**Создание графического приложения**

Графический интерфейс пользователя основывается на иерархии объектов View и ViewGroup. Объекты View представляют виджеты, такие как кнопки или текстовые поля. Объекты ViewGroup представляют собой контейнеры для виждетов, управляют их расположением и компоновкой.

Итак, создадим новый проект приложения по тому способу, который мы использовали ранее. Либо используем предыдущий проект.

**Создание линейной разметки**

Выбранный при создании проекта для activity шаблон EmptyActivity добавляет все определение графического интерфейса в файл activity\_main.xml, которые находятся в проекте в папке res/layout, поэтому изменим файл activity\_main.xml следующим образом:

<?xml version="1.0" encoding="utf-8"?>

<LinearLayout

    xmlns:android="http://schemas.android.com/apk/res/android"

    android:layout\_width="match\_parent"

    android:layout\_height="match\_parent"

    android:orientation="horizontal" >

</LinearLayout>

Здесь определяется элемент LinearLayout, который является подклассом ViewGroup и который располагает дочерние элементы в горизонтальный или вертикальный ряд. Ориентация элементов указывается в атрибуте android:orientation. Каждый элемент в контейнере LinearLayout отображается на экране в том порядке, в каком он объявлен в файле XML.

Другие два атрибута - android:layout\_width и android:layout\_height требуются для всех виджетов для определения размеров. Поскольку LinearLayout является корневым элементом компоновки, он должен заполнить все пространство экрана, поэтому для его высоты и ширины указывается значение "match\_parent". Данное значение растягивает виджет до границ родителького контейнера.

### Добавление текстового поля

### Для добавления текстового поля внутри элемента LinearLayout создадим элемент EditText. Как и для любого объекта View, для EditText также надо объявить определенные xml-атрибуты:

<EditText android:id="@+id/edit\_message"

    android:layout\_width="wrap\_content"

    android:layout\_height="wrap\_content"

    android:layout\_weight="1"

    android:hint="Введите сообщение" />

Итак, мы определили следующие атрибуты:

* android:id: обеспечивает уникальный идентификатор виджета, по которому мы можем ссылаться на объект

Знак (@) для ссылки на ссылочный объект в файле XML. После него идет тип ресурса (в данном случае id), слеш и затем имя ресурса (edit\_message).

Знак плюса (+) перед типом ресурса необходимо, когда в первый раз определяется ID ресурса. При компиляции приложения SDK использует имя ID для создания нового ресурса ID в файле gen/R.java. После этого больше не требуется употреблять знак плюса.

* android:layout\_width и android:layout\_height: для этих свойств устнавливаем значение wrap\_content, которое задаст для виджетов величины, достаточные для отображения в контейнере
* android:layout\_weight: позволяет определить занимаемую полем ширину. Значение 1 в данном случае позволяет растянуть поле на всю ширину.
* android:hint: указывает на текст, который будет отображаться в текстовом поле по умолчанию, когда оно пустое.

### Добавление кнопки

Теперь добавим в файл activity\_main.xml кнопку - элемент Button сразу после элемента EditText:

<Button

    android:layout\_width="wrap\_content"

    android:layout\_height="wrap\_content"

    android:text="Отправить" />

В случае с кнопкой ее высота и ширина также имеют значение wrap\_content, поэтому кнопка будет иметь те размеры, которые достаточны для вывода на ней ее текста. Для кнопки не нужно указывать атрибут android:id, поскольку мы не будем на нее ссылаться в коде MainActivity

В итоге файл будет иметь следующий вид:

<?xml version="1.0" encoding="utf-8"?>

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"

    android:layout\_width="match\_parent"

    android:layout\_height="match\_parent"

    android:orientation="horizontal" >

    <EditText android:id="@+id/edit\_message"

        android:layout\_width="wrap\_content"

        android:layout\_height="wrap\_content"

        android:layout\_weight="1"

        android:hint="Введите сообщение" />

    <Button

        android:layout\_width="wrap\_content"

        android:layout\_height="wrap\_content"

        android:text="Отправить" />

</LinearLayout>

И на данный момент приложение будет иметь следующий интерфейс:

