Федеральное государственное автономное образовательное учреждение высшего образования «Санкт-Петербургский политехнический университет Петра Великого»

Институт компьютерных наук и технологий

Кафедра компьютерных систем и программных технологий

**Отчёт по лабораторной работе №1**

**Дисциплина:** проектирование мобильных приложений

**Тема:** «Layouts»

**Вариант №14**

Выполнил студент гр. 3530901/70202

Д.В. Павлов

*(подпись)*

Преподаватель

А.Н. Кузнецов

*(подпись)*

Санкт-Петербург

2019

1. Цели работы

- Ознакомиться со средой разработки Android Studio;

- Изучить основные принципы верстки layout с использованием View и ViewGroup;

- Изучить основные возможности и свойства LinearLayout;

- Изучить основные возможности и свойства ConstraintLayout.

2. Выполнение работы

2.1. LinearLayout

Задачей является создание ресурсов для макетов, представленных на Рис.2.1.1 и Рис.2.1.2 с использованием *LinearLayout*.

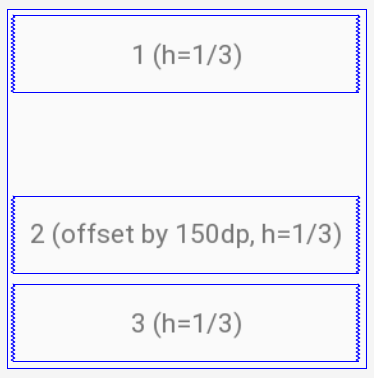


Рис.2.1.1

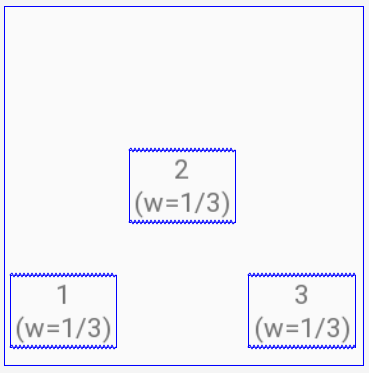


Рис.2.1.2

Перед началом выполнения работы было проведено изучение LinearLayout по документации от производителя ОС.

LinearLayout представляет собой набор виджетов, которые могут располагаться горизонтально или вертикально. Каждому элементу может быть задан определенный вес, который влияет на размерность. Мы можем задавать высоту и длину, по-разному располагать содержимое в виджетах (картинки в ImageView, текст в TextView).

Решение для первого макета экрана представлено на Рис.2.1.3 и Рис.2.1.4.

<?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">  
  
 <Button  
 android:id="@+id/button"  
 android:layout\_width="match\_parent"  
 android:layout\_height="0dp"  
 android:layout\_marginBottom="150dp"  
 android:layout\_weight="1"  
 android:text="Button" />  
  
 <ImageView  
 android:id="@+id/imageView2"  
 android:layout\_width="match\_parent"  
 android:layout\_height="0dp"  
 android:layout\_weight="1"  
 tools:srcCompat="@tools:sample/avatars[13]" />  
  
 <TextView  
 android:layout\_width="match\_parent"  
 android:layout\_height="0dp"  
 android:layout\_weight="1"  
 android:gravity="center"  
 android:text="Hello World!" />  
  
</LinearLayout>

Рис.2.1.3

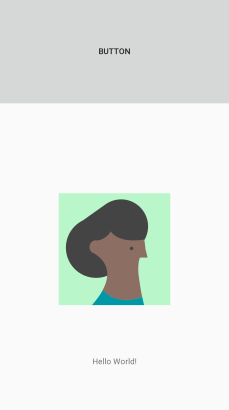


Рис.2.1.4

Ниже представлено описание используемых атрибутов:

**margin** – создание отступов между двумя виджетами (виджетом и родителем).

**layout\_weight** – задание веса объекта. В данном случае для всех трех виджетов введен вес равный единице с той целью, чтобы они равномерно занимали отданное под них пространство;

**gravity** – определяет положения содержимого виджета. В представленном решении текст располагается в середине кнопки и *textVIew*, что определяется как *gravity=”center”*. Однако в случае с кнопкой удалось обойтись без него.

Решение для второго макета представлено на Рис.2.1.5-2.1.6.

<?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"  
 tools:context=".Main2Activity">  
  
 <Button  
 android:id="@+id/button2"  
 android:layout\_width="0dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_gravity="bottom"  
 android:layout\_weight="1"  
 android:text="Button" />  
  
 <ImageView  
 android:id="@+id/imageView"  
 android:layout\_width="0dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_gravity="center"  
 android:layout\_weight="1"  
 tools:srcCompat="@tools:sample/avatars[2]" />  
  
 <TextView  
 android:id="@+id/textView"  
 android:layout\_width="0dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_gravity="bottom"

android:gravity="center"  
 android:layout\_weight="1"  
 android:text="TextView" />  
</LinearLayout>

Рис.2.1.5

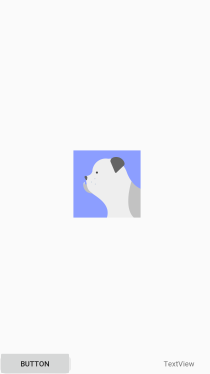


Рис.2.1.6.

В данном случае был задействован новый атрибут – *layout\_gravity*. Отличие от *gravity* заключается в том, что *layout\_gravity* определяет положение виджета в пространстве, а не его контента внутри.

По условиям задания, было приведено еще одно решение первого пункта задачи. Вместо использования *marginBottom* применительно к верхнему виджету был использован *marginTop* применительно к среднему виджету. (Рис.2.1.7).

<?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">  
  
 <Button  
 android:id="@+id/button"  
 android:layout\_width="match\_parent"  
 android:layout\_height="0dp"  
 android:layout\_weight="1"  
 android:text="Button" />  
  
 <ImageView  
 android:id="@+id/imageView2"  
 android:layout\_width="match\_parent"  
 android:layout\_height="0dp"  
 android:layout\_weight="1"  
 android:layout\_marginTop="150dp"  
 tools:srcCompat="@tools:sample/avatars[13]" />  
  
 <TextView  
 android:layout\_width="match\_parent"  
 android:layout\_height="0dp"  
 android:layout\_weight="1"  
 android:gravity="center"  
 android:text="Hello World!" />  
  
</LinearLayout>

Рис.2.1.7.

2.2. ConstraintLayout

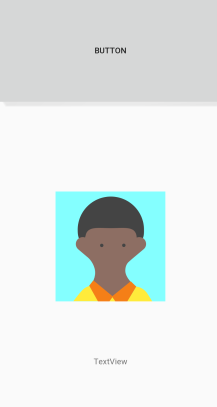
Задачей является создание ресурсов по макетам из п.2.1. с использованием *ConstraintLayout*.

*ConstraintLayout* позволяет создавать сложные ресурсы. Для этого каждый виджет имеет зависимость либо от других виджетов, либо от родителя. Допускается их коминация. Для определения позиции виджета в *ConstraintLayout* необходимо добавить как минимум либо одну горизонтальную, либо одну вертикальную зависимость. *ConstraintLayout* схож с *RelativeLayout,* но является более универсальным инструментом.

Решение первой задачи с использованием *ConstraintLayout* представлено на Рис.2.2.1-2.2.2.

<?xml version="1.0" encoding="utf-8"?>  
<android.support.constraint.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"  
 tools:context=".Main3Activity">  
  
 <Button  
 android:id="@+id/button3"  
 android:layout\_width="match\_parent"  
 android:layout\_height="0dp"  
 android:layout\_marginStart="8dp"  
 android:layout\_marginLeft="8dp"  
 android:layout\_marginEnd="8dp"  
 android:layout\_marginRight="8dp"  
 android:text="Button"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="1.0"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintBottom\_toTopOf="@id/space1"  
 app:layout\_constraintVertical\_weight="1" />  
  
 <ImageView  
 android:id="@+id/imageView3"  
 android:layout\_width="match\_parent"  
 android:layout\_height="0dp"  
 app:layout\_constraintVertical\_weight="1"  
 app:layout\_constraintTop\_toBottomOf="@id/space1"  
 app:layout\_constraintBottom\_toTopOf="@id/textView2"  
 tools:srcCompat="@tools:sample/avatars[10]" />  
  
 <TextView  
 android:id="@+id/textView2"  
 android:layout\_width="match\_parent"  
 android:layout\_height="0dp"  
 android:layout\_marginStart="8dp"  
 android:layout\_marginLeft="8dp"  
 android:layout\_marginTop="8dp"  
 android:layout\_marginEnd="8dp"  
 android:layout\_marginRight="8dp"  
 android:gravity="center"  
 android:text="TextView"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/imageView3"  
 app:layout\_constraintVertical\_weight="1" />  
  
 <Space  
 android:id="@+id/space1"  
 android:layout\_width="match\_parent"  
 android:layout\_height="150dp"  
 app:layout\_constraintBottom\_toTopOf="@+id/imageView3"  
 app:layout\_constraintTop\_toBottomOf="@+id/button3"  
 app:layout\_constraintVertical\_bias="0.0"  
 app:layout\_constraintVertical\_weight="1"  
 tools:layout\_editor\_absoluteX="-16dp" />  
</android.support.constraint.ConstraintLayout>

2.2.1



2.2.3

Ниже представлено описание используемых атрибутов:

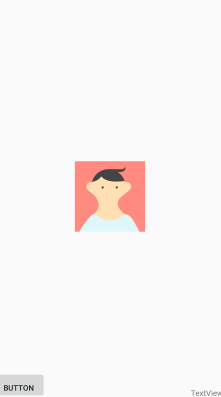
**margin –** задание отступов объекта от других объектов/родителя;

**constraint –** привязка (задание связей) к другим объектам/родителю;

**constraintVertical\_weight –** задание веса по вертикальной;

**constraintVertical\_bias -** координаты расположения объекта на вертикальной оси.

Решение второй задачи с использованием *ConstraintLayout* представлено на Рис.2.2.3-2.2.5.



2.2.3

<?xml version="1.0" encoding="utf-8"?>  
<android.support.constraint.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"  
 tools:context=".Main4Activity">  
  
 <Button  
 android:id="@+id/button4"  
 android:layout\_width="0dp"  
 android:layout\_height="wrap\_content"  
 android:gravity="bottom"  
 android:text="Button"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintLeft\_toLeftOf="parent"  
 app:layout\_constraintVertical\_weight="1" />  
  
 <ImageView  
 android:id="@+id/imageView4"  
 android:layout\_width="0dp"  
 android:layout\_height="wrap\_content"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintLeft\_toLeftOf="parent"  
 app:layout\_constraintRight\_toRightOf="parent"  
 app:layout\_constraintVertical\_weight="1"  
 tools:srcCompat="@tools:sample/avatars" />  
  
 <TextView  
 android:id="@+id/textView3"  
 android:layout\_width="0dp"  
 android:layout\_height="wrap\_content"  
 android:gravity="bottom"  
 android:text="TextView"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintRight\_toRightOf="parent"  
 app:layout\_constraintVertical\_weight="1" />  
</android.support.constraint.ConstraintLayout>

2.2.4

2.3. ConstraintLayout

Данный пункт предусматривает создание ресурса по макету, представленному на Рис.2.3.1.

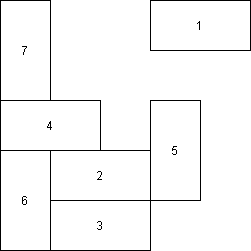


Рис.2.3.1

ConstraintLayout имеет форму квадрата, занимает максимальную площадь и располагается в центре экрана.

Решение задачи представлено на Рис.2.3.2-2.3.3.

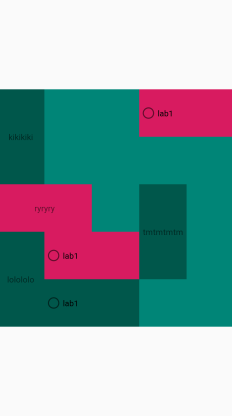


Рис.2.3.2

<?xml version="1.0" encoding="utf-8"?>  
<android.support.constraint.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"  
 tools:context=".Main5Activity">  
  
 <android.support.constraint.Guideline  
 android:id="@+id/guidelineStart"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:orientation="vertical"  
 app:layout\_constraintGuide\_percent="0.0" />  
  
 <android.support.constraint.Guideline  
 android:id="@+id/guidelineEnd"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical"  
 app:layout\_constraintGuide\_percent="1.0" />  
  
 <android.support.constraint.ConstraintLayout  
 android:layout\_width="0dp"  
 android:layout\_height="0dp"  
 android:background="@color/colorPrimary"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintDimensionRatio="1:1"  
 app:layout\_constraintLeft\_toLeftOf="@id/guidelineStart"  
 app:layout\_constraintRight\_toRightOf="@id/guidelineEnd"  
 app:layout\_constraintTop\_toTopOf="parent">  
  
 <TextView  
 android:id="@+id/view1"  
 android:layout\_width="0dp"  
 android:layout\_height="0dp"  
 android:background="@color/colorPrimaryDark"  
 android:gravity="center"  
 android:text="kikikiki"  
 app:layout\_constraintHorizontal\_bias="0.0"  
 app:layout\_constraintVertical\_weight="1"  
 app:layout\_constraintHorizontal\_weight="0.5"  
 app:layout\_constraintBottom\_toTopOf="@+id/view5"  
 app:layout\_constraintDimensionRatio="1:2"  
 app:layout\_constraintLeft\_toLeftOf="parent"  
 app:layout\_constraintRight\_toLeftOf="@+id/view2"  
 app:layout\_constraintTop\_toTopOf="parent" />  
  
  
 <Space  
 android:id="@+id/view2"  
 android:layout\_width="0dp"  
 android:layout\_height="0dp"  
 app:layout\_constraintVertical\_weight="1"  
 app:layout\_constraintHorizontal\_weight="0.5"  
 app:layout\_constraintRight\_toLefttOf="@+id/view3"  
 app:layout\_constraintDimensionRatio="1:2"  
 app:layout\_constraintLeft\_toRightOf="@+id/view1"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintBottom\_toTopOf="@+id/view5"/>  
  
 <Space  
 android:id="@+id/view3"  
 android:layout\_width="0dp"  
 android:layout\_height="0dp"  
 app:layout\_constraintHorizontal\_weight="0.5"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintDimensionRatio="1:2"  
 app:layout\_constraintLeft\_toRightOf="@+id/view2"  
 app:layout\_constraintRight\_toLeftOf="@+id/view4"/>  
  
 <RadioButton  
 android:id="@+id/view4"  
 android:layout\_width="0dp"  
 android:layout\_height="0dp"  
 android:background="@color/colorAccent"  
 android:text="@string/app\_name"  
 app:layout\_constraintHorizontal\_weight="1"  
 app:layout\_constraintLeft\_toRightOf="@id/view3"  
 app:layout\_constraintDimensionRatio="2:1"  
 app:layout\_constraintRight\_toRightOf="parent"  
 app:layout\_constraintBottom\_toTopOf="@+id/view8"  
 app:layout\_constraintTop\_toTopOf="parent"/>  
  
 <Space  
 android:id="@+id/view8"  
 android:layout\_width="0dp"  
 android:layout\_height="0dp"  
 android:background="@color/colorAccent"  
 android:text="@string/app\_name"  
 app:layout\_constraintLeft\_toRightOf="@id/view3"  
 app:layout\_constraintDimensionRatio="2:1"  
 app:layout\_constraintRight\_toRightOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/view4"/>  
  
 <Space  
 android:id="@+id/view10"  
 android:layout\_width="0dp"  
 android:layout\_height="0dp"  
 android:background="@color/colorAccent"  
 android:text="@string/app\_name"  
 app:layout\_constraintDimensionRatio="1:2"  
 app:layout\_constraintRight\_toRightOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/view8"/>  
  
 <Space  
 android:id="@+id/view13"  
 android:layout\_width="0dp"  
 android:layout\_height="0dp"  
 android:background="@color/colorAccent"  
 android:text="@string/app\_name"  
 app:layout\_constraintLeft\_toRightOf="@id/view12"  
 app:layout\_constraintDimensionRatio="2:1"  
 app:layout\_constraintRight\_toRightOf="parent"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/view10"/>  
  
 <RadioButton  
 android:id="@+id/view12"  
 android:layout\_width="0dp"  
 android:layout\_height="0dp"  
 android:background="@color/colorPrimaryDark"  
 android:text="@string/app\_name"  
 app:layout\_constraintLeft\_toRightOf="@id/view6"  
 app:layout\_constraintTop\_toBottomOf="@id/view11"  
 app:layout\_constraintDimensionRatio="2:1"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 />  
  
 <RadioButton  
 android:id="@+id/view11"  
 android:layout\_width="0dp"  
 android:layout\_height="0dp"  
 android:background="@color/colorAccent"  
 android:text="@string/app\_name"  
 app:layout\_constraintLeft\_toRightOf="@id/view6"  
 app:layout\_constraintDimensionRatio="2:1"  
 app:layout\_constraintBottom\_toTopOf="@+id/view12"  
 app:layout\_constraintTop\_toBottomOf="@+id/view5" />  
  
 <TextView  
 android:id="@+id/view9"  
 android:layout\_width="0dp"  
 android:layout\_height="0dp"  
 android:background="@color/colorPrimaryDark"  
 android:gravity="center"  
 android:text="tmtmtmtm"  
 app:layout\_constraintBottom\_toTopOf="@+id/view13"  
 app:layout\_constraintDimensionRatio="1:2"  
 app:layout\_constraintLeft\_toRightOf="@+id/view11"  
 app:layout\_constraintRight\_toLeftOf="@+id/view10"  
 app:layout\_constraintTop\_toBottomOf="@+id/view8" />  
  
 <TextView  
 android:id="@+id/view5"  
 android:layout\_width="0dp"  
 android:layout\_height="0dp"  
 android:background="@color/colorAccent"  
 android:gravity="center"  
 android:text="ryryry"  
 app:layout\_constraintHorizontal\_bias="0.0"  
 app:layout\_constraintVertical\_weight="0.5"  
 app:layout\_constraintDimensionRatio="2:1"  
 app:layout\_constraintLeft\_toLeftOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/view1"  
 app:layout\_constraintBottom\_toTopOf="@+id/view6"/>  
  
 <TextView  
 android:id="@+id/view6"  
 android:layout\_width="0dp"  
 android:layout\_height="0dp"  
 android:background="@color/colorPrimaryDark"  
 android:gravity="center"  
 android:text="lolololo"  
 app:layout\_constraintHorizontal\_bias="0.0"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintDimensionRatio="1:2"  
 app:layout\_constraintLeft\_toLeftOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/view5"  
 app:layout\_constraintVertical\_weight="1"  
 />  
  
  
 </android.support.constraint.ConstraintLayout>  
  
</android.support.constraint.ConstraintLayout>

Рис.2.3.3

Для выполнения данной задачи потребовался *constraintDimensionRatio.* С его помощью можно задавать соотношение сторон в формате “*width:height*”.

А также было задействовано много Space объектов.

3. Выводы

При выполнении данной лабораторной работы было проведено ознакомление со средой разработки AndroidStudio и двумя типами *layout: ConstraintLayout, LinearLayout*. Были изучены их основные возможности и свойства, а именно: отмечены их отличительные особенности и описаны основные атрибуты, использовавшиеся при создании ресурсов в соответствии с заданием.

*LinearLayout* является удобным и простым инструментом, так как позволяет расположить объекты в ряд (по вертикали/горизонтали), что не является сложной задачей.

*ConstraintLayout* является более сложным по сравнению с *LinearLayout*, однако предоставляет более широкие возможности расположения объектов. Виджеты здесь располагаются любым возможным способом, не только по вертикали/горизонтали. Однако это несет за собой определенный сложности. При работе с данным типом *layout* необходимо обозначать большее количество атрибутов, настраивать связи относительно минимум одного объекта, чтобы вся конструкция смотрелась органично.