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 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_М.Ю. Петров

(подпись)

Принял старший преподаватель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_А.Н. Кузнецов

(подпись)

“ ” 2021 г.

Санкт-Петербург

2021

### Цели работы

* 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

* Создайте layout ресурсы для следующих макетов экрана с использованием LinearLayout.

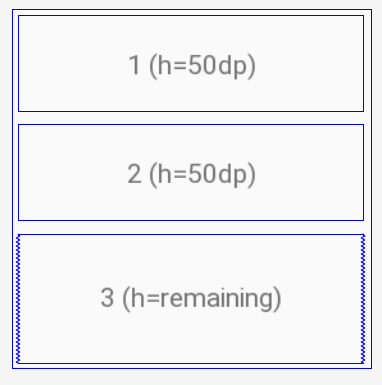
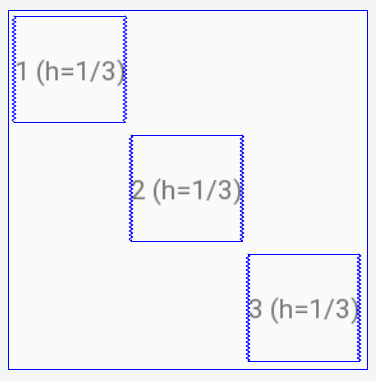
 

Рис. 1. Рис. 2.

* Решите задачу 1 (обе подзадачи) с использованием ConstraintLayout.

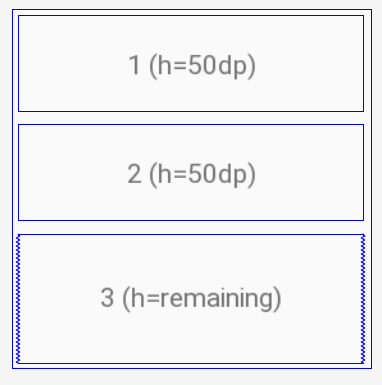
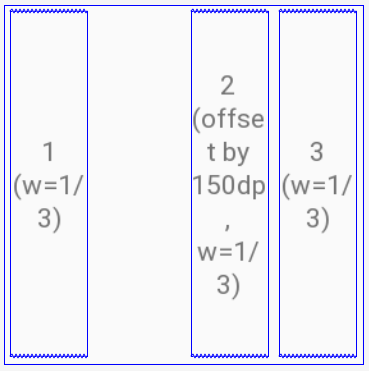
 

Рис. 3. Рис. 4.

* Создайте layout ресурс для следующего макета экрана с использованием ConstraintLayout.

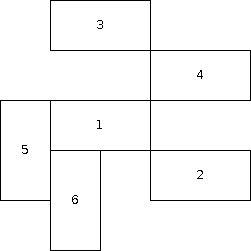


Рис. 5.

### Ход работы

### 3.1 Задача 1

### 3.1.1 Изображение 1

Для выполнения задачи была изучена документация о Linear Layout. Для моего макета необходимо, чтобы два виджета занимали одинаковое пространство высотой 50 dp, а третий виджет занимал все оставшееся пространство.

За то, сколько места на пространстве занимает каждый элемент, отвечает атрибут android:layout\_weight. Для моего макета необходимо использовать вертикальный Linear Layout, поэтому я выставил android:layout\_height в 50 dp для первых двух виджетов и 0 dp для последнего, а его вес указал равным 1. Таким образом, последний элемент занял оставшееся пустым пространство. Ширина виджетов задана, как android:layout\_width=”match\_parent”, чтобы они занимали всю ширину экрана. Также для удобства восприятия элементы были окрашены в различные цвета при помощи атрибута android:background.

Листинг 1. XML-код решения задачи 1.1

|  |
| --- |
| <?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="vertical">   <ImageView  android:id="@+id/imageView"  android:layout\_width="match\_parent"  android:layout\_height="50dp"  android:background="@color/black" />   <ImageButton  android:id="@+id/imageButton"  android:layout\_width="match\_parent"  android:layout\_height="50dp"  android:background="#FF0000" />   <Button  android:id="@+id/button"  android:layout\_width="match\_parent"  android:layout\_height="0dp"  android:layout\_weight="1" />  </LinearLayout> |

Из всех приведенных выше тегов опциональным является только android:background. При удалении остальных тегов нарушаются пропорции расположения элементов на экране.

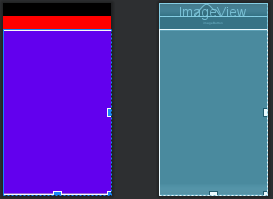


Рис. 6. Визуальное представление решения задачи 1.1.

### Изображение 2

Для приведенного на рис. 2 макета необходимо расположить три равных по высоте виджета диагонально так, чтобы они занимали всю ширину экрана. Ниже приведены два способа решения данной задачи: с использованием атрибута android:layout\_gravity и с использованием атрибута android:marginLeft.

### Способ 1

Атрибуты android:layout\_height, android:layout\_width, android:layout\_weight и android:background использованы в тех же целях, что и в предыдущем задании. Поскольку используется вертикальный LinearLayout, атрибут android:layout\_gravity отвечает за положение элементов относительно левой и правой сторон экрана.

Листинг 2. XML-код решения задачи 1.2.

|  |
| --- |
| <?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="vertical">   <ImageView  android:id="@+id/imageView"  android:layout\_width="140dp"  android:layout\_height="0dp"  android:background="@color/black"  android:layout\_weight="1" />   <ImageButton  android:id="@+id/imageButton"  android:layout\_width="140dp"  android:layout\_height="0dp"  android:layout\_gravity="center"  android:background="#FF0000"  android:layout\_weight="1" />   <Button  android:id="@+id/button"  android:layout\_width="140dp"  android:layout\_height="0dp"  android:layout\_gravity="end"  android:layout\_weight="1" />  </LinearLayout> |

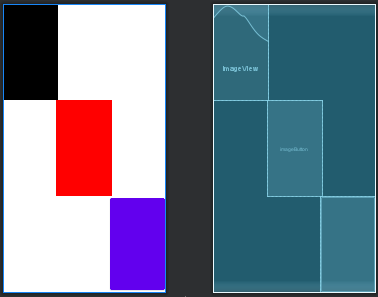


Рис. 7. Визуальное представление решения задачи 1.2.

### 2 способ

Вместо android:layout\_gravity можно использовать android:marginLeft. Данный атрибут задает сдвиг на указанное количество dp влево, что также позволяет выставить виджеты на требуемые макетом позиции.

Листинг 3. XML-код решения задачи 1.2(2) .

|  |
| --- |
| <?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="vertical">   <ImageView  android:id="@+id/imageView"  android:layout\_width="140dp"  android:layout\_height="0dp"  android:background="@color/black"  android:layout\_weight="1" />   <ImageButton  android:id="@+id/imageButton"  android:layout\_width="140dp"  android:layout\_height="0dp"  android:layout\_weight="1"  android:layout\_marginLeft="140dp"  android:background="#FF0000" />   <Button  android:id="@+id/button"  android:layout\_width="140dp"  android:layout\_height="0dp"  android:layout\_marginLeft="280dp"  android:layout\_weight="1" />  </LinearLayout> |

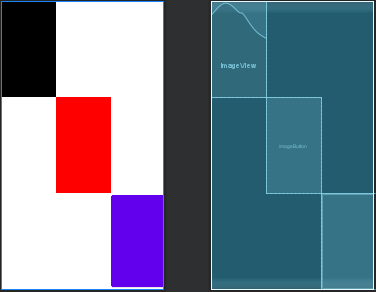


Рис. 8. Визуальное представление решения задачи 1.2(2).

### 3.2 Задача 2

### 3.2.1 Изображение 1

Для решения этой задачи использовался Constraint Layout. Он позволяет привязывать края объекта к краям объектов или родителя и задавать отступы.

Для макета на рис. 3 верхний объект привязан к родителю сверху. Следующие элементы привязываются сверху к низу предыдущих элементов, а последний привязан также к низу родителя, чтобы растянуть его на всю высоту экрана. Высота первых двух элементов равна 50 dp. Ширина всех элементов соответствует ширине экрана.

Листинг 4. XML-код решения задачи 2.1.

|  |
| --- |
| <?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"  android:id="@+id/linearLayout"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent">   <ImageView  android:id="@+id/imageView"  android:layout\_width="match\_parent"  android:layout\_height="50dp"  android:background="@color/black"  app:layout\_constraintTop\_toTopOf="parent" />   <ImageButton  android:id="@+id/imageButton"  android:layout\_width="match\_parent"  android:layout\_height="50dp"  android:background="#FF0000"  app:layout\_constraintTop\_toBottomOf="@+id/imageView" />   <Button  android:id="@+id/button"  android:layout\_width="match\_parent"  android:layout\_height="0dp"  app:layout\_constraintBottom\_toBottomOf="parent"  app:layout\_constraintTop\_toBottomOf="@+id/imageButton" />  </androidx.constraintlayout.widget.ConstraintLayout> |

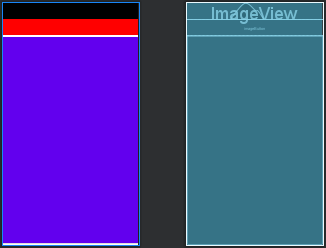


Рис. 9. Визуальное представление решения задачи 2.1.

### 3.2.2 Изображение 2

Для макета на рис. 4 все объекты связаны между собой, но конец первого объекта сдвинут на требуемые 150 dp при помощи атрибута android:layout\_marginEnd. Первый элемент слева и последний справа привязаны к родителю. Высота всех виджетов равна высоте экрана.

Листинг 5. XML-код решения задачи 2.2.

|  |
| --- |
| <?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:id="@+id/linearLayout2"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent">   <ImageView  android:id="@+id/imageView"  android:layout\_width="0dp"  android:layout\_height="match\_parent"  android:layout\_marginEnd="150dp"  android:background="@color/black"  app:layout\_constraintEnd\_toStartOf="@+id/imageButton"  app:layout\_constraintStart\_toStartOf="parent" />   <ImageButton  android:id="@+id/imageButton"  android:layout\_width="0dp"  android:layout\_height="match\_parent"  android:background="#FF0000"  app:layout\_constraintEnd\_toStartOf="@+id/button"  app:layout\_constraintStart\_toEndOf="@+id/imageView" />   <Button  android:id="@+id/button"  android:layout\_width="0dp"  android:layout\_height="match\_parent"  app:layout\_constraintEnd\_toEndOf="parent"  app:layout\_constraintStart\_toEndOf="@+id/imageButton" />  </androidx.constraintlayout.widget.ConstraintLayout> |

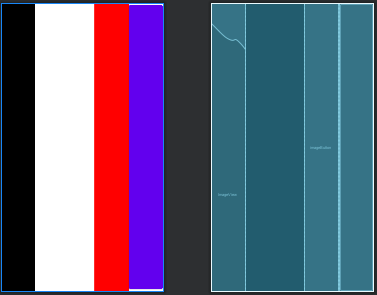


Рис. 10. Визуальное представление решения задачи 2.2.

### 3.3 Задача 3

Для того, чтобы квадрат с элементами был квадратом и занимал наибольшую площадь вне зависимости от ориентации девайса, была выполнена привязка его краев к краям экрана и установлено соотношение сторон 1:1 с помощью атрибута app:layout\_constraintDimensionRatio.

Два элемента Guideline являются ориентирами для выстраивания всех элементов. Первый делит квадрат в соотношении 1:4, а второй – 3:2.

Атрибут layout\_constraintDimensionRatio помог также и для того, чтобы выставить размеры для каждого блока и их соотношение сторон 1:2 (2:1).

Листинг 6. XML-код решения задачи 3.

|  |
| --- |
| <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">  <androidx.constraintlayout.widget.ConstraintLayout  android:layout\_width="0dp"  android:layout\_height="0dp"  app:layout\_constraintDimensionRatio="1:1"  app:layout\_constraintBottom\_toBottomOf="parent"  app:layout\_constraintLeft\_toLeftOf="parent"  app:layout\_constraintRight\_toRightOf="parent"  app:layout\_constraintTop\_toTopOf="parent">  <androidx.constraintlayout.widget.Guideline  android:id="@+id/guideline20"  android:layout\_width="0dp"  android:layout\_height="0dp"  android:orientation="vertical"  app:layout\_constraintGuide\_percent="0.2" />  <androidx.constraintlayout.widget.Guideline  android:id="@+id/guideline60"  android:layout\_width="0dp"  android:layout\_height="0dp"  android:orientation="vertical"  app:layout\_constraintGuide\_percent="0.6" />  <CalendarView  android:id="@+id/w1"  android:layout\_width="0dp"  android:layout\_height="0dp"  android:background="#90042D59"  app:layout\_constraintTop\_toBottomOf="@+id/w4"  app:layout\_constraintDimensionRatio="2:1"  app:layout\_constraintStart\_toStartOf="@+id/guideline20"  app:layout\_constraintEnd\_toEndOf="@+id/guideline60" />  <CheckBox  android:id="@+id/w2"  android:layout\_width="0dp"  android:layout\_height="0dp"  android:background="@color/teal\_200"  app:layout\_constraintDimensionRatio="2:1"  app:layout\_constraintEnd\_toEndOf="parent"  app:layout\_constraintStart\_toStartOf="@+id/guideline60"  app:layout\_constraintTop\_toBottomOf="@+id/w1" />  <VideoView  android:id="@+id/w3"  android:layout\_width="0dp"  android:layout\_height="0dp"  app:layout\_constraintDimensionRatio="2:1"  app:layout\_constraintEnd\_toEndOf="@+id/guideline60"  app:layout\_constraintStart\_toStartOf="@+id/guideline20"  app:layout\_constraintTop\_toTopOf="parent" />  <ImageButton  android:id="@+id/w4"  android:layout\_width="0dp"  android:layout\_height="0dp"  app:layout\_constraintDimensionRatio="2:1"  app:layout\_constraintEnd\_toEndOf="parent"  app:layout\_constraintStart\_toStartOf="@+id/guideline60"  app:layout\_constraintTop\_toBottomOf="@+id/w3" />  <ImageView  android:id="@+id/w5"  android:layout\_width="0dp"  android:layout\_height="0dp"  android:background="@color/black"  app:layout\_constraintDimensionRatio="1:2"  app:layout\_constraintEnd\_toStartOf="@+id/guideline20"  app:layout\_constraintStart\_toStartOf="parent"  app:layout\_constraintTop\_toTopOf="@+id/w1" />  <Button  android:id="@+id/w6"  android:layout\_width="0dp"  android:layout\_height="0dp"  app:layout\_constraintDimensionRatio="1:2"  app:layout\_constraintTop\_toBottomOf="@+id/w1"  app:layout\_constraintStart\_toStartOf="@+id/guideline20"  app:layout\_constraintBottom\_toBottomOf="parent" />  </androidx.constraintlayout.widget.ConstraintLayout> </androidx.constraintlayout.widget.ConstraintLayout> |

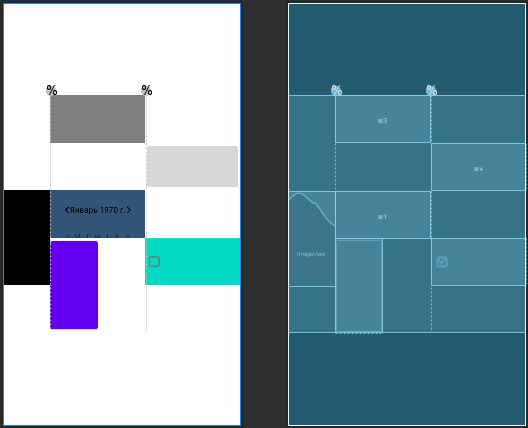


Рис. 11. Визуальное представление решения задачи 3.

### Вывод

В ходе работы были изучены основы верстки android-приложений и возможности и свойства таких структур как Linear Layout и Constraint Layout.

Основным их отличием является то, что в первом случае элементы представляются в линейном виде (один за другим), а во втором элементы привязываются друг к другу в произвольном порядке. Выбор конструкции зависит от целей макета: для простых макетов, не требующих сложного расположения элементов проще использовать Linear Layout. Если же элементы располагаются более сложно, лучше использовать Constraint Layout, что выгоднее по времени выполнения, так как при вложенных друг в друга Linear Layout при указании весов расположение элементов будет рассчитываться дольше.

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. Поэтому можно сказать, что все цели работы были выполнены.