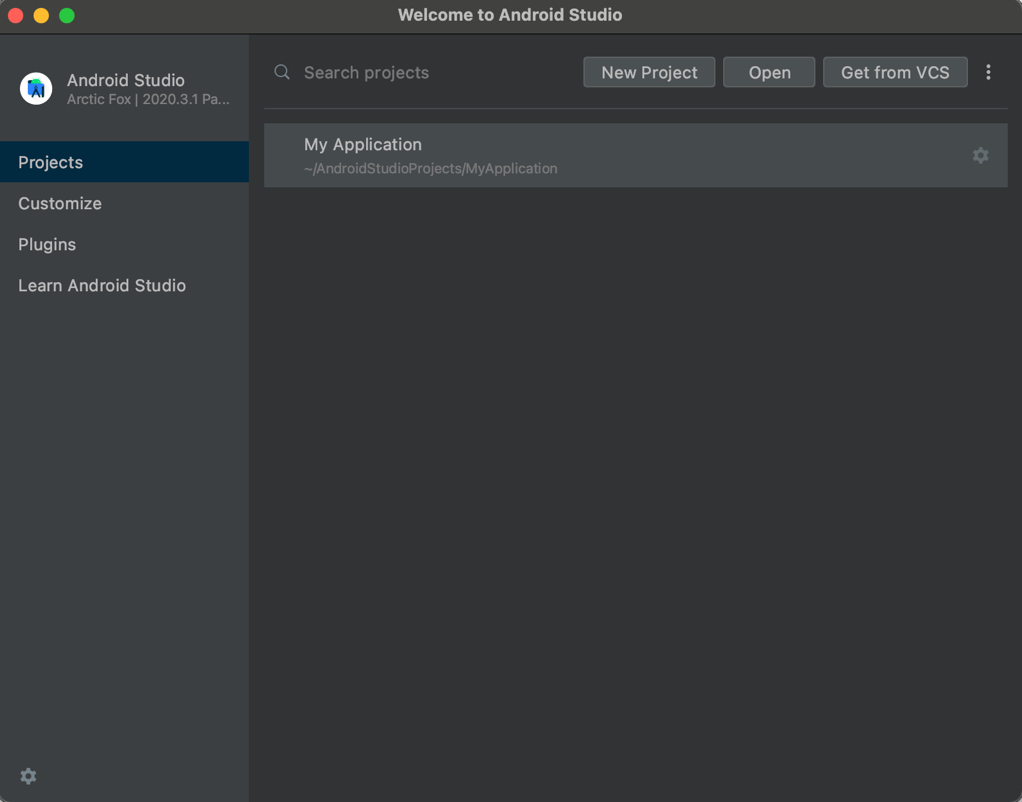
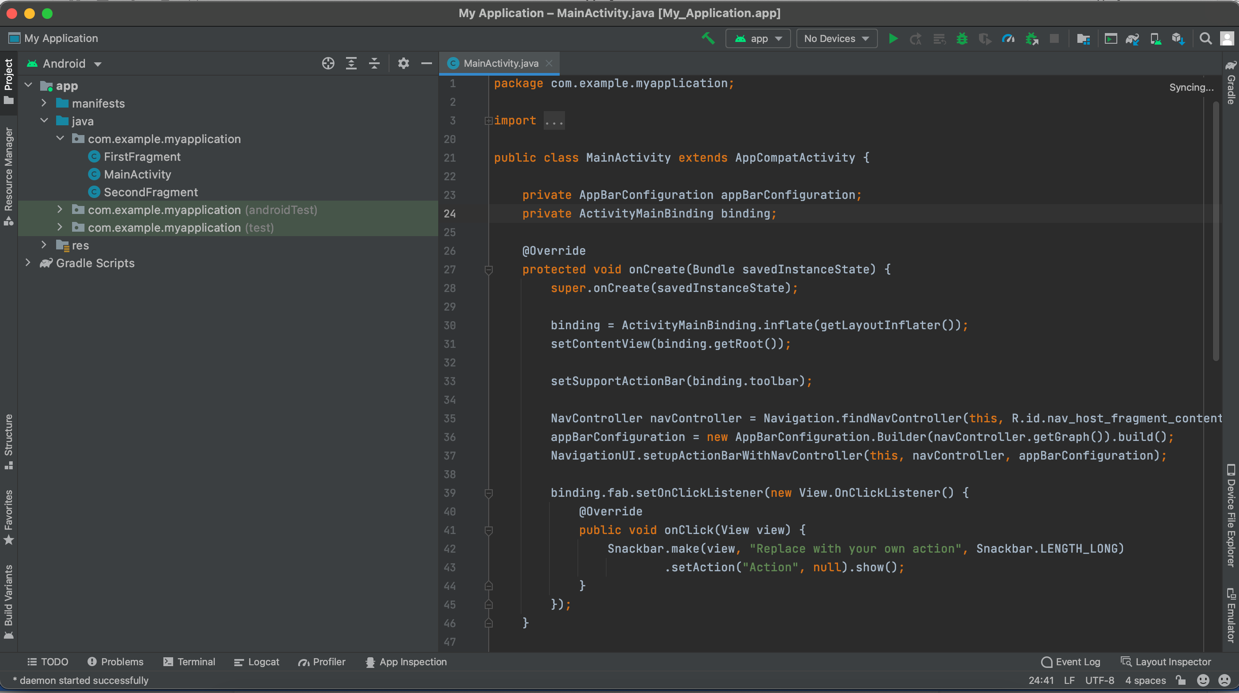
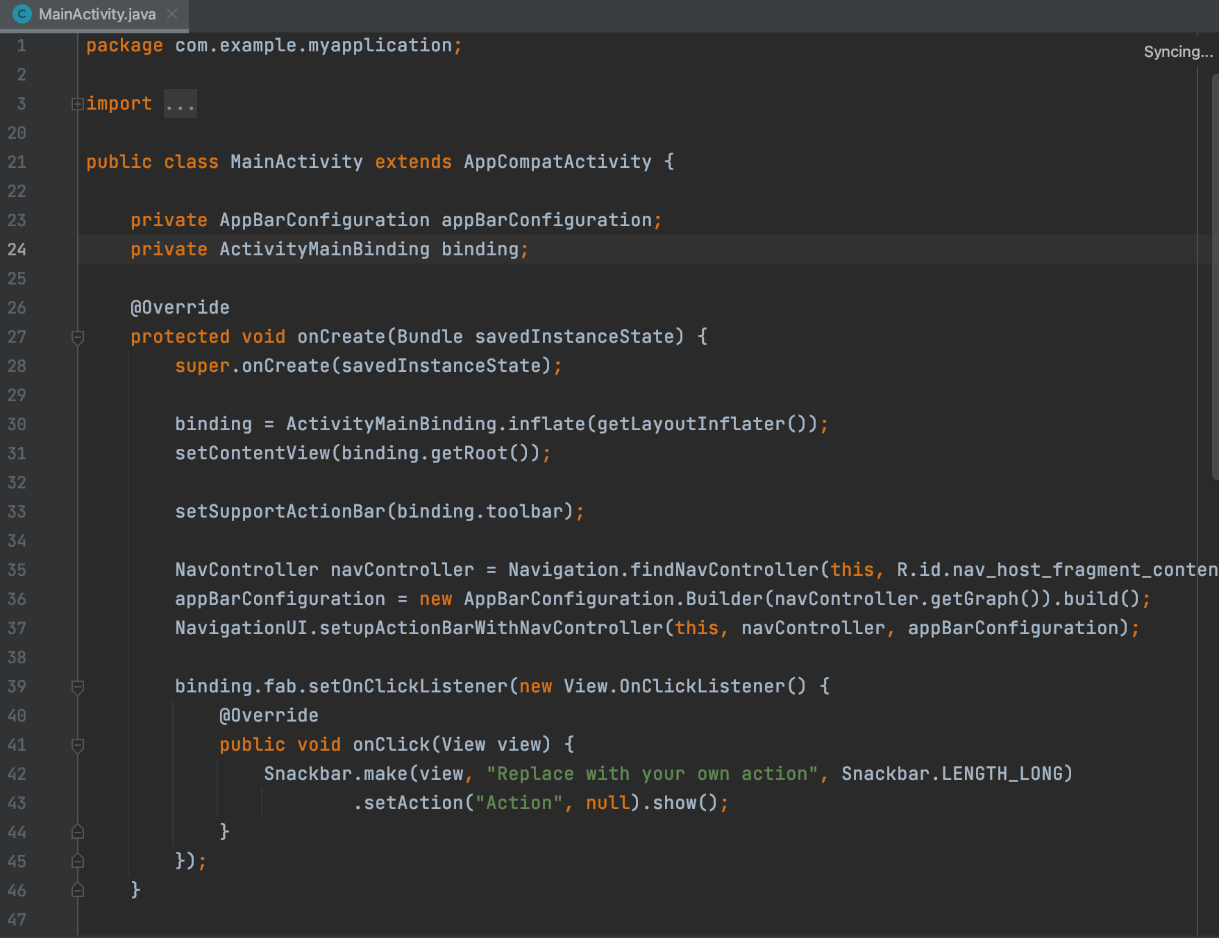
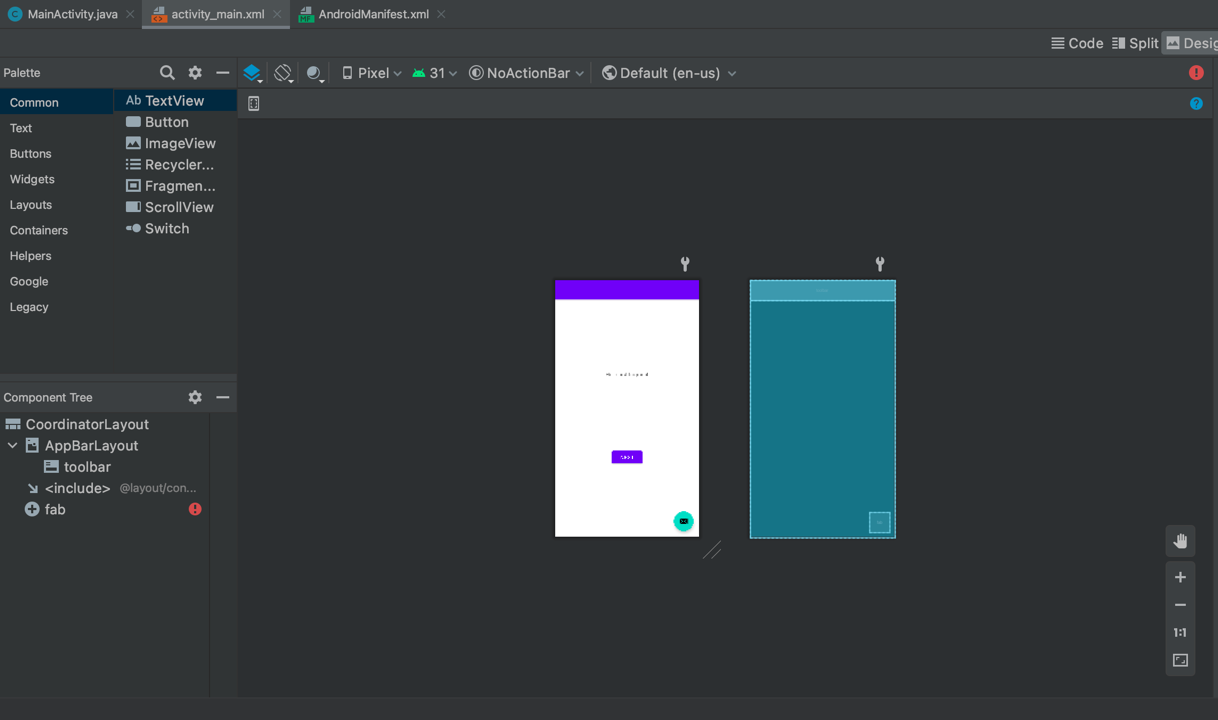
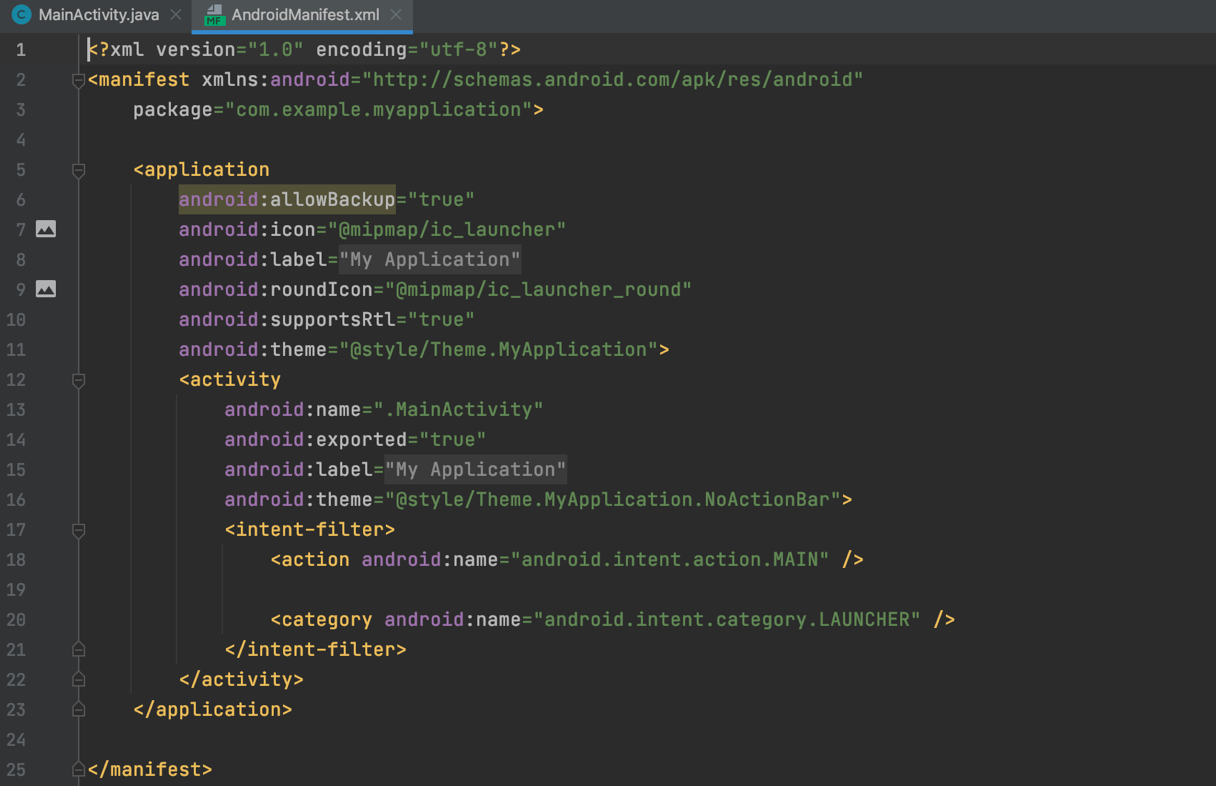
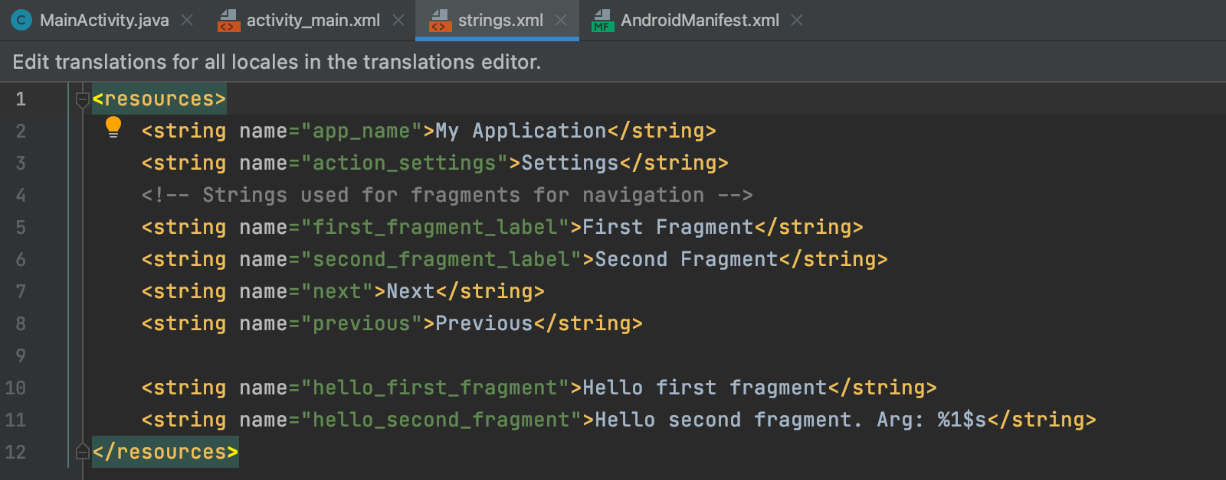
***Установка Android Studio. “Hello, World”***

**Цель работы**

Получить навыки установки и первоначальной настройки среды разработки, получить базовое представление о структуре приложения и порядке его запуска.

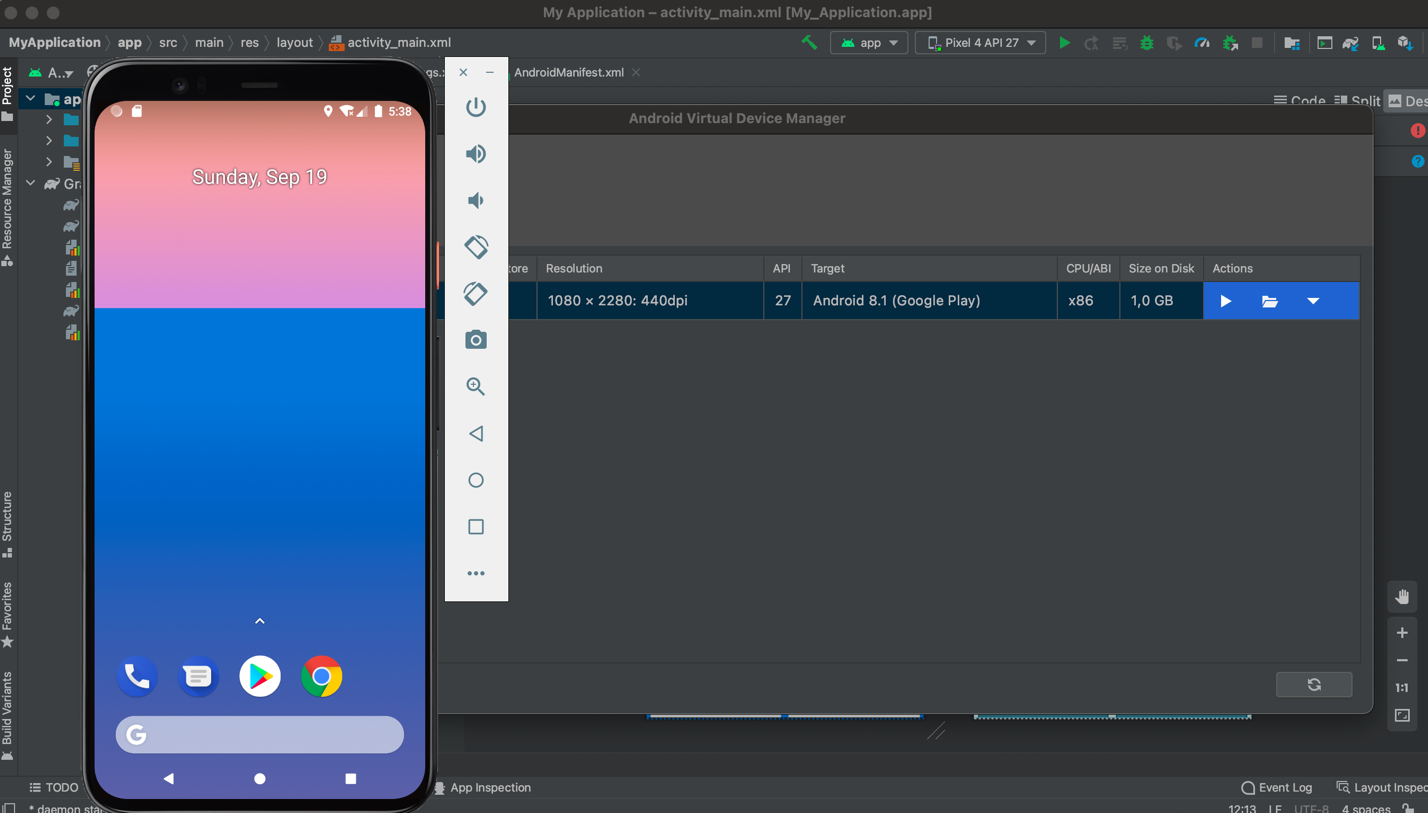
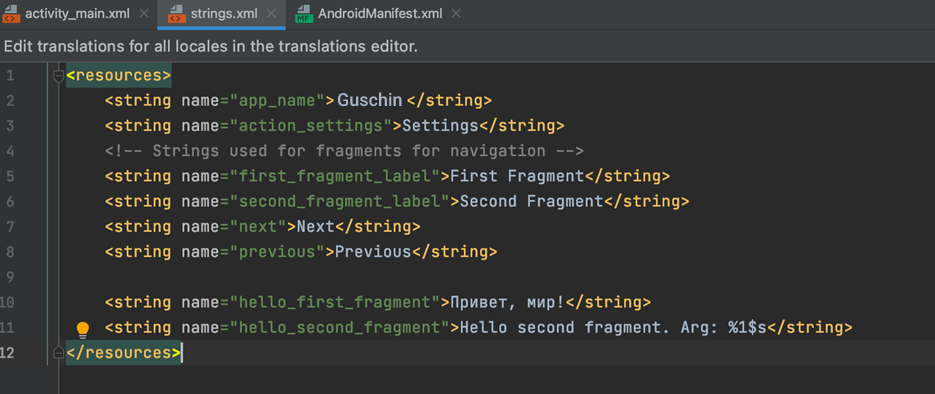
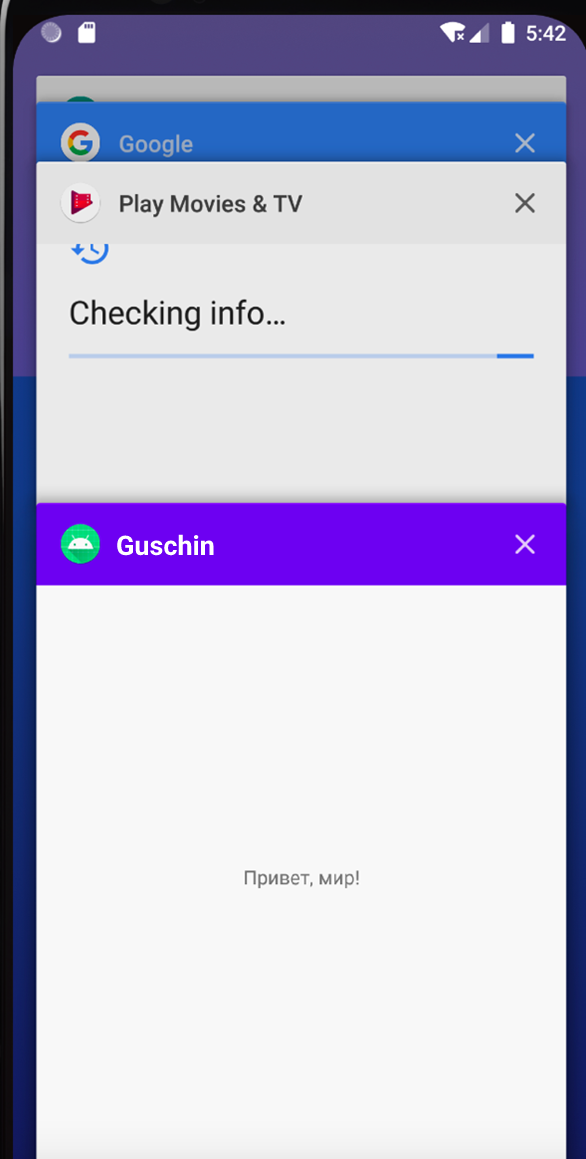
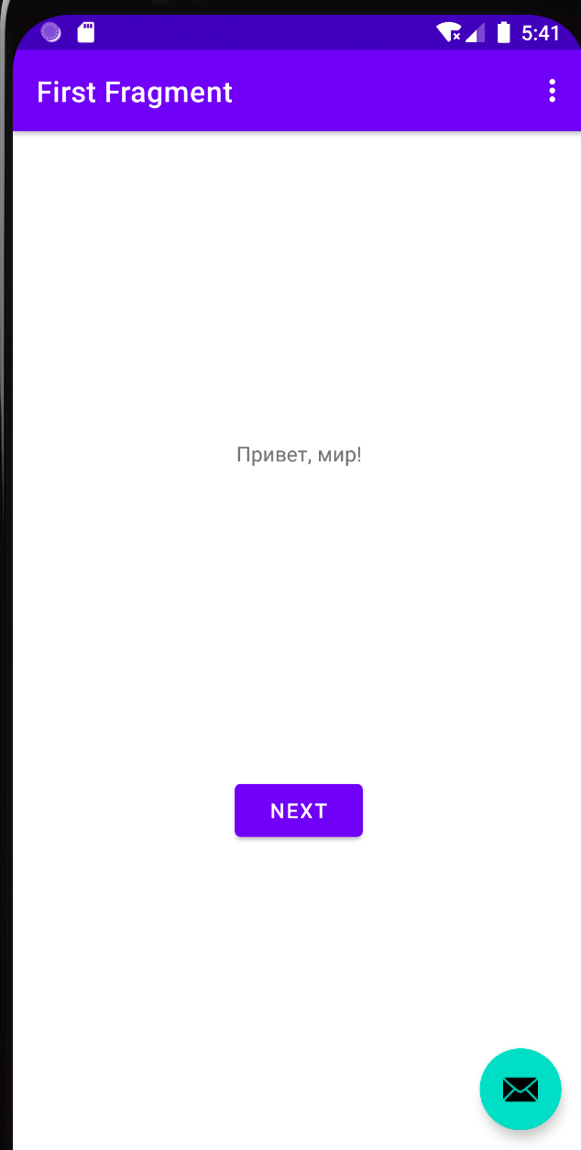
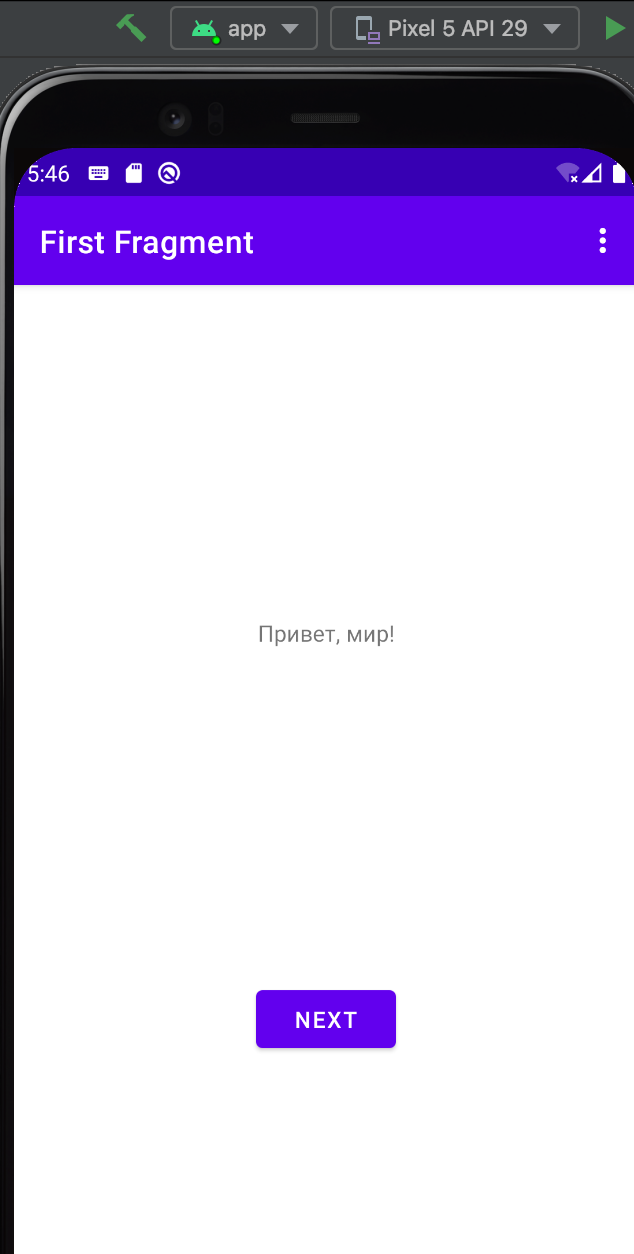
**Задания для выполнения**

1. Установите на компьютер среды разработки JDK и Android Studio.  
   
2. Запустите Android Studio.  
    
3. Создайте новый пустой проект на языке программирования Java.  
   
4. Познакомьтесь со структурой проекта и с содержанием главных файлов:
   1. MainActivity.java  
      
   2. activity\_main.xml  
      
   3. AndroidManifest.xml
   4. strings.xml  
      
5. Найдите в справочном материале и кратко опишите своими словами назначение и содержание каждого из этих файлов.  
   MainActivity.java – редактирование исходных файлов. Класс java, в начале которого идет определение класса, идет импорт классов из других пакетов, функциональность которых используется в MainActivity. Затем идет определение класса. По умолчанию MainActivity наследуется от класса AppCompatActivity, который выше подключен с помощью директивы импорта. Класс AppCompatActivity по сути представляет отдельный экран (страницу) приложения или его визуальный интерфейс. И MainActivity наследует весь этот функционал.

По умолчанию MainActivity содержит только один метод onCreate(), в котором фактически и создается весь интерфейс приложения.   
В метод setContentView() передается ресурс разметки графического интерфейса.

Activity\_main.xml – визуальный редактор. Но также мы можем работать с файлом в режиме кода, поскольку activity\_main.xml - это обычный текстовый файл с разметкой xml. Для переключения к коду нажмем на кнопку Code над графическим представлением.   
AndroidManifest.xml - манифест или конфиг-файл приложения.

можно настроить следующие возможности:  
1. Указать имя Java-пакета приложения, который служит уникальным идентификатором;  
2. Описать компоненты приложения, службы;  
3. Указать список необходимых разрешений для обращения к защищенным частям API и взаимодействия с другими приложениями;  
4. Разрешить доступ к ресурсам, которые сторонние приложения обязаны иметь для взаимодействия с компонентами данного приложения;  
5. Указать минимальный и максимальный уровень API Android, необходимый для работы приложения;  
strings.xml - хранятся строки для заголовка приложения и выводимого сообщения

1. Познакомьтесь с присутствующими в проекте ресурсами.
2. Запустите этот проект на виртуальном устройстве Android. Если виртуальных устройств не существует, создайте.  
   
3. Модифицируйте проект таким образом, чтобы в названии приложения выводилась Ваша фамилия, а в строке на главном экране - любая другая строка по Вашему желанию.  
   
4. Запустите данное приложение на нескольких виртуальных устройствах с разными версиями API и разными размерами и разрешениями экрана.   
   

**Методические указания**

Скачать JDK можно с сайта Oracle (http://www.oracle.com)

Чтобы скачать JDK нужно сначала принять условия лицензионного соглашения, а затем выбрать нужную версию.

После скачивания запустите setup-файл и установите JDK.

Скачать среду можно с сайта для разработчиков Android (http://developer.android.com/sdk/index.html).

**Контрольные вопросы**

1. Что такое Activity и зачем оно нужно?  
   Activity — это отдельный экран в Android. Это как окно в приложении для рабочего стола, или фрейм в программе на Java. Activity позволяет вам разместить все ваши компоненты пользовательского интерфейса или виджеты на этом экране. Activity – основная единица графического интерфейса (аналог окна или экранной формы)
2. Что такое View и зачем оно нужно?  
   View - базовый класс для всех представлений.   
   View объекты являются основными строительными элементами элементов пользовательского интерфейса (UI) в Android.

View - это простой прямоугольник, который реагирует на действия пользователя.Примерами являются EditText, Button, CheckBox и т.д.

View относится к классу android.view.View, который является базовым классом всех классов пользовательского интерфейса.

1. Что такое Layout и зачем оно нужно?  
   **layout/** — компоновка элементов пользовательского интерфейса. Layout бывает различных типов и отвечает за то, как будут расположены его дочерние View на экране (таблицей, строкой, столбцом …)
2. Почему все строки в Layout указываются косвенно из файла с ресурсами?
3. Что такое версия API?  
   описание способов (набор классов, процедур, функций, структур или констант), которыми одна компьютерная программа может взаимодействовать с другой программой

**Дополнительные задания**

1. Замените RelativeLayout в главном окне приложения на LinearLayout с вертикальным расположением элементов.
2. Присвойте строке, отображаемой в TextView идентификатор следующим атрибутом:

android:id=”@+id/main\_textview”

1. Измените название ресурса строки на “text1”
2. Модифицируйте строку динамически из кода Java, добавив в обработчик события onCreate следующий код:

// 1. Access the TextView defined in layout XML

// and then set its text

mainTextView = (TextView) findViewById(R.id.main\_textview);

mainTextView.setText(“Set in Java!”);