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

Отчёт по лабораторной работе № 1

Дисциплина: Проектирование мобильных приложений

Тема: Layouts

Выполнил студент гр. 3530901/90201 _____З.А. Фрид
(подпись)

Принял преподаватель _____ А.Н. Кузнецов
(подпись)

“____” _____ 2021 г.

Санкт-Петербург
2021

Репозиторий GitHub:

https://github.com/zina-frid/AndroidLabs/tree/main/projects/lab1

1. 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

2. Задачи

Задача 1. LinearLayout

Создайте layout ресурсы для следующих макетов экрана с использованием LinearLayout. Согласно варианту 23 это изображения 1 и 10 (Рис. 1 и Рис. 2 соответственно).

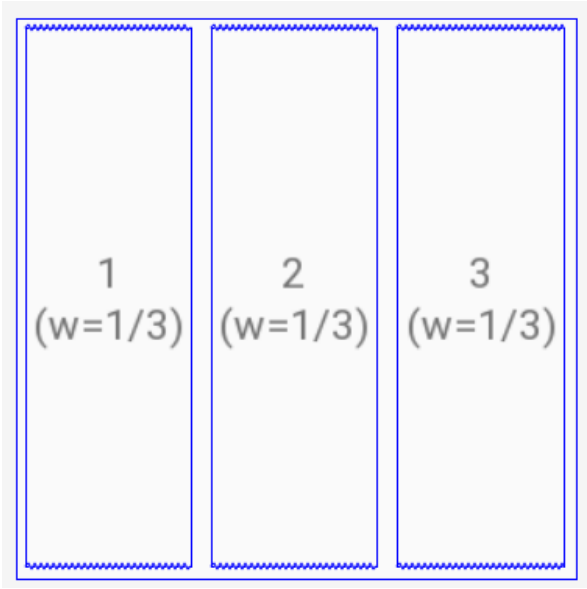

Рис. 1

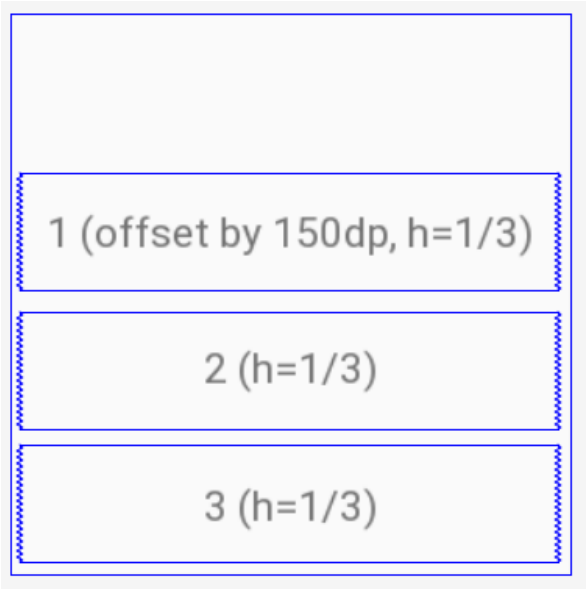

Рис. 2

Задача 2. ConstraintLayout

Решите задачу 1 (обе подзадачи) с использованием ConstraintLayout.

Задача 3. ConstraintLayout

Создайте layout ресурс для следующего макета экрана с использованием ConstraintLayout. Согласно варианту 23 это изображение 23 (Рис. 3).

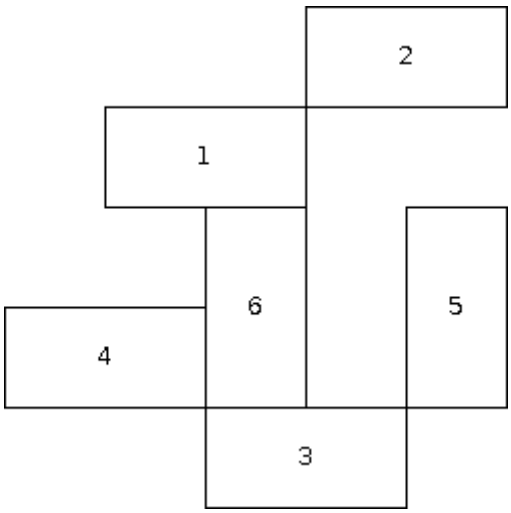

Рис. 3

3. Ход работы

3.1. Задача 1. LinearLayout

Для выполнения задачи была изучена документация о LinearLayout.

Изображение 1 (Рис. 1)

Для данного макета было необходимо создать три виджета, которые в ширину занимают одинаковое пространство. Атрибут *android:layout_width* отвечает за ширину элемента. Атрибут *android:layout_weight* отвечает за то, сколько места занимает элемент, то есть, за вес элемента. Для того, чтобы элементы поровну делили всё доступное пространство, для каждого элемента я выставила значение ширины равное 0dp и вес равным 1.

Листинг 1 XML-код решения задачи 1 (Изображение 1)

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

```
    <!-- task 1, image 1 -->

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

        android:text="@string/widget1"
        android:textAlignment="center"
        android:gravity="center"
        android:textColor="@color/black"
        android:textSize="24sp" />

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

        android:text="@string/widget2"
        android:textSize="20sp" />

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

        android:background="@color/green"
        android:contentDescription="@string/widget4"
        android:src="@android:drawable/ic_menu_delete" />

</LinearLayout>
```

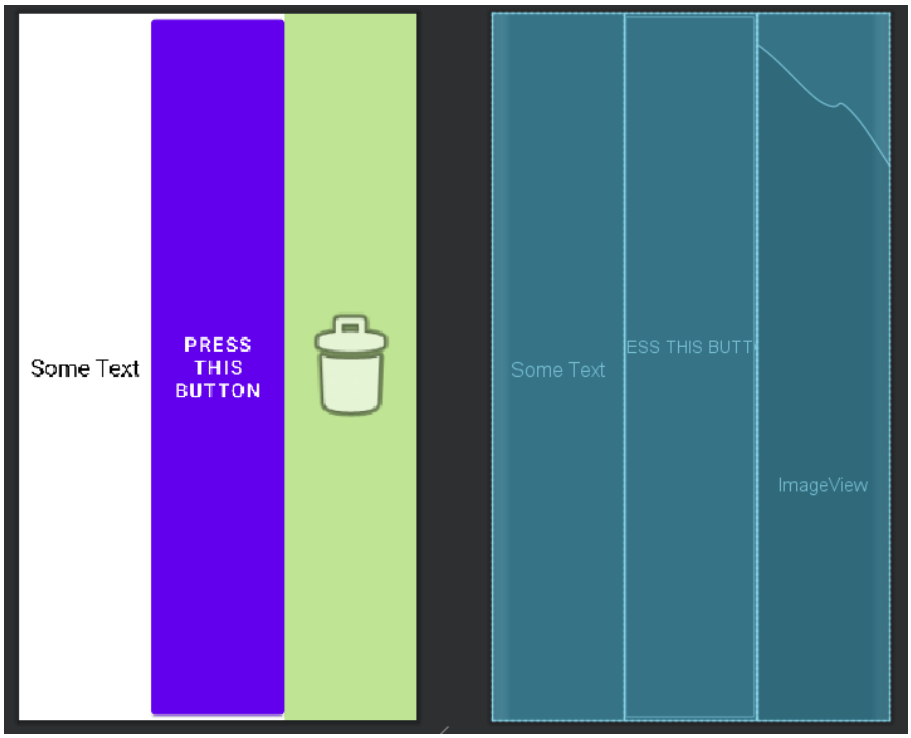

Рис. 4 Визуальное представление решения 1 (Изображение 1)

Изображение 10 (Рис. 2)

Для данного макета необходимо сделать отступ сверху на 150dp, и три элемента, которые, как и в первом макете, занимают пространство в равных соотношениях.

Было разработано два способа решения этой задачи: один с использованием атрибута *android:layout_marginTop*, а второй с помощью дополнительного элемента Space.

Способ 1

Атрибут *android:layout_marginTop* задает внешний отступ в пикселях от верхнего края. Для того, чтобы элементы поровну делили всё оставшееся доступное пространство, для каждого элемента в этот раз я выставила значение высоты равное 0dp, а вес оставила равным 1.

Листинг 2 XML-код решения задачи 1 (Изображение 10, способ 1)

```
<?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">

    <!-- task 1, image 10 -->

    <CheckBox
        android:id="@+id/textView"
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_marginTop="150dp"
        android:layout_weight="1"

        android:text="@string/widget3"
        android:background="@color/green"
        android:textColor="@color/black"
        android:textSize="32sp" />

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

        android:gravity="center"
        android:text="@string/widget2">

       <RadioButton
```

```
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="@string/choice1"
            android:textSize="24sp" />

        <RadioButton
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="@string/choice2"
            android:textSize="24sp" />

        <RadioButton
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="@string/choice3"
            android:textSize="24sp" />

    </RadioGroup>

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

        android:text="@string/widget5"
        android:background="@color/yellow"
        android:gravity="center"
        android:textSize="24sp" />

</LinearLayout>
```

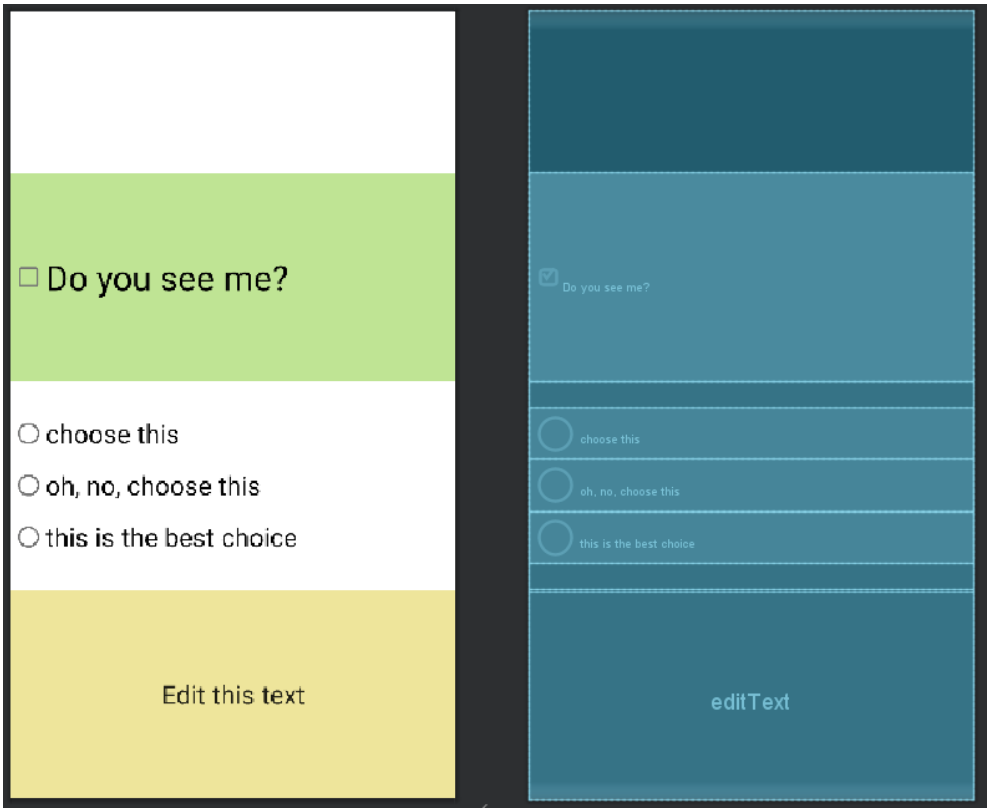

Рис. 5 Визуальное представление решения 1 (Изображение 10, способ 1)

Способ 2

В данном способе вместо внешнего отступа в 150dp был добавлен дополнительный элемент Space с высотой равной150dp. Значения остальных элементов не изменились.

Листинг 3 XML-код решения задачи 1 (Изображение 10, способ 2)

```
<?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">

    <!-- task 1, image 10 alternative-->

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

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

        android:text="@string/widget3"
        android:background="@color/green"
        android:textColor="@color/black"
        android:textSize="32sp" />

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

        android:gravity="center"
        android:text="@string/widget2">

        <RadioButton
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="@string/choice1"
            android:textSize="24sp" />

        <RadioButton
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="@string/choice2"
            android:textSize="24sp" />

        <RadioButton
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="@string/choice3"
            android:textSize="24sp" />
```

```
    </RadioGroup>

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

        android:text="@string/widget5"
        android:background="@color/yellow"
        android:gravity="center"
        android:textSize="24sp" />


</LinearLayout>
```

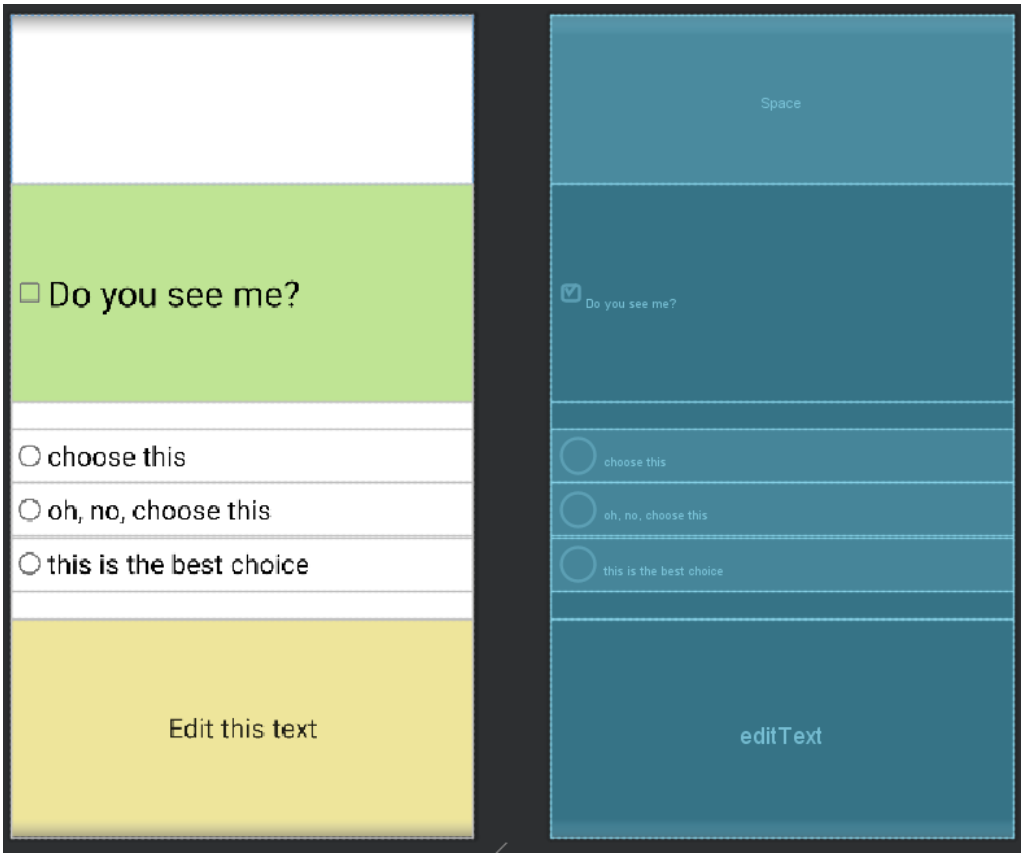

Рис. 6 Визуальное представление решения 1 (Изображение 10, способ 2)

3.2. Задача 2. ConstraintLayout

Для решения этой задачи использовался Constraint Layout. Он позволяет «привязывать» края объекта к другим краям, задавать фиксированные отступы.

Изображение 1 (Рис. 1)

Для решения данной задачи левый, верхний и нижний края левого элемента были привязаны в левому, верхнему и нижнему краям «родителя» соответственно, а правый край в левому краю центрального элемента. Верхний и нижний края центрального элемента привязаны к верхнему и нижнему краям «родителя» соответственно, левый край привязан к правому краю левого элемента, а правый – к левому краю правого элемента. Верхний, нижний и правый края правого элемента привязаны к верхнему, нижнему и правому краям «родителя» соответственно, а левый край привязан к центральному элементу. После выставления атрибутов ширины и высоты в 0dp для всех объектов, они «поделили» пространство экрана поровну. При этом элементы «привязаны» друг к другу и не создают пустого пространства.

Листинг 4 XML-код решения задачи 2 (Изображение 1)

```
<?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">

    <!-- task 2, image 1 -->

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

        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintEnd_toStartOf="@id/button2"

        android:gravity="center"
        android:text="@string/widget1"
        android:textAlignment="center"
        android:textColor="@color/black"
        android:textSize="24sp" />

    <Button
        android:id="@+id/button2"
        android:layout_width="0dp"
        android:layout_height="0dp"


        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintStart_toEndOf="@id/textView2"
```

```
        app:layout_constraintEnd_toStartOf="@id/imageView2"

        android:text="@string/widget2"
        android:textSize="20sp" />

    <ImageView
        android:id="@+id/imageView2"
        android:layout_width="0dp"
        android:layout_height="0dp"

        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintStart_toEndOf="@id/button2"
        app:layout_constraintEnd_toEndOf="parent"

        android:background="@color/green"
        android:contentDescription="@string/widget4"
        android:src="@android:drawable/ic_menu_delete" />

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

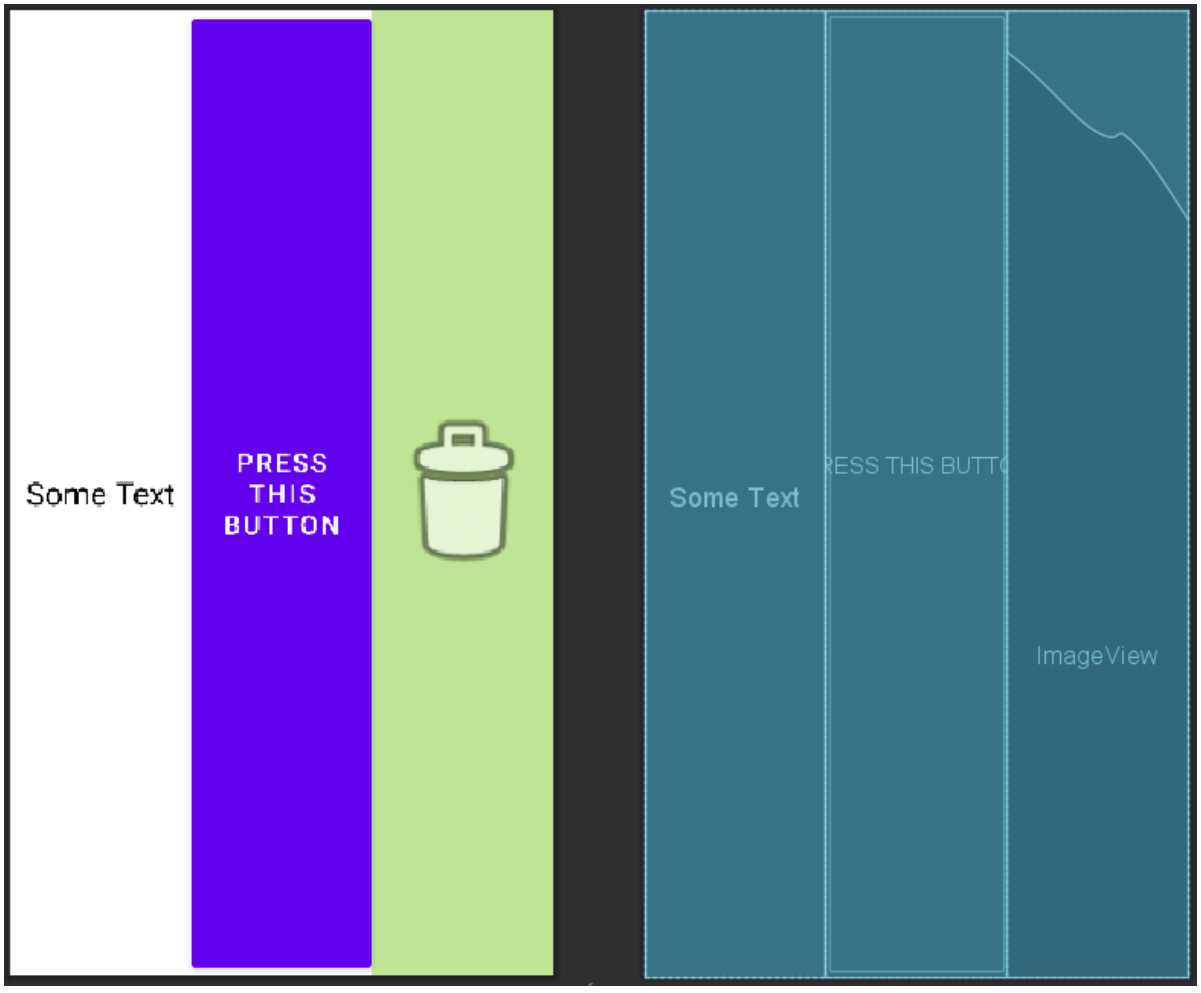

Рис. 7 Визуальное представление решения 2 (Изображение 1)

Изображение 10 (Рис. 2)

Для этого макета использовалось свойство Constaint Layout, которое позволяет задавать фиксированный отступ. Таким образом, верхний край верхнего элемента привязан к верхнему краю «родителя», а отступ составляет 150dp. Нижний край верхнего привязан к верхнему краю центрального элемента, нижний край центрального элемента привязан к верхнему краю нижнего элемента, нижний край нижнего элемента привязан к нижнему краю «родителя». Правые и левые края элементов привязаны к правому и левому краям «родителя» соответственно. После выставления атрибутов ширины и высоты в 0dp для всех объектов, они «поделили» оставшееся пространство экрана поровну.

Листинг 5 XML-код решения задачи 2 (Изображение 10)

```
<?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:id="@+id/linearLayout"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <!-- task 2, image 10 -->

    <CheckBox
        android:id="@+id/textView"
        android:layout_width="0dp"
        android:layout_height="0dp"
        android:layout_marginTop="150dp"

        android:background="@color/green"
        android:text="@string/widget3"
        android:textColor="@color/black"
        android:textSize="32sp"

        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintBottom_toTopOf="@+id/radioGroup"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintEnd_toEndOf="parent" />


    <RadioGroup
        android:id="@+id/radioGroup"
        android:layout_width="0dp"
        android:layout_height="0dp"

        app:layout_constraintTop_toBottomOf="@+id/textView"
        app:layout_constraintBottom_toTopOf="@+id/editText"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
```

```
        android:gravity="center"
        android:text="@string/widget2">

        <RadioButton
            android:id="@+id/radioButton"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="@string/choice1"
            android:textSize="24sp" />

        <RadioButton
            android:id="@+id/radioButton2"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="@string/choice2"
            android:textSize="24sp" />

        <RadioButton
            android:id="@+id/radioButton3"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginBottom="26dp"
            android:text="@string/choice3"
            android:textSize="24sp" />

    </RadioGroup>

    <EditText
        android:id="@+id/editText"
        android:layout_width="0dp"
        android:layout_height="0dp"

        app:layout_constraintTop_toBottomOf="@+id/radioGroup"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintEnd_toEndOf="parent"

        android:background="@color/yellow"
        android:gravity="center"
        android:text="@string/widget5"
        android:textSize="24sp"/>

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

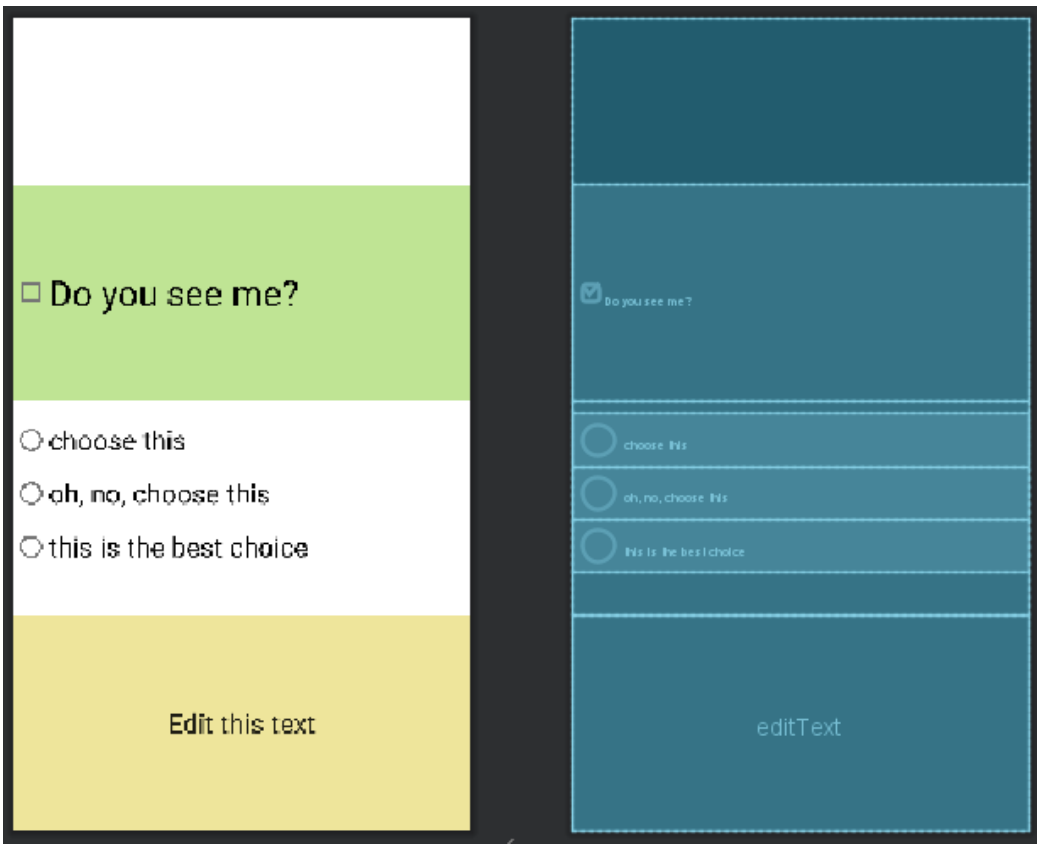

Рис. 8 Визуальное представление решения 2 (Изображение 10)

3.3. Задача 3. ConstraintLayout

Изображение 23 (Рис. 3)

По заданию необходимо, чтобы все представленные на рисунке элементы были вписаны в квадрат, для этого создадим ещё один ConstraintLayout, который имеет форму квадрата, занимает максимальную площадь и располагается в центре экрана. Этому ConstraintLayout зададим соотношение сторон равное 1 с помощью атрибута *app:layout_constraintDimensionRatio.*

Также было добавлено два элемента Guideline, вертикальный и горизонтальный, которые помогли в выстраивании почти всех элементов. Вертикальная линия делит квадрат в соотношении 3:2, а горизонтальная – 4:1.

Атрибут *layout_constraintDimensionRatio* помог также и для того, чтобы выставить размеры для каждого блока, их соотношение сторон 0.5 или 2 в зависимости от ориентации.

Атрибуты для привязки элементов и настройки высоты и ширины использовались как в предыдущих решениях.

Листинг 6 XML-код решения задачи 3 (Изображение 23)

```
<?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"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <!-- task 3, image 23 -->

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

        <androidx.constraintlayout.widget.Guideline
            android:id="@+id/vertical_guideline"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:orientation="vertical"
            app:layout_constraintGuide_percent="0.6" />

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

        <TextView
            android:id="@+id/view1"
            android:layout_width="0dp"
            android:layout_height="0dp"
            app:layout_constraintDimensionRatio="2"
            app:layout_constraintTop_toBottomOf="@id/view2"
            app:layout_constraintBottom_toTopOf="@id/view6"
            app:layout_constraintEnd_toStartOf="@id/vertical_guideline"

            android:text="@string/str_1"
            android:gravity="center"
            android:textColor="@color/black"
            android:textSize="24sp"
            android:background="@color/yellow"/>

        <Button
```

```
        android:id="@+id/view2"
        android:layout_width="0dp"
        android:layout_height="0dp"
        app:layout_constraintDimensionRatio="2"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintStart_toEndOf="@id/vertical_guideline"
        app:layout_constraintEnd_toEndOf="parent"

        android:text="@string/str_2"
        android:gravity="center"
        android:textSize="24sp"/>

    <CheckBox
        android:id="@+id/view3"
        android:layout_width="0dp"
        android:layout_height="0dp"
        app:layout_constraintDimensionRatio="2"
        app:layout_constraintTop_toBottomOf="@id/view5"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintStart_toEndOf="@id/view4"

        android:text="@string/str_3"
        android:gravity="center"
        android:textColor="@color/black"
        android:textSize="24sp"
        android:background="@color/orange"/>

    <androidx.appcompat.widget.SwitchCompat
        android:id="@+id/view4"
        android:layout_width="0dp"
        android:layout_height="0dp"
        app:layout_constraintDimensionRatio="2"
        app:layout_constraintBottom_toTopOf="@+id/horizontal_guideline"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintEnd_toStartOf="@id/view3"

        android:text="@string/str_4"
        android:gravity="center"
        android:textColor="@color/black"
        android:textSize="24sp"
        android:background="@color/light_blue"/>

    <ImageView
        android:id="@+id/view5"
        android:layout_width="0dp"
        android:layout_height="0dp"
        app:layout_constraintDimensionRatio="0.5"
        app:layout_constraintTop_toBottomOf="@id/view1"
        app:layout_constraintBottom_toTopOf="@id/horizontal_guideline"
        app:layout_constraintStart_toEndOf="@id/view3"
        app:layout_constraintEnd_toEndOf="parent"

        android:contentDescription="@string/widget4"
        android:src="@android:drawable/ic_menu_delete"
        android:background="@color/light_pink"/>

    <RadioGroup
        android:id="@+id/view6"
        android:layout_width="0dp"
        android:layout_height="0dp"
        app:layout_constraintDimensionRatio="0.5"
        app:layout_constraintBottom_toTopOf="@id/horizontal_guideline"
        app:layout_constraintStart_toEndOf="@+id/view4"
        app:layout_constraintEnd_toStartOf="@id/vertical_guideline"
```

```
            android:gravity="center"
            android:background="@color/light_purple">

            <RadioButton
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:text="@string/str_6"
                android:textSize="24sp" />

            <RadioButton
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:text="@string/str_6"
                android:textSize="24sp"/>

        </RadioGroup>

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

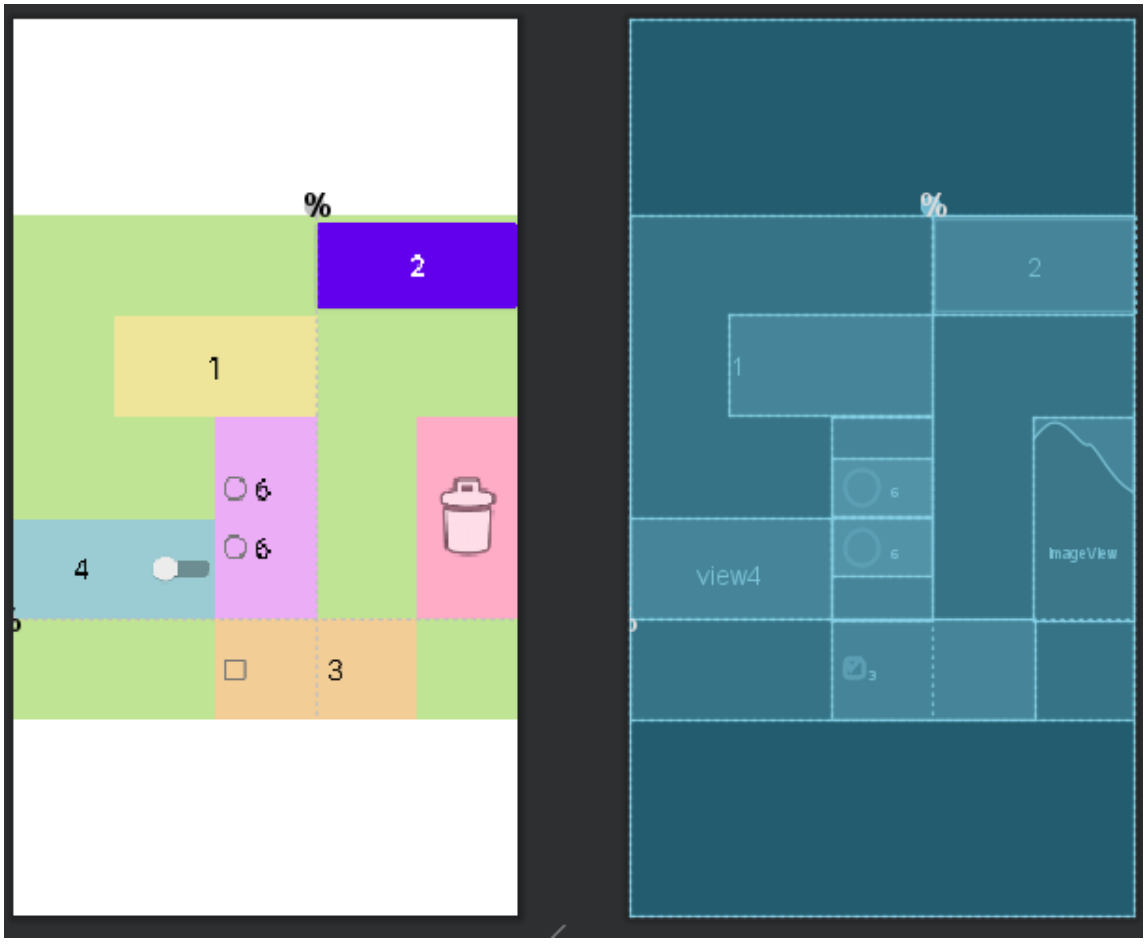

Рис. 9 Визуальное представление решения 3 (Изображение 23)

Если приблизить Рис. 9 (Рис. 10) видно, что в preview между виджетами видны «зазоры» и «наезды».

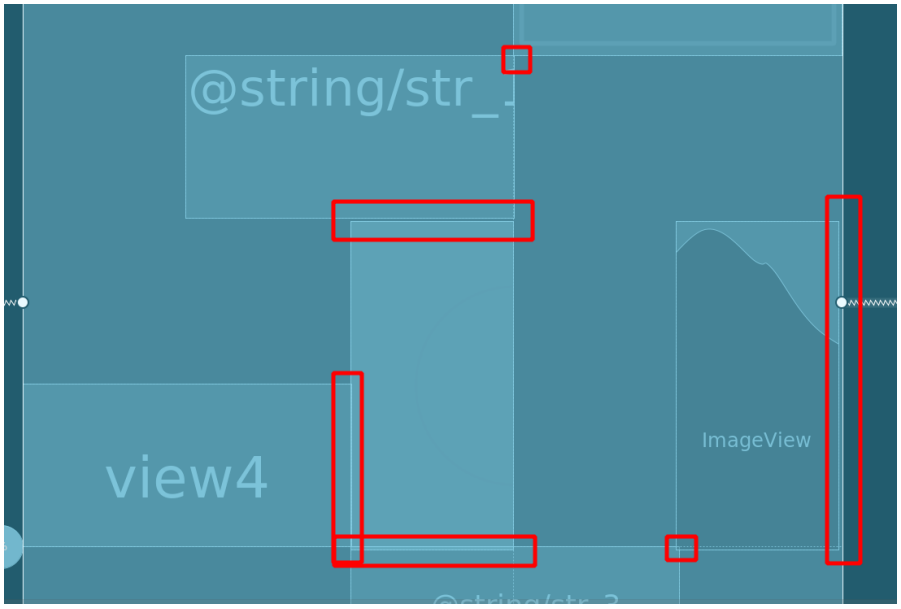

Рис. 10 Увеличенный вид

Однако причина не в коде, а в самом представлении компонентов в preview. Продемонстрируем это на тривиальном примере, разместив три виджета в ряд.

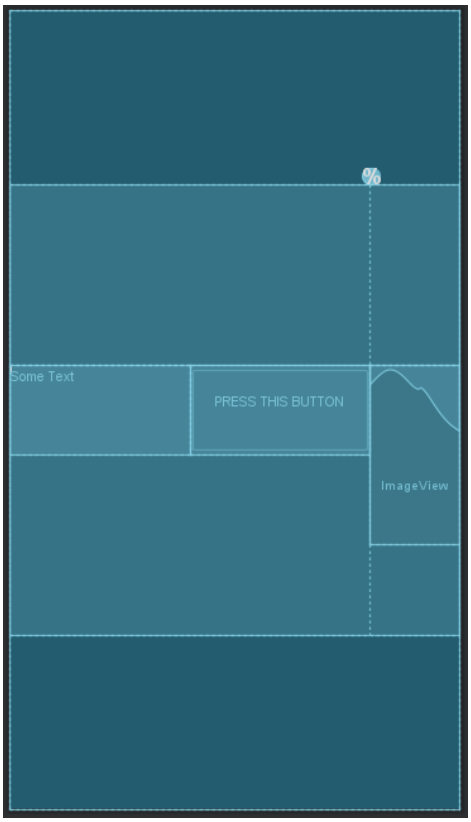

Рис. 11

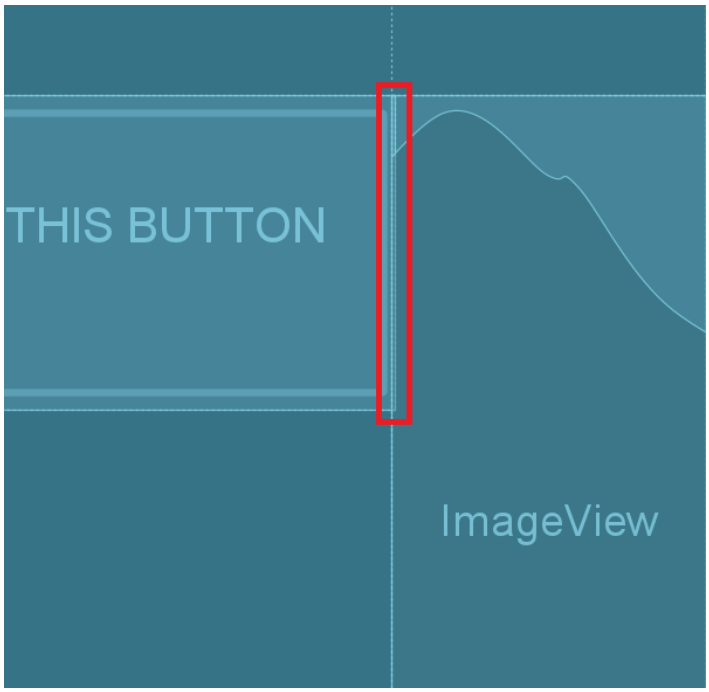

Рис. 12 Выделенное пересечение

На Рис. 12 видно, что один виджет наезжает на другой.

Посмотрим отображение layout_3_23.xml в Layout Inspector (Рис. 13-16). На рисунках видно, что «зазоры» и «наезды» отсутствуют.

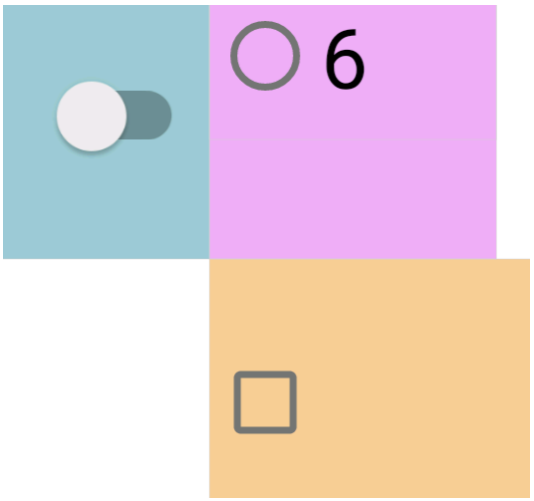

Рис. 13 Стык 4-6 элементов

Рис. 14 Стык 3 и 5 элементов

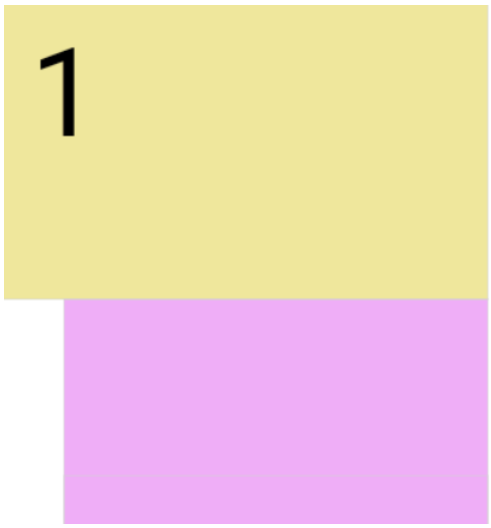

Рис. 15 Стык 1 и 6 элементов

Рис. 16 Стык 5 элемента и «родителя»

4. Вывод

В ходе данной лабораторной работы были изучены основные элементы среды разработки Android Studio, и так как она основа на уже знакомой IDE от JetBrains, то большинство функций для работы с кодом и проектом интуитивно понятно. Однако в Android Studio есть достаточно много нового, например, различные функции, связанные с проектированием. Оказалось, удобным выносить константы в отдельные файлы ресурсов, к примеру, строки и цвета (strings.xml и colors.xml). Отдельное внимание стоит уделить редактору компоновок, который позволяет создавать layout и размещать и редактировать на нем различные виджеты интерактивно вместо того, чтобы задавать описание xml файла.

Так же, были изучены основы верстки android-приложений, возможности и свойства таких структур как LinearLayout и ConstraintLayout.

Главным их отличием является то, что в первом случае элементы предоставляются в линейном виде (один за другим), а во втором же может производиться «привязка» элементов друг к другу в произвольном порядке. Выбор оптимальной конструкции зависит от целей макета: если он должен быть максимально простым, не требуется сложное расположение элементов, они располагаются в строчку или колонку, то лучше использовать LinearLayout. В случае более сложного макета расположения элементов лучше использовать ConstraintLayout, это более оптимально, так как при вложенных друг в друга LinearLayout при указании весов будет долго рассчитываться расположение элементов.

В результате работы удалось создать макеты в соответствии с исходными заданиями, они были протестированы на устройствах с различными размерами экрана, при этом макеты не меняли своего вида. Таким образом, можно сказать, что все цели работы были выполнены.