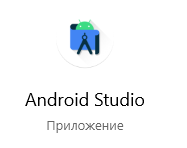
***Кузнецов ПИ20-5***

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

**Задания для выполнения**

1. Установите на компьютер среды разработки JDK и Android Studio.



1. Запустите Android Studio.
2. Создайте новый пустой проект на языке программирования Java.

Изображение выглядит как текст

Автоматически созданное описание

1. Познакомьтесь со структурой проекта и с содержанием главных файлов:

**a. MainActivity.java**

Основной Java файл. «Бэкграунд» приложения. Относится именно к приложению, к «main activity». В нем пишутся все необходимые функции, относящиеся именно к «main activity», например: onCreate() – выполняется при создании приложения.

**b. activity\_main.xml**

Основной файл layout’а. Далее его можно использовать чтобы установить содержимое Activity. Используется для отображения coordinatorLayout, tabLayout, viewPager, viewPager и т.д.

**c. AndroidManifest.xml**

Файл AndroidManifest.xml является одним из самых важных в Android проекте. В нем содержится информация о пакетах приложения, компонентах типа Activity, Service (и других пока что не знакомых нам компонентах, о которых я расскажу в следующих статьях).

Файл AndroidManifest.xml выполняет следующие задачи:

* предоставляет разрешения приложению на использование или доступ к другим компонентам системы.
* определяет как будут запускаться, например, Activity (какие фильтры использовать).

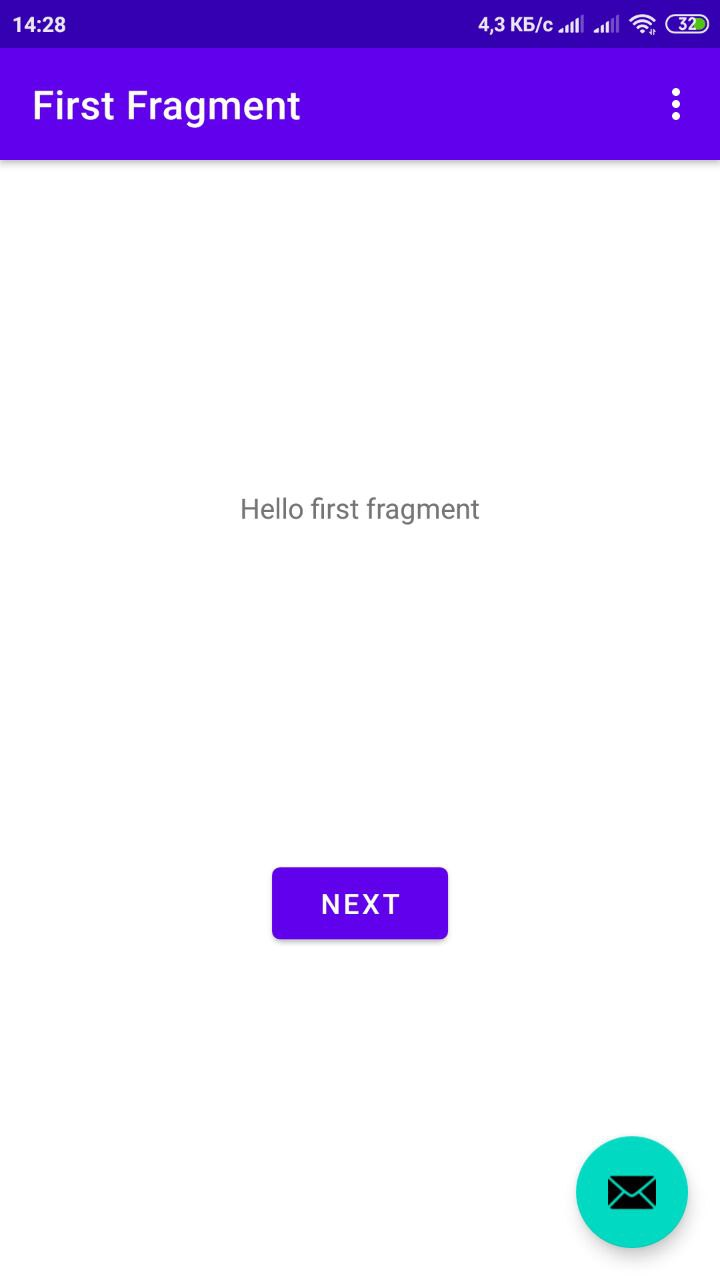
**d. strings.xml**

В файле strings.xml находится текст всех строчек приложения. При помощи него можно редактировать нужную информацию.

1. Найдите в справочном материале и кратко опишите своими словами назначение и содержание каждого из этих файлов.
2. Познакомьтесь с присутствующими в проекте ресурсами.
3. Запустите этот проект на виртуальном устройстве Android. Если виртуальных устройств не существует, создайте.

Запустил на своем телефоне в режиме отладки.



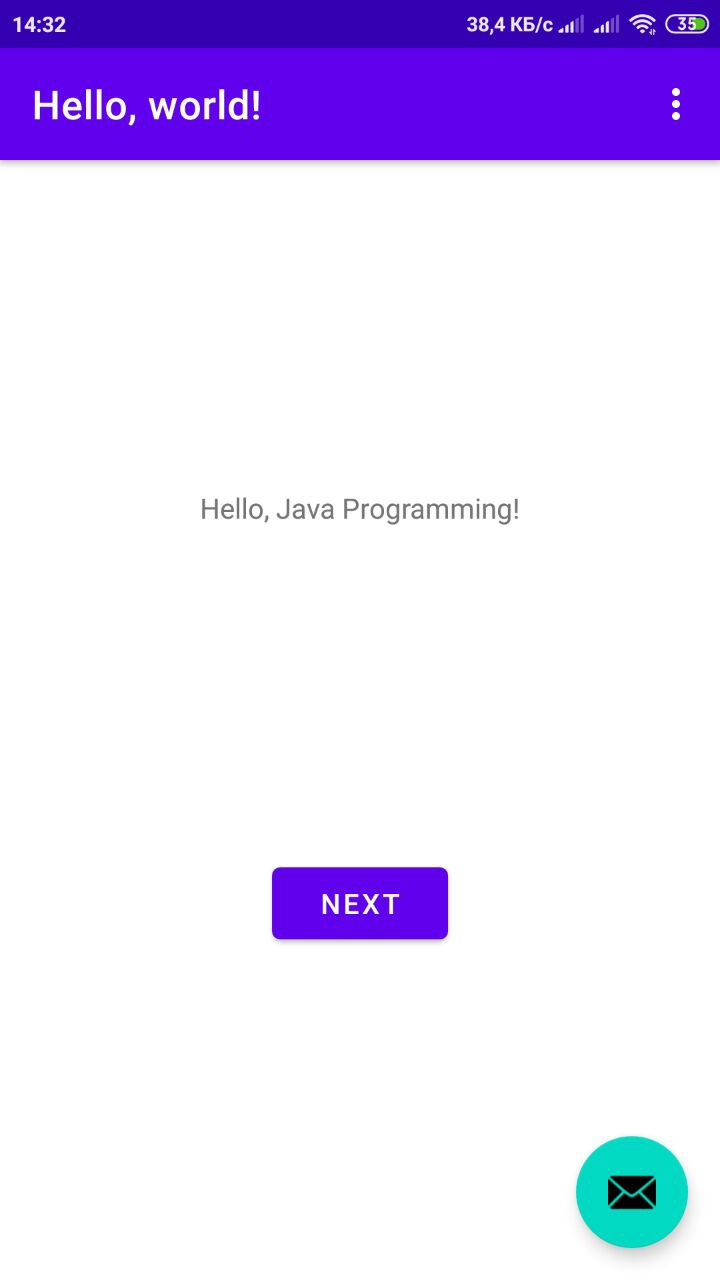


1. Модифицируйте проект таким образом, чтобы в названии приложения выводилась Ваша фамилия, а в строке на главном экране - любая другая строка по Вашему желанию.

Изображение выглядит как текст

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, визитка

Автоматически созданное описание

1. Запустите данное приложение на нескольких виртуальных устройствах с разными версиями API и разными размерами и разрешениями экрана.

Изображение выглядит как текст

Автоматически созданное описаниеИзображение выглядит как текст, визитка, снимок экрана

Автоматически созданное описание

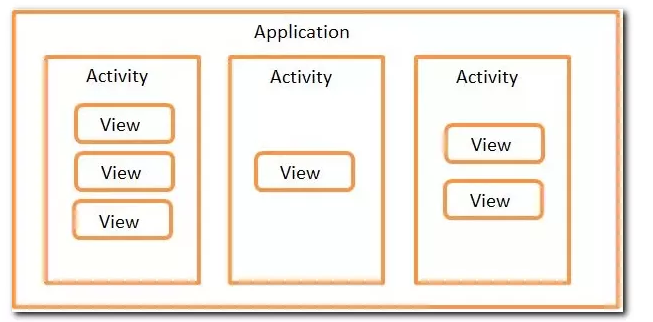
**Контрольные вопросы**

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.

1. **Что такое View и зачем оно нужно?**

Содержимое Activity формируется из различных компонентов, называемых View. Самые распространенные View — это кнопка, поле ввода, чекбокс и т.д.



1. **Что такое Layout и зачем оно нужно?**

Компоновка (также используются термины разметка или макет) хранится в виде XML-файла в папке /res/layout. Это сделано для того, чтобы отделить код от дизайна, как это принято во многих технологиях (HTML и CSS). Кроме основной компоновки для всего экрана, существуют дочерние элементы компоновки для группы элементов. По сути, компоновка – это некий визуальный шаблон для пользовательского интерфейса вашего приложения, который позволяет управлять элементами управления, их свойствами и расположением.

1. **Почему все строки в Layout указываются косвенно из файла с ресурсами?**

Понимание, что используемый в программе текст — это отдельный ресурс, такой же как изображения и звук, приходит на сразу. Но стоит несколько раз поменять имя программы в паре десятков файлов или заняться исправлением однотипной синтаксической ошибки в пяти, шести разных местах и необходимость хранить строки отдельно от кода становиться очевидной.

1. **Что такое версия/уровень API?**

Расшифровывается как Application Programming Interface (программный интерфейс приложения). Это просто интерфейс, уровень абстракции, который обеспечивает связь между двумя разными «частями» программного обеспечения. Он работает как договор между поставщиком (например, библиотекой) и потребителем (например, приложением).

Это набор формальных определений, таких как классы, методы, функции, модули, константы, которые могут использоваться другими разработчиками для написания своего кода. При этом API не включает в себя реализацию.

Уровень API — это целочисленное значение, однозначно идентифицирующее версию API фреймворка, предлагаемую платформой Android.

**Дополнительные задания**

1. Замените RelativeLayout в главном окне приложения на LinearLayout с вертикальным расположением элементов.

Изображение выглядит как текст

Автоматически созданное описание+ строчка android:orientation=”vertical”

1. Присвойте строке, отображаемой в TextView идентификатор следующим атрибутом:

android:id=”@+id/main\_textview”

Изображение выглядит как текст

Автоматически созданное описание

1. Измените название ресурса строки на “text1”

Изображение выглядит как текст

Автоматически созданное описание

1. 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!”);

Изображение выглядит как текст

Автоматически созданное описание

