

Санкт-Петербургский политехнический университет Петра Великого
Институт компьютерных наук и технологий
Высшая школа интеллектуальных систем и суперкомпьютерных технологий

Лабораторная работа № 1

Дисциплина: Проектирование мобильных приложений

Тема: layouts

Выполнил студент гр. 3530901/90201 _____ С.А. Федоров
(подпись)

Принял старший преподаватель _____ А.Н. Кузнецов
(подпись)

“ ____ ” _____ 2021 г.

Санкт-Петербург

2021

Оглавление

Цели	3
Задачи	3
Введение	4
LinearLayout	5
Задача 1_9	6
Задача 1_22	11
Задача 1_9_alt	14
ConstraintLayout	16
Задача 2_9	16
Задача 2_22	18
Задача 3_22	19
Выводы	25
Список источников	26

Цели

- Познакомиться со средой разработки Android Studio
- Изучить основные принципы верстки layout с использованием XML
- Изучить основные возможности и свойства LinearLayout
- Изучить основные возможности и свойства ConstraintLayout

Задачи

- Изучить основные layout ресурсы, а именно LinearLayout и ConstraintLayout при помощи документации:
<https://developer.android.com/>
- Создать в соответствии с индивидуальным вариантом layout ресурсы при помощи LinearLayout, используя документацию:
<https://developer.android.com/guide/topics/ui/layout/linear>
- Создать в соответствии с индивидуальным вариантом layout ресурсы при помощи ConstraintLayout, используя документацию:
<https://developer.android.com/training/constraint-layout>
- Ответить на вопрос: в каких случаях целесообразно использовать LinearLayout, в каких ConstraintLayout?

Введение

Layout или же компоновка – макет, который определяет структуру пользовательского интерфейса в приложении. Все элементы макета построены с использованием иерархии View и ViewGroup объектов.

View – это класс, который является базовым контейнером для компонентов пользовательского интерфейса. View обычно рисует то, что видит пользователь и с чем можем взаимодействовать (так называемые виджеты).

ViewGroup – это невидимый контейнер, который определяет структуру макета для View и других объектов ViewGroup (основа для всех layout).

LinearLayout

LinearLayout представляет собой группу представлений, которая выравнивает всех дочерних элементов в одном направлении (вертикально или горизонтально). Также можно указать направление макета при помощи атрибута *android:orientation="horizontal/vertical"*.

Все дочерние LinearLayout элементы располагаются друг за другом, поэтому в вертикальном списке будет только один дочерний элемент на строку, независимо от их ширины, а горизонтальный список будет иметь высоту только в одну строку.

Основные атрибуты:

- **Layout_width** – задает ширину используемого виджета/элемента
 - Можно указывать явное число dp
 - Использование *match_parent* – максимально возможная ширина (т.к. рассматриваем *width*) в пределах родителя
 - Использование *wrap_content* – размер определяется по содержимому элемента (сколько требуется места)
- **Layout_height** – задает высоту виджета
 - Значения можно задавать такие же, как и у *width*
- **Layout_weight** – для создания линейного макета, в котором каждый дочерний элемент использует одинаковое количество места на экране
 - Если *orientation* задано *vertical*, то *height = 0dp*, а *weight = 1*
 - Если *orientation* задано *horizontal*, то *width = 0dp*, а *weight = 1*
- **Layout_gravity** – задает местоположение/гравитацию элемента относительно его родителя
 - Разновидности: Center, top, bottom, end, start, left, right и др.
- **Gravity** – местоположение/гравитация содержимого используемого элемента (например, где будет находиться текст внутри TextView)

- **Layout_margin** – создание отступа с какой-либо стороны или с нескольких
 - Указывается в dp
 - Разновидности: top, bottom, end, start, left, right, horizontal, vertical

Задача 1_9

Требуется создать layout ресурс для макета экрана, представленного на Рис.1 с использованием LinearLayout.

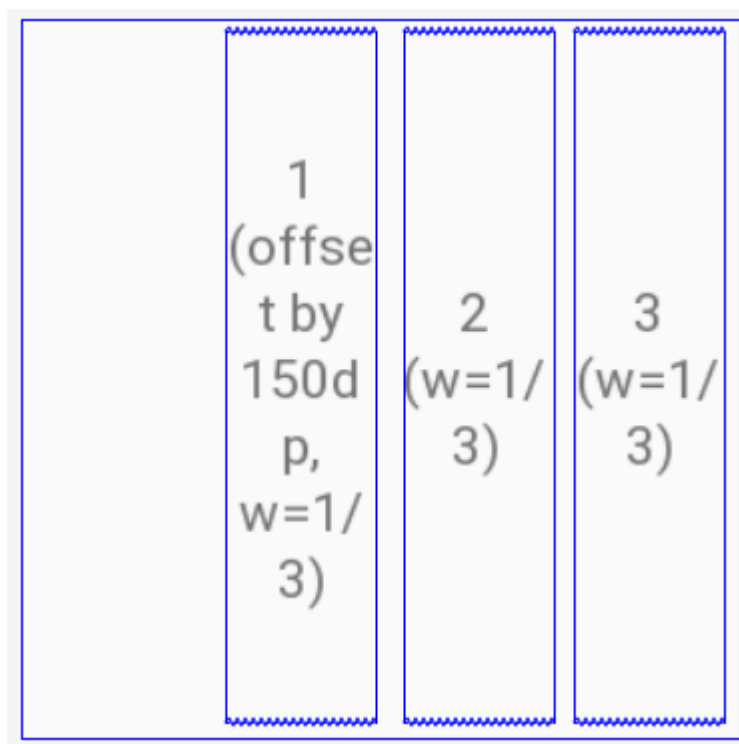


Рис.1 Макет для задачи 1_9

Компоновка или же макет – архитектура расположения элементов интерфейса пользователя для конкретного окна.

Способы создания компоновки:

- Объявить элементы пользовательского интерфейса в XML-файле
- Создать компоновку окна в коде программе (код на языке Kotlin) во время выполнения (Compose)

Для компоновки используем XML-файл:

Листинг 1. Задача 1_9

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="horizontal"
    tools:context=".MainActivity">

    <TextView
        android:layout_width="0dp"
        android:layout_height="match_parent"
        android:layout_weight="1"
        android:layout_marginStart="150dp"
        android:layout_gravity="end"

        android:text="Hello World!"
        android:gravity="center"
        android:textSize="17sp"
        android:background="#ffa500"
        android:textColor="#000"/>

    <Button
        android:id="@+id/button5"
        android:layout_width="0dp"
        android:layout_height="match_parent"
        android:layout_weight="1"

        android:backgroundTint="#0f0"
        android:gravity="center"
        android:text="@string/click_me"
        android:textColor="#616161" />

    <ImageView
        android:id="@+id/imageView3"
        android:layout_width="0dp"
        android:layout_height="match_parent"
        android:layout_weight="1"

        android:contentDescription="@string/todo"
        tools:srcCompat="@tools:sample/avatars"
        android:gravity="center"
        android:background="#4b0082"/>

</LinearLayout>
```

Корневой элемент компоновки – это `LinearLayout` с параметрами:

- *android:layout_width* – параметр ширины компонента, т.к. нам требуется создать макет на весь экран, то укажем значение *match_parent* (максимально возможная ширина/высота в пределах родителя)
- *android:layout_height* – параметр высоты компонента, по аналогии с *layout_width* зададим значение *match_parent*
- *android:orientation="horizontal"* – атрибут, указывающий на направление макета, как будут располагаться элементы внутри (горизонтально или вертикально), в нашем случае требуется *horizontal*
- *tools:context* – атрибут корневого элемента, позволяющий определить связь между макетом и классом активности (`MainActivity`), в который данный макет будет реализован

Далее создадим дочерние элементы в соответствии в макетом:

- `TextView` – виджет для отображения текста без возможности редактирования его пользователем
 - *layout_margin* выберем `start` со значением `150dp`, т.к. требуется сделать отступ слева на `150dp`
 - *layout_width* поставим значение `0`, следовательно, *layout_weight* надо указать `1`, так как по макету требуется создать вертикальные полосы одинаковой ширины (у всех остальных элементов укажем такие же параметры)
 - *layout_height* поставим *match_parent*, чтобы созданный вертикальный элемент занимал всю высоту экрана
 - Укажем параметр элемента *text* со значением “Hello World!”
 - Чтобы *text* располагался по центру виджета укажем *gravity* со значением `center`

- Чтобы *text* был виден внутри виджета выберем ему цвет при помощи *textColor*
- Размер текста укажем при помощи *textSize* со значением 17sp
- Чтобы виджет `TextView` был выделялся на экране укажем параметр *background* с произвольным цветом
- `Button` – элемент кнопка
 - *android:id* – уникальное имя ресурса для элемента, которое можно использовать для получения ссылки на объект `ViewGroup` из приложения
 - *layout_width*, *layout_weight*, *layout_height* укажем такие же, как и у `TextView`
 - Далее зададим параметр *backgroundTint*, чтобы кнопка была некоторого цвета и выделялась на экране
 - Напишем *text* на кнопке, выберем цвет этого текста *textColor* и расположение *gravity*
- `ImageView` – виджет для отображения изображений
 - Укажем *android:id*, а также зададим параметры *layout_width*, *layout_weight*, *layout_height*
 - Установим расположение нашего изображения при помощи *gravity*, а также заполним оставшееся место виджет цветом при помощи *background*
 - *android:contentDescription* – описание картинки
 - *android:src* – путь к изображению для его загрузки

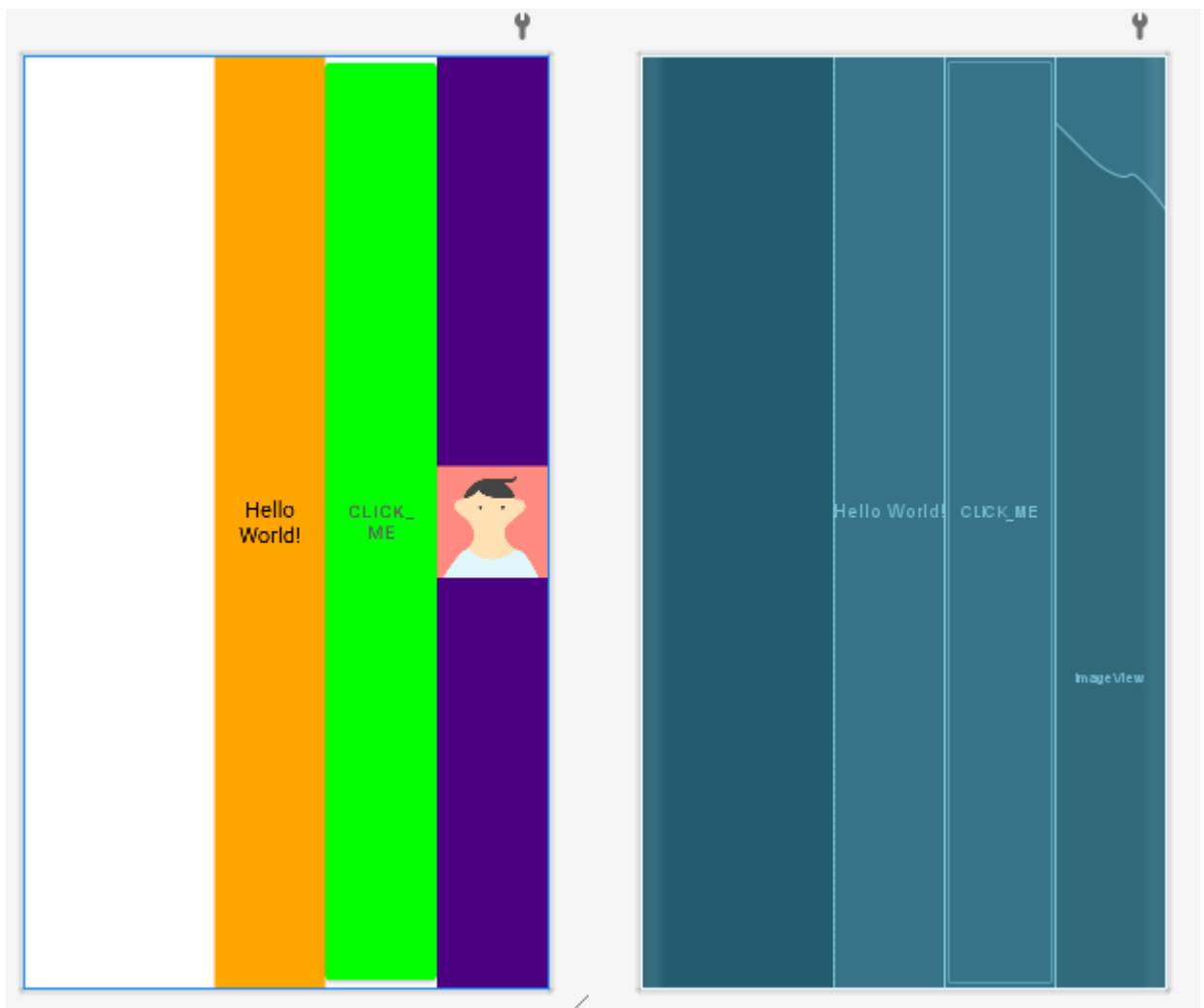


Рис.2 Получившийся layout

Задача 1_22

Требуется создать layout ресурс для макета экрана, представленного на Рис.3 с использованием LinearLayout.

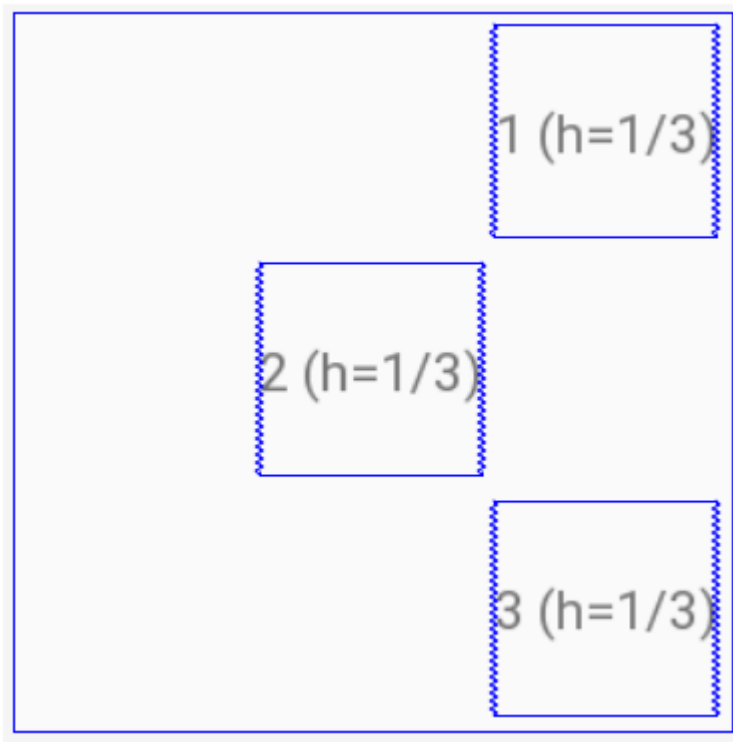


Рис.3 Макет для задачи 1_22

Для компоновки используем XML-файл:

Листинг 2. Задача 1_22

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".MainActivity">

    <TextView
        android:layout_width="136dp"
        android:layout_height="0dp"
        android:layout_weight="1"
        android:layout_gravity="end"

        android:text="Hello World!"
        android:gravity="center"
        android:textSize="25sp"
        android:background="#ffa500"
        android:textColor="#000"/>

    <Button
```

```

        android:id="@+id/button5"
        android:layout_width="136dp"
        android:layout_height="0dp"
        android:layout_gravity="center"
        android:layout_weight="1"

        android:backgroundTint="#0f0"
        android:gravity="center"
        android:text="@string/click_me"
        android:textColor="#616161" />

<ImageView
    android:id="@+id/imageView3"
    android:layout_width="136dp"
    android:layout_height="0dp"
    android:layout_weight="1"
    android:layout_gravity="end"

    tools:srcCompat="@tools:sample/avatars"
    android:gravity="center"
    android:background="#4b0082"
    android:contentDescription="@string/todo" />
</LinearLayout>

```

Корневой элемент компоновки – это `LinearLayout` с параметрами:

- `android:orientation="vertical"`, т.к. по макету требуется вертикальная ориентация

Дочерние элементы:

- По макету требуется создать виджеты одинаковой высоты для заполнения всей высоты экрана. Для этого выставим равный вес и параметр `layout_height` со значением `0dp`
- По макету требуется расположить виджеты 1 и 3 в правом углу, а виджет 2 – по центру. Для этого воспользуемся атрибутом `layout_gravity` со значением `end` для 1 и 3, а для 2 - `center`
- Чтобы получившийся макет максимально соответствовал задаче (Рис.2) у всех виджетов был выставлен параметр `layout_width` со значением `136dp` (примерно 1/3 ширины экрана)
- Все остальные параметры дочерних элементов были уже описаны в пункте [Задача 1 9](#)

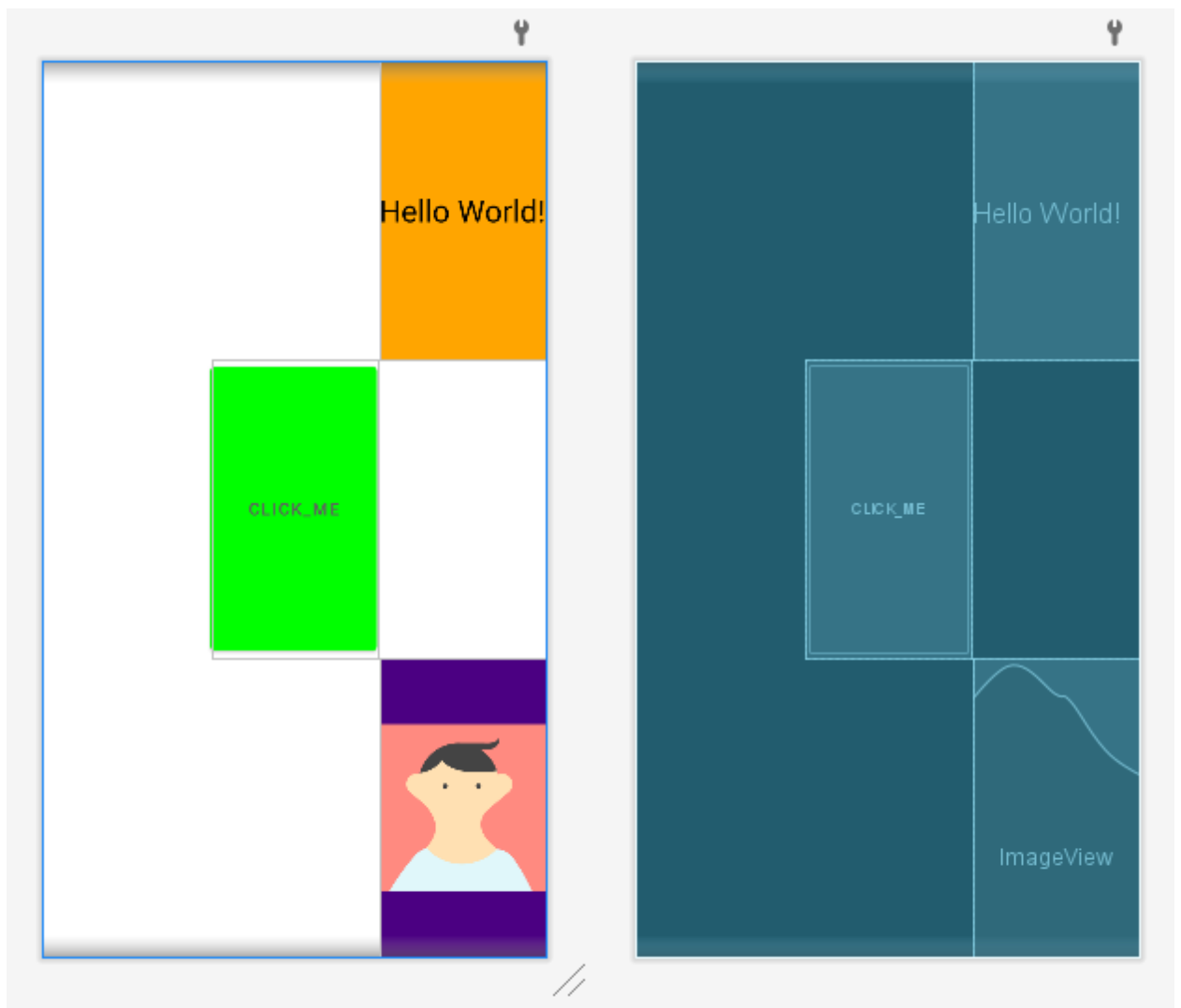


Рис.4 Получившийся layout

Задача 1_9_alt

В данном пункте применим альтернативное решение для [Задача 1_9](#)

Здесь, чтобы добиться расположения виджетов 1, 2 и 3 друг за другом, причем с отступ от левого края на 150dp вместо атрибута *layout_marginStart* для первого виджета, можно воспользоваться виджетом *Space*. *Space* – это виджет, предназначенный для создания отступов между элементами.

Листинг 3. Задача 1_22_alt

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="horizontal"
    tools:context=".MainActivity">

    <Space
        android:layout_width="150dp"
        android:layout_height="match_parent"/>

    <TextView
        android:layout_width="0dp"
        android:layout_height="match_parent"
        android:layout_weight="1"
        android:layout_gravity="end"

        android:text="Hello World!"
        android:gravity="center"
        android:textSize="17sp"
        android:background="#ffa500"
        android:textColor="#000"/>

    <Button
        android:id="@+id/button5"
        android:layout_width="0dp"
        android:layout_height="match_parent"
        android:layout_gravity="center"
        android:layout_weight="1"

        android:backgroundTint="#0f0"
        android:gravity="center"
        android:text="@string/click_me"
        android:textColor="#616161" />

    <ImageView
        android:id="@+id/imageView3"
        android:layout_width="0dp"
        android:layout_height="match_parent"
        android:layout_weight="1"
        android:layout_gravity="end"

        android:contentDescription="@string/todo"
        tools:srcCompat="@tools:sample/avatars"
        android:gravity="center"
```

```
android:background="#4b0082"/>  
</LinearLayout>
```

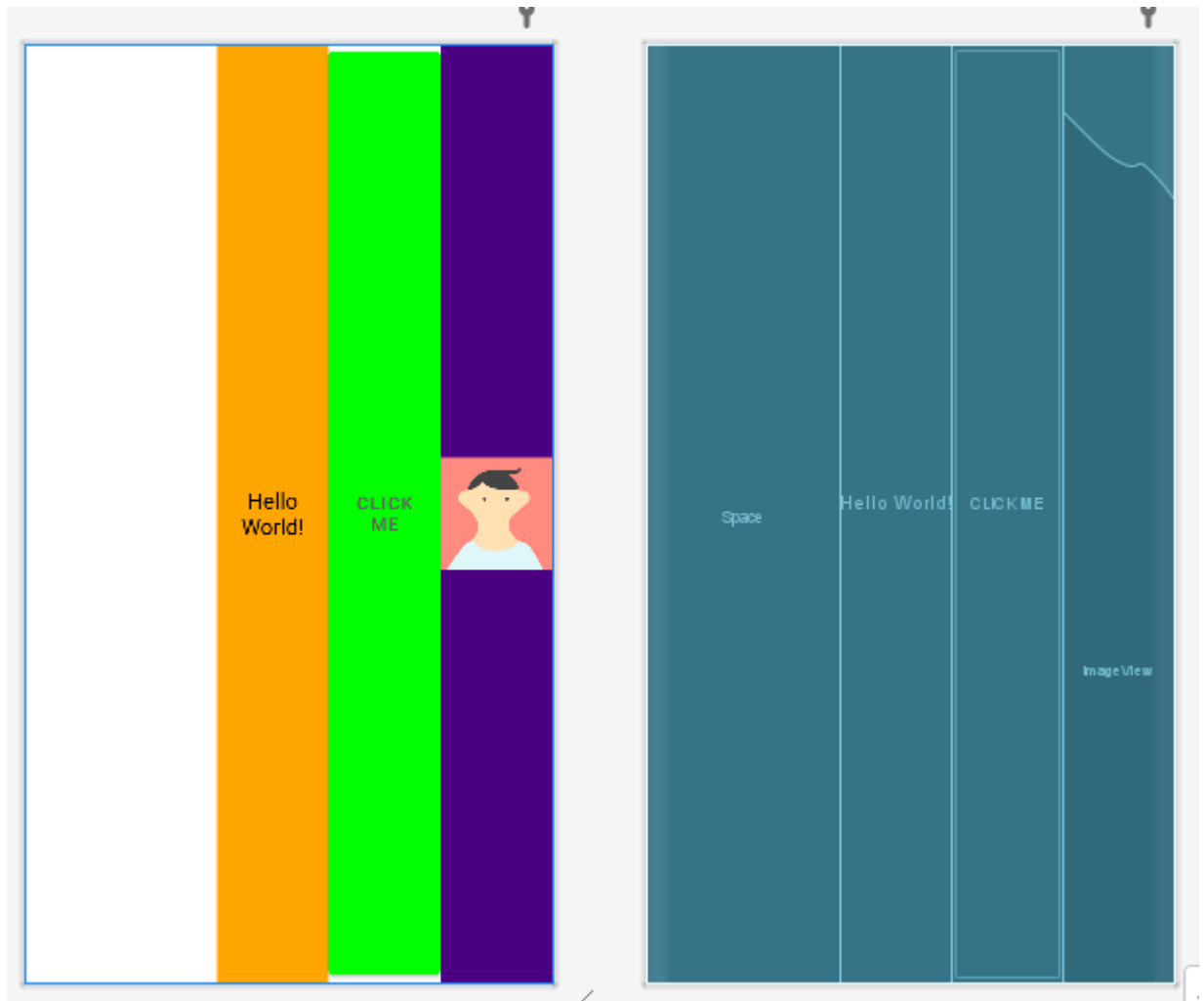


Рис.5 Получившийся layout

Исходя из Рис.5 можно сделать вывод, что получилась точно такая же компоновка, как и на Рис. 2.

ConstraintLayout

Вид компоновки, который позволяет создавать большие и сложные макеты с плоской иерархией представлений (без вложенных групп представлений), то есть, появляется возможность более гибкого позиционирования компонентов между собой.

Для этого в *ConstraintLayout* предусмотрены атрибуты, связанные с взаимным расположением объектов, а именно одной из сторон текущего элемента (top, bottom, left или right) с стороной другого:

- *layout_constraintBottom_toBottomof="parent"* – эта команда означает то, что нижнюю сторону текущего элемента соединяем с нижней частью parent (если родитель – это наш экран или элемент, занимающий весь макет, то текущий элемент будет привязан к нижней части макета)
- Также при использовании *ConstraintLayout* требуется создавать идентификаторы компонентов, чтобы указывать их как значение элемента, которому идет привязка (в примере выше идентификатор – это parent).

Примечание: чтобы определить положение представления в *ConstraintLayout*, требуется добавить по крайней мере одно горизонтальное и одно вертикальное ограничение представления.

Задача 2_9

Требуется выполнить [Задача 1_9](#) при помощи *ConstraintLayout*

Листинг 4. Задача 2_9

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
```



```

<TextView
    android:id="@+id/textView2"
    android:layout_width="0dp"
    android:layout_height="match_parent"

    android:layout_marginStart="150dp"
    android:background="#ffa500"
    android:gravity="center"
    android:text="Hello World!"
    android:textColor="#000"

    android:textSize="17sp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toStartOf="@+id/button"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_weight="1"/>

<Button
    android:id="@+id/button"
    android:layout_width="0dp"

    android:layout_height="match_parent"
    android:backgroundTint="#0f0"
    android:gravity="center"
    android:text="@string/click_me"
    android:textColor="#616161"

    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toStartOf="@+id/imageView"
    app:layout_constraintStart_toEndOf="@+id/textView2"
    app:layout_constraintTop_toTopOf="@+id/textView2"
    app:layout_constraintVertical_weight="1"/>

<ImageView
    android:id="@+id/imageView"
    android:layout_width="0dp"
    android:layout_height="match_parent"
    android:background="#4b0082"

    android:contentDescription="@string/todo"
    android:gravity="center"

    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toEndOf="@+id/button"
    app:layout_constraintTop_toTopOf="@+id/button"
    app:layout_constraintVertical_weight="1"
    tools:srcCompat="@tools:sample/avatars" />

</androidx.constraintlayout.widget.ConstraintLayout>

```

Исходя из Листинга 4, видно, что для каждого дочернего элемента указано расположение относительно другого элемента. Например, элемент `TextView` соединен сверху и снизу с родителем, справа с элементов `Button`, а слева соединен с родителем на отступе в `150dp`.

Задача 2_22

Требуется выполнить [Задача 1_22](#) при помощи ConstarintLayout

Листинг 5. Задача 2_22

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <TextView
        android:id="@+id/textView2"
        android:layout_width="136dp"

        android:layout_height="0dp"
        android:background="#ffa500"
        android:gravity="center"
        android:text="Hello World!"
        android:textColor="#000"

        android:textSize="25sp"
        app:layout_constraintBottom_toTopOf="@+id/button"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_weight="1" />

    <Button
        android:id="@+id/button"
        android:layout_width="136dp"

        android:layout_height="0dp"
        android:backgroundTint="#0f0"
        android:gravity="center"
        android:text="@string/click_me"
        android:textColor="#616161"

        app:layout_constraintBottom_toTopOf="@+id/imageView"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/textView2"
        app:layout_constraintVertical_weight="1" />

    <ImageView
        android:id="@+id/imageView"
        android:layout_width="137dp"
        android:layout_height="0dp"
        android:background="#4b0082"

        android:contentDescription="@string/todo"
        android:gravity="center"

        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/button"
        tools:srcCompat="@tools:sample/avatars"
        app:layout_constraintVertical_weight="1"/>

</androidx.constraintlayout.widget.ConstraintLayout>
```

Задача 3_22

Требуется создать layout ресурс для макета экрана на Рис.6

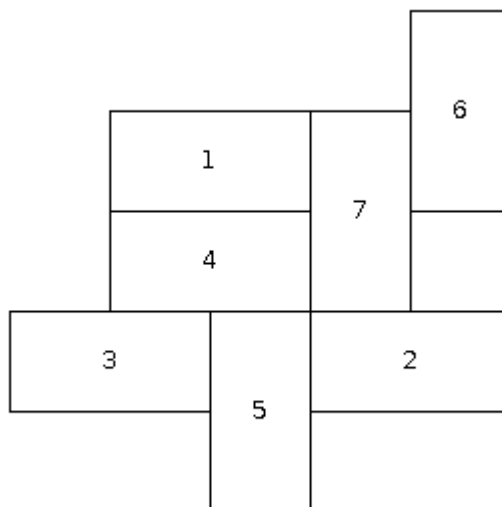


Рис.6 Макет для задачи 3_22

Воспользуемся ConstraintLayout, так как макет довольно сложный и имеет множество связей (соединений или привязок).

Листинг 6. Задача 3_22

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <androidx.constraintlayout.widget.ConstraintLayout
        android:layout_width="0dp"
        android:layout_height="0dp"
        android:background="#ffa500"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintDimensionRatio="1:1"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent">

        <androidx.constraintlayout.widget.Guideline
            android:id="@+id/line1"
            android:layout_width="0dp"
            android:layout_height="0dp"
            android:orientation="vertical"
            app:layout_constraintGuide_percent="0.2" />

        <androidx.constraintlayout.widget.Guideline
            android:id="@+id/line2"
            android:layout_width="0dp"
            android:layout_height="0dp"
```

```

        android:orientation="horizontal"
        app:layout_constraintGuide_percent="0.2" />

<androidx.constraintlayout.widget.Guideline
    android:id="@+id/line3"
    android:layout_width="0dp"
    android:layout_height="0dp"
    android:orientation="horizontal"
    app:layout_constraintGuide_percent="0.4" />

<androidx.constraintlayout.widget.Guideline
    android:id="@+id/line4"
    android:layout_width="0dp"
    android:layout_height="0dp"
    android:orientation="horizontal"
    app:layout_constraintGuide_percent="0.6" />

<androidx.constraintlayout.widget.Guideline
    android:id="@+id/line5"
    android:layout_width="0dp"
    android:layout_height="0dp"
    android:orientation="horizontal"
    app:layout_constraintGuide_percent="0.8" />

<Switch
    android:id="@+id/switch2"
    android:layout_width="0dp"
    android:layout_height="0dp"

    android:background="#8BC34A"
    android:gravity="center"
    android:text="@string/_2"
    android:textColor="@color/black"
    android:textSize="25sp"

    app:layout_constraintBottom_toTopOf="@+id/line5"

    app:layout_constraintDimensionRatio="2:1"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintTop_toTopOf="@+id/line4"
    tools:ignore="UseSwitchCompatOrMaterialXml" />

<CheckBox
    android:id="@+id/checkBox"
    android:layout_width="0dp"
    android:layout_height="0dp"

    android:background="#E91E63"
    android:gravity="center"
    android:text="@string/_5"
    android:textColor="@color/black"
    android:textSize="25sp"

    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintDimensionRatio="1:2"
    app:layout_constraintEnd_toStartOf="@+id/switch2"
    app:layout_constraintTop_toTopOf="@+id/line4" />

<ImageView
    android:id="@+id/imageView3"
    android:layout_width="0dp"

```

```

        android:layout_height="0dp"
        android:background="#E1EAE"

        android:contentDescription="@string/todo"
        android:gravity="center"

        app:layout_constraintDimensionRatio="2:1"
        app:layout_constraintBottom_toTopOf="@+id/line5"
        app:layout_constraintEnd_toStartOf="@+id/checkbox"
        app:layout_constraintStart_toStartOf="parent"
        app:srcCompat="@drawable/number_three" />

<TextView
    android:id="@+id/textView6"
    android:layout_width="0dp"
    android:layout_height="0dp"
    android:background="#00f"
    android:gravity="center"

    android:text="@string/_6"
    android:textColor="#9E9E9E"
    android:textSize="25sp"

    app:layout_constraintBottom_toTopOf="@+id/line3"
    app:layout_constraintDimensionRatio="1:2"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintTop_toTopOf="parent" />

<Button
    android:id="@+id/button7"
    android:layout_width="0dp"
    android:layout_height="0dp"

    android:backgroundTint="#9C27B0"
    android:gravity="center"
    android:text="@string/_7"
    android:textColor="@color/black"

    android:textSize="25sp"

    app:layout_constraintBottom_toTopOf="@+id/line4"
    app:layout_constraintDimensionRatio="1:2"
    app:layout_constraintEnd_toStartOf="@+id/textView6"
    app:layout_constraintTop_toTopOf="@+id/line2" />

<RadioButton
    android:id="@+id/radioButton"
    android:layout_width="0dp"
    android:layout_height="0dp"

    android:background="#DAEA0E"
    android:gravity="center"
    android:text="@string/_4"
    android:textColor="@color/black"
    android:textSize="25sp"

    app:layout_constraintBottom_toTopOf="@+id/line4"
    app:layout_constraintDimensionRatio="2:1"
    app:layout_constraintEnd_toStartOf="@+id/button7"
    app:layout_constraintStart_toStartOf="@+id/line1" />

<ToggleButton

```

```

        android:id="@+id/toggleButton1"
        android:layout_width="0dp"
        android:layout_height="0dp"

        android:background="#88737B"
        android:gravity="center"
        android:text="@string/_1"
        android:textColor="#000000"

        android:textSize="25sp"

        app:layout_constraintBottom_toTopOf="@+id/radioButton"
        app:layout_constraintDimensionRatio="2:1"
        app:layout_constraintEnd_toStartOf="@+id/button7"
        app:layout_constraintStart_toStartOf="@+id/line1" />

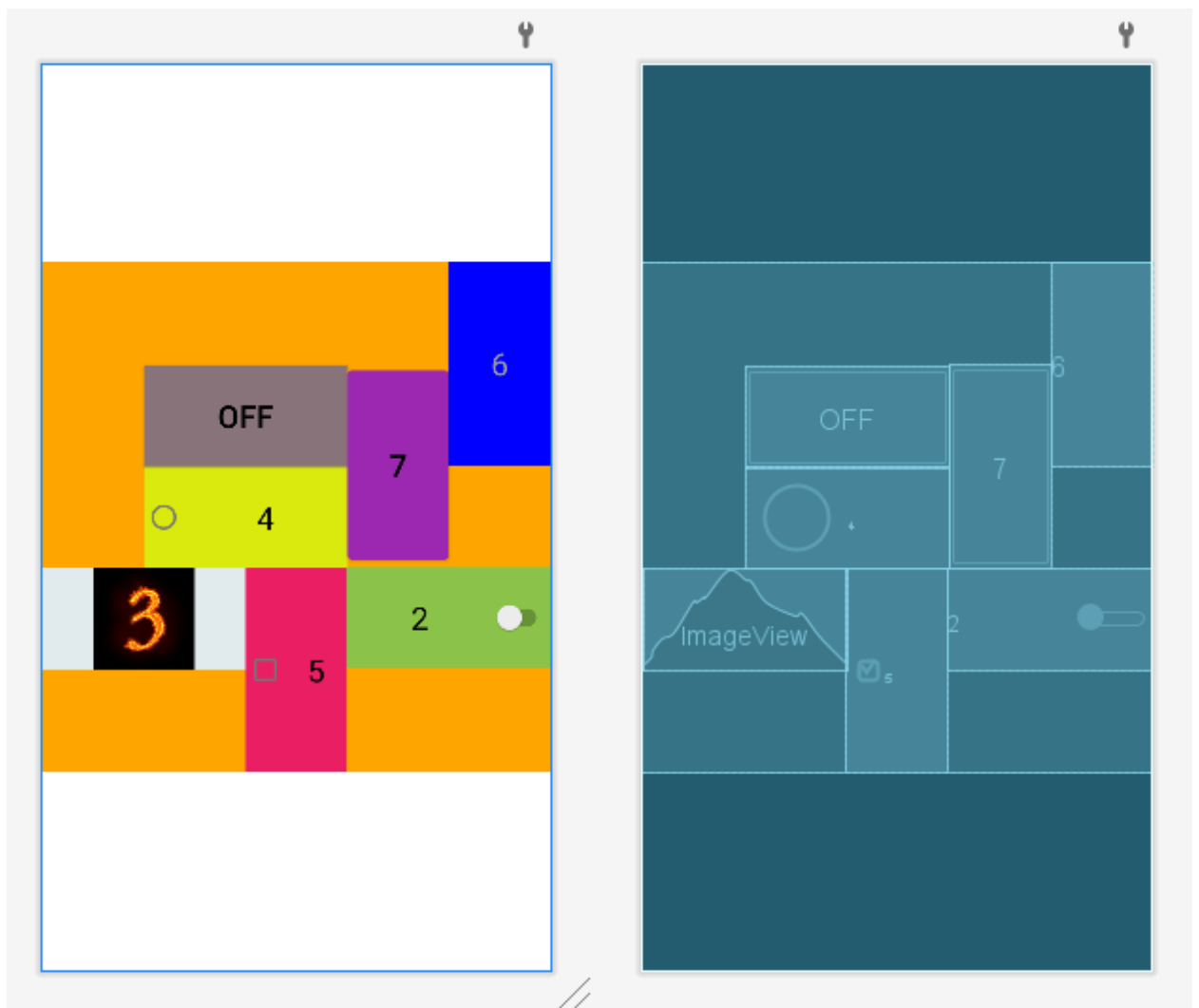
</androidx.constraintlayout.widget.ConstraintLayout>
</androidx.constraintlayout.widget.ConstraintLayout>

```

Был создан основной элемент при помощи *ConstraintLayout*, далее был создан дочерний (вложенный) элемент – квадрат, занимающий максимальную площадь (при помощи атрибута, отвечающего за соотношение сторон) и располагающий по центру экрана, при его создании также использовалась компоновка *ConstraintLayout*. Для улучшения видимости квадрата был выставлен атрибут *background* (цвет квадрата - оранжевый).

Каждый виджет (1-7 по макету), является дочерним элементом созданного квадрата. Размер каждого объекта был вычислен по формуле $A \cdot B / 5$, где A – некоторое число, B – размер стороны квадрата.

Для того, чтобы добиться заданного расположения элементов использовался класс-помощник *GuideLines*. Использовались привязки элементов друг к другу, к родителю и к созданным при помощи *GuideLines* линиям. Теперь для задания размеров виджетов можно использовать атрибут соотношения сторон, тогда данное решение будет выглядеть корректно на любом нестандартном экране.



Выводы

- В ходе выполнения данной лабораторной работы было произведено практическое ознакомление со средой разработки Android приложений Android Studio. Android Studio, также как и IDE является продуктом JetBrains, значит большинство функций для работы с проектом и непосредственно с кодом были уже изучены в ранних курсах в таких средах, как IntelliJ IDEA, CLion, PyCharm. Из новых элементов было изучено окно визуализации XML кода (то, как будет выглядеть макет на различных физических устройствах/экранах) и его атрибуты:
 - изменение физического устройства для представления XML файла
 - изменение ориентации экрана
 - компонент palette, который позволяет проще добавлять новые виджеты.
- Было создано 6 макетов (6 XML-файлов) с использованием различных компоновок согласно индивидуальному варианту
- Были изучены основные принципы верстки layout с использованием XML
- Также изучены основные возможности и свойства двух основных компоновок: LinearLayout и ConstraintLayout

Вопрос: в каких случаях целесообразно использовать LinearLayout, в каких ConstraintLayout?

- LinearLayout следует использовать в тех случаях, когда требуется создать простой интерфейс, который имеет горизонтальную или же вертикальную структуру расположения виджетов. Если в случаях создания простых интерфейсов использовать ConstraintLayout, то придется указывать множество связей, которые будут загромождать код. Также для указания связей потребуются теги, из-за которых XML-файл будет дольше обрабатываться.
- ConstraintLayout следует использовать во всех остальных случаях, когда использование LinearLayout не целесообразно. То есть, когда требуется создать сложный интерфейс, а ConstraintLayout как раз обладает большими возможностями настройки виджетов и их расположения.

Ссылка на github: https://github.com/sergeyfedorov02/Android_labs.git

Список источников

<https://developer.android.com/>

<https://github.com/andrei-kuznetsov/android-lectures>