САНКТ-ПЕТЕРБУРГСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

ПЕТРА ВЕЛИКОГО

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Институт компьютерных наук и технологий

Высшая школа интеллектуальных систем и суперкомпьютерных технологий

Отчет по лабораторной работе №1

«Layouts»

Дисциплина

«Проектирование мобильных приложений»

Выполнил: Луняк Н. А.

Группа: 3530901/80202

Преподаватель: Алексюк А. О.

Санкт-Петербург

2020

СОДЕРЖАНИЕ

[Список иллюстраций 3](#_Toc50930719)

[Список Листингов 3](#_Toc50930720)

[1 Цели 4](#_Toc50930721)

[2 Общее 4](#_Toc50930722)

[3 Задача 1. LinearLayout 4](#_Toc50930723)

[3.1 15.png 4](#_Toc50930724)

[3.2 02.png 6](#_Toc50930725)

[3.3 06.png 8](#_Toc50930726)

[4 Задача 2. ConstraintLayout 10](#_Toc50930727)

[4.1 15.png 10](#_Toc50930728)

[4.2 02.png 12](#_Toc50930729)

[4.3 06.png 12](#_Toc50930730)

[5 Задача 3. ConstraintLayout 13](#_Toc50930731)

[5.1 15.png 13](#_Toc50930732)

[5.2 06.png 17](#_Toc50930733)

[6 Выводы 21](#_Toc50930734)

# Список иллюстраций

[Рис. 2‑1 labs/01/linear/15.png 4](#_Toc50930735)

[Рис. 2‑2 Результат 6](#_Toc50930736)

[Рис. 2‑3 Проверка android:layout\_gravity 6](#_Toc50930737)

[Рис. 2‑1 labs/01/linear/02.png 7](#_Toc50930738)

[Рис. 2‑2 Результат 8](#_Toc50930739)

[Рис. 2‑1 labs/01/linear/06.png 9](#_Toc50930740)

[Рис. 2‑2 Результат 10](#_Toc50930741)

[Рис. 4‑1 labs/01/constraint/lab01\_constraint\_v15.png 14](#_Toc50930742)

[Рис. 4‑2 Визуализация 17](#_Toc50930743)

[Рис. 4‑1 labs/01/constraint/lab01\_constraint\_v06.png 18](#_Toc50930744)

[Рис. 4‑2 Визуализация 21](#_Toc50930745)

# Список Листингов

[Листинг 2‑1 Код для labs/01/linear/15.png 5](#_Toc50930746)

[Листинг 2‑2 Код для android:layout\_gravity 6](#_Toc50930747)

[Листинг 2‑1 Код для labs/01/linear/02.png 7](#_Toc50930748)

[Листинг 2‑1 Код для labs/01/linear/06.png 9](#_Toc50930749)

[Листинг 3‑1 Используем ConstraintLayout 11](#_Toc50930750)

[Листинг 3‑1 Эффект «вытеснения» 11](#_Toc50930751)

[Листинг 3‑1 Используем ConstraintLayout 12](#_Toc50930752)

[Листинг 3‑1 Используем ConstraintLayout 13](#_Toc50930753)

[Листинг 4‑1 Куча guideline’ов 17](#_Toc50930754)

[Листинг 4‑1 Куча guideline’ов 20](#_Toc50930755)

# Цели

* 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

# Общее

Так как я в группе нахожусь на 6-ой позиции, то img1 = 15, img2 = 2.

Интересный факт: если бы я обратил на это внимание, когда начинал делать лабу, настоящий отчет был бы в 1.5 раза короче.

# Задача 1. LinearLayout

## 15.png

Хотим получить вот такое.

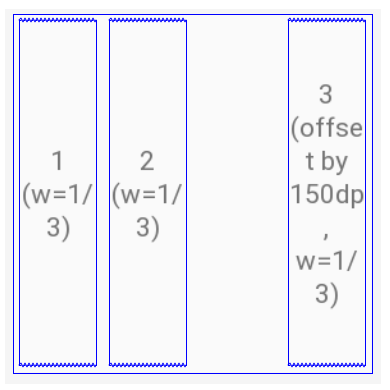


Рис. 3‑1 labs/01/linear/15.png

Делаем так.

|  |
| --- |
| *<?*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">   <Button  android:layout\_width="0dp"  android:layout\_height="match\_parent"  android:text="Hello World!"  android:layout\_weight="0.333" />   <TextView  android:layout\_width="0dp"  android:layout\_height="match\_parent"  android:text="Hello World!"  android:layout\_weight="0.333" />   <ImageView  android:layout\_width="0dp"  android:layout\_height="match\_parent"  android:contentDescription="Hello World!"  android:src="@mipmap/ic\_launcher"  android:scaleType="centerCrop"  android:layout\_weight="0.333"  android:layout\_marginStart="150dp"  android:layout\_marginLeft="150dp" />  </LinearLayout> |

Листинг ‑ Код для labs/01/linear/15.png

И получаем вот это.

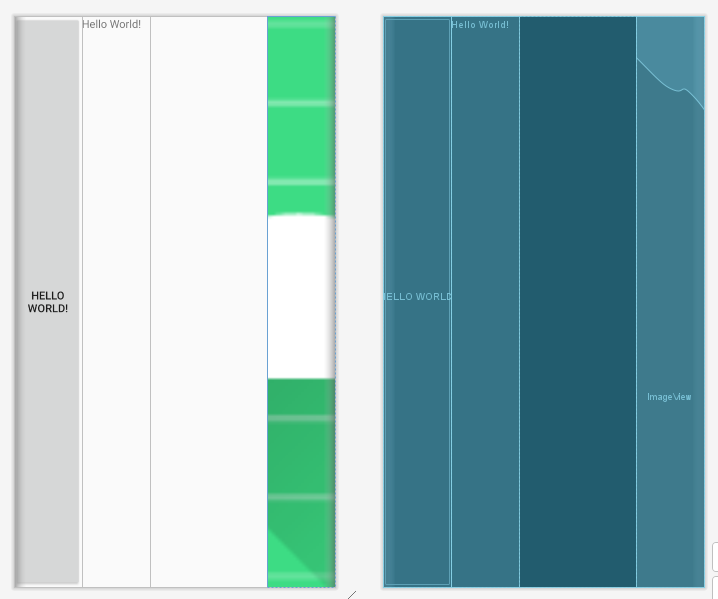


Рис. 3‑2 Результат

Строчка xml:android=”<http://schemas.android.com/apk/res/android>” задает пространство имен android для атрибутов, используемых дальше.

Для того, чтобы сказать layout’у, что нам надо расположить элементы горизонтально, мы задаем соответствующий android:orientation. Он все равно является обязательным.

Атрибуты android:layout\_width и android:layout\_height обязательны, иначе летят ошибки. Значение wrap\_content заставляет элемент подстроиться под потребности своего содержимого, а match\_parent – под свободное место в контейнере. В полях android:layout\_width я выставил 0dp, хотя успешно работает любое значение, если его выставить всем трем View (кроме match\_parent, из-за него первые два занимают честно 1/3 экрана и оказываются больше третьего). Одинаковый android:layout\_weight заставляет все View равномерно распределить свободное пространство между собой.

Атрибут android:contentDescription является обязательным, и он вроде как нужен для accessibility. Я так понимаю, что он может быть использован, например, screen reader’ми для контента, у которого изначально не предусмотрено текстовое представление. android:scaleType задает режим наложения картинки. В данном случае она растягивается пропорционально так, чтобы заполнить все место, при этом располагаясь в центре.

В задании еще спрашивают про android:gravity, который говорит View, где оно должно располагать свое содержимое (center, например, заставит его кучковаться в центре), и про android:layout\_gravity, которое задает позиционирование своего View внутри родителя. Последнее лучше показать картинкой, а то там не очень понятно будет.

|  |
| --- |
| *<?*xml version="1.0" encoding="utf-8"*?>* <androidx.appcompat.widget.LinearLayoutCompat xmlns:android="http://schemas.android.com/apk/res/android"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:orientation="vertical">   <Button  android:layout\_width="wrap\_content"  android:layout\_height="50dp"  android:text="Hello World!"  android:layout\_gravity="center" /></androidx.appcompat.widget.LinearLayoutCompat> |

Листинг ‑ Код для android:layout\_gravity

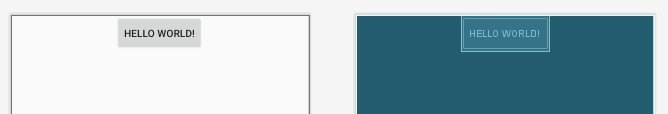


Рис. 3‑3 Проверка android:layout\_gravity

## 02.png

Хотим получить вот такое.

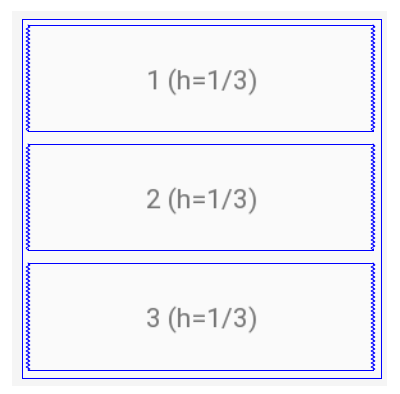


Рис. 3‑4 labs/01/linear/02.png

Делаем так.

|  |
| --- |
| *<?*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">   <Button  android:layout\_width="match\_parent"  android:layout\_height="0dp"  android:text="Hello World!"  android:layout\_weight="1" />   <TextView  android:layout\_width="match\_parent"  android:layout\_height="0dp"  android:text="Hello World!"  android:layout\_weight="1" />   <ImageView  android:layout\_width="match\_parent"  android:layout\_height="0dp"  android:contentDescription="Hello World!"  android:src="@mipmap/ic\_launcher"  android:scaleType="centerCrop"  android:layout\_weight="1" />  </LinearLayout> |

Листинг ‑ Код для labs/01/linear/02.png

И получаем вот это.

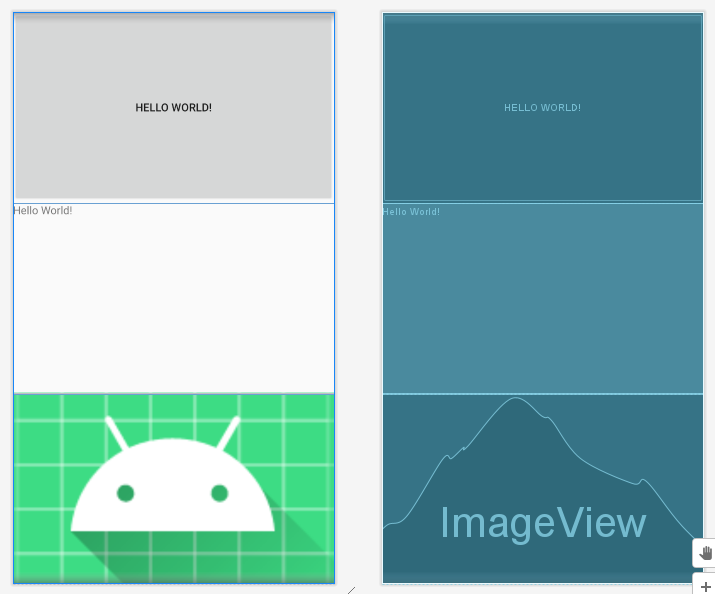


Рис. 3‑5 Результат

Тут все, думаю, очевидно. Я просто распределяю свободное пространство равномерно, задав всем один и тот же android:layout\_weight.

## 06.png

Хотим получить вот такое.

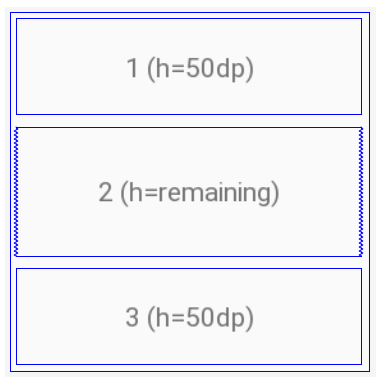


Рис. 3‑6 labs/01/linear/06.png

Делаем так.

|  |
| --- |
| *<?*xml version="1.0" encoding="utf-8"*?>* <androidx.appcompat.widget.LinearLayoutCompat xmlns:android="http://schemas.android.com/apk/res/android"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:orientation="vertical">   <Button  android:layout\_width="match\_parent"  android:layout\_height="50dp"  android:text="Hello World!" />   <TextView  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_weight="1"  android:text="Hello World!" />   <ImageView  android:layout\_width="match\_parent"  android:layout\_height="50dp"  android:contentDescription="Hello World!"  android:src="@mipmap/ic\_launcher"   android:scaleType="centerCrop" />  </androidx.appcompat.widget.LinearLayoutCompat> |

Листинг 3‑4 Код для labs/01/linear/06.png

И получаем вот это.

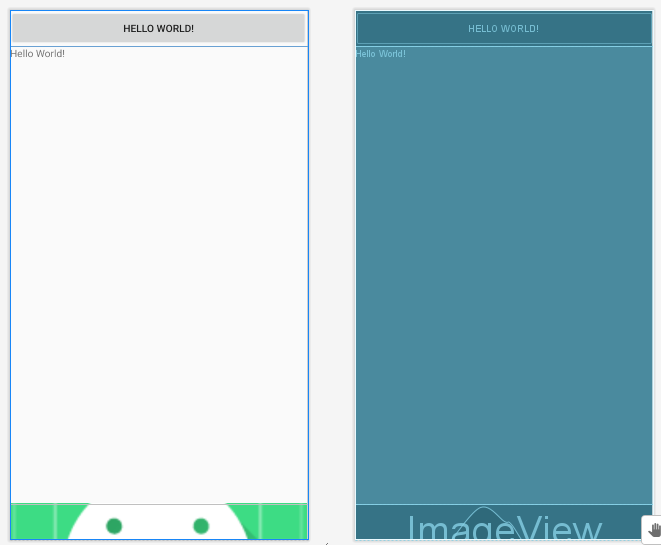


Рис. 3‑7 Результат

Для того, чтобы сказать layout’у, что нам надо расположить элементы вертикально, мы задаем соответствующий android:orientation.

Атрибуты android:layout\_width и android:layout\_height обязательны, иначе летят ошибки. Значение wrap\_content заставляет элемент подстроиться под потребности своего содержимого, а match\_parent – под свободное место в контейнере. Правда, во втором TextView значение android:layout\_height не играет роли, там действует android:layout\_weight, и так как у нас размеры остальных View заданы жестко, а других View с весами нет, то второй элемент заполняет все доступное место. Изначально я выбрал случайно LinearLayoutCompat вместо LinearLayout и решил оставить «как есть», потому что это не особо важно для нас. В LinearLayout высвечивается предупреждение на данном атрибуте, и вместо wrap\_content советуют просто ставить 0dp «для лучшей производительности».

# Задача 2. ConstraintLayout

## 15.png

От нас требуется сделать то же самое, но через ConstraintLayout.

|  |
| --- |
| *<?*xml version="1.0" encoding="utf-8"*?>* <androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  xmlns:app="http://schemas.android.com/apk/res-auto">   <Button  android:id="@+id/first"  android:layout\_width="0dp"  android:layout\_height="match\_parent"  android:text="Hello World!"  app:layout\_constraintLeft\_toLeftOf="parent"  app:layout\_constraintRight\_toLeftOf="@id/intermediate"  app:layout\_constraintHorizontal\_weight="0.333" />   <TextView  android:id="@+id/intermediate"  android:layout\_width="0dp"  android:layout\_height="match\_parent"  android:text="Hello World!"  app:layout\_constraintLeft\_toRightOf="@id/first"  app:layout\_constraintRight\_toLeftOf="@id/last"  app:layout\_constraintHorizontal\_weight="0.333" />   <ImageView  android:id="@+id/last"  android:layout\_width="0dp"  android:layout\_height="match\_parent"  android:contentDescription="Hello World!"  android:src="@mipmap/ic\_launcher"  android:scaleType="centerCrop"  app:layout\_constraintLeft\_toRightOf="@id/intermediate"  app:layout\_constraintRight\_toRightOf="parent"  app:layout\_constraintHorizontal\_weight="0.333"   android:layout\_marginLeft="150dp"  android:layout\_marginStart="150dp" />  </androidx.constraintlayout.widget.ConstraintLayout> |

Листинг ‑ Используем ConstraintLayout

Прежде всего, обратим внимание, что у нас появился новый неймспейс: xmlns:app=”<http://schemas.android.com/apk/res-auto>”.

Теперь мы задаем уникальные идентификаторы для всех элементов и соединяем их в т. н. «цепь» (у всех элементов заданы двунаправленные constraint’ы). Делается это через app:layout\_constraint\*\_to\*Of. В цепи можно попросить View вычислять свои размеры через веса при помощи app:layout\_constraintHorizontal\_weight. При этом всем элементам я задаю ширину в 0dp. Если попробовать им задать одинаковое большое значение (например, 1000dp), то на первый взгляд покажется, что будет все то же самое, но, если приглядеться, можно будет заметить, что боковые элементы оказываются несколько «вытесненными» за границы экрана.



Листинг ‑ Эффект «вытеснения»

В задании просят описать значения еще нескольких атрибутов. Так, например, app:layout\_constraintDimensionRatio позволяет задать элементу пропорциональную зависимость между его высотой и шириной, а затем он будет придерживаться заданных пропорций при изменении количества доступного пространства. В каком-нибудь CSS мы бы стали делать что-то вроде padding: 100% в подобном случае, а тут у нас есть более удобное средство.

## 02.png

Теперь ко второму.

|  |
| --- |
| *<?*xml version="1.0" encoding="utf-8"*?>* <androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  xmlns:app="http://schemas.android.com/apk/res-auto">   <Button  android:id="@+id/first"  android:layout\_width="match\_parent"  android:layout\_height="0dp"  android:text="Hello World!"   app:layout\_constraintTop\_toTopOf="parent"  app:layout\_constraintBottom\_toTopOf="@id/intermediate"  app:layout\_constraintVertical\_weight="1" />   <TextView  android:id="@+id/intermediate"  android:layout\_width="match\_parent"  android:layout\_height="0dp"  android:text="Hello World!"  app:layout\_constraintTop\_toBottomOf="@id/first"  app:layout\_constraintBottom\_toTopOf="@id/last"  app:layout\_constraintVertical\_weight="1" />   <ImageView  android:id="@+id/last"  android:layout\_width="match\_parent"  android:layout\_height="0dp"  android:contentDescription="Hello World!"  android:src="@mipmap/ic\_launcher"  android:scaleType="centerCrop"  app:layout\_constraintTop\_toBottomOf="@id/intermediate"  app:layout\_constraintBottom\_toBottomOf="parent"  app:layout\_constraintVertical\_weight="1" />  </androidx.constraintlayout.widget.ConstraintLayout> |

Листинг ‑ Используем ConstraintLayout

Тут все аналогично, только теперь мы не задаем никаких margin’ов.

## 06.png

Погнали.

|  |
| --- |
| *<?*xml version="1.0" encoding="utf-8"*?>* <androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  xmlns:app="http://schemas.android.com/apk/res-auto">   <Button  android:id="@+id/first"  android:layout\_width="match\_parent"  android:layout\_height="50dp"  android:text="Hello World!"  app:layout\_constraintTop\_toTopOf="parent" />   <TextView  android:layout\_width="match\_parent"  android:layout\_height="0dp"  android:text="Hello World!"  app:layout\_constraintTop\_toBottomOf="@id/first"  app:layout\_constraintBottom\_toTopOf="@id/last" />   <ImageView  android:id="@+id/last"  android:layout\_width="match\_parent"  android:layout\_height="50dp"  android:contentDescription="Hello World!"  android:src="@mipmap/ic\_launcher"  android:scaleType="centerCrop"  app:layout\_constraintBottom\_toBottomOf="parent" />  </androidx.constraintlayout.widget.ConstraintLayout> |

Листинг 4‑4 Используем ConstraintLayout

Теперь мы задаем уникальные идентификаторы для верхнего и нижнего элементов, а среднему говорим растянуться так, чтобы своими сторонами он касался сторон соседей. Для TextView посередине мы задаем android:layout\_height в 0dp, из-за чего его высота вычисляется динамически constraint’ми (это MATCH\_CONSTRAINT).

# Задача 3. ConstraintLayout

## 15.png

Теперь мы верстаем вот это.

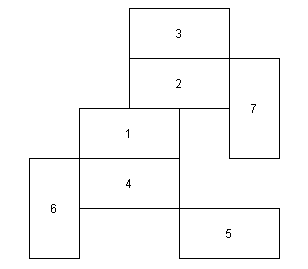


Рис. 5‑1 labs/01/constraint/lab01\_constraint\_v15.png

Можно, например, напихать кучу guideline’ов и получится вот так.

|  |
| --- |
| *<?*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">   <androidx.constraintlayout.widget.ConstraintLayout  app:layout\_constraintDimensionRatio="1:1"  android:layout\_width="0dp"  android:layout\_height="0dp"  app:layout\_constraintTop\_toTopOf="parent"  app:layout\_constraintLeft\_toLeftOf="parent"  app:layout\_constraintRight\_toRightOf="parent"  app:layout\_constraintBottom\_toBottomOf="parent"  android:background="#eeeeee">   <androidx.constraintlayout.widget.Guideline  android:id="@+id/guideline\_horizontal\_20"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:orientation="horizontal"  app:layout\_constraintGuide\_percent="0.2"/>  <androidx.constraintlayout.widget.Guideline  android:id="@+id/guideline\_horizontal\_40"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:orientation="horizontal"  app:layout\_constraintGuide\_percent="0.4"/>  <androidx.constraintlayout.widget.Guideline  android:id="@+id/guideline\_horizontal\_60"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:orientation="horizontal"  app:layout\_constraintGuide\_percent="0.6"/>  <androidx.constraintlayout.widget.Guideline  android:id="@+id/guideline\_horizontal\_80"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:orientation="horizontal"  app:layout\_constraintGuide\_percent="0.8"/>   <androidx.constraintlayout.widget.Guideline  android:id="@+id/guideline\_vertical\_20"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:orientation="vertical"  app:layout\_constraintGuide\_percent="0.2"/>  <androidx.constraintlayout.widget.Guideline  android:id="@+id/guideline\_vertical\_40"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:orientation="vertical"  app:layout\_constraintGuide\_percent="0.4"/>  <androidx.constraintlayout.widget.Guideline  android:id="@+id/guideline\_vertical\_60"  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/guideline\_vertical\_80"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:orientation="vertical"  app:layout\_constraintGuide\_percent="0.8"/>   <Button  android:layout\_width="0dp"  android:layout\_height="0dp"  android:text="Hello World!"  app:layout\_constraintLeft\_toRightOf="@id/guideline\_vertical\_20"  app:layout\_constraintTop\_toBottomOf="@id/guideline\_horizontal\_40"  app:layout\_constraintRight\_toRightOf="@id/guideline\_vertical\_60"  app:layout\_constraintBottom\_toBottomOf="@id/guideline\_horizontal\_60" />   <TextView  android:layout\_width="0dp"  android:layout\_height="0dp"  android:text="Hello World!"  android:background="#ffbaba"  android:gravity="center"  app:layout\_constraintLeft\_toRightOf="@id/guideline\_vertical\_40"  app:layout\_constraintTop\_toBottomOf="@id/guideline\_horizontal\_20"  app:layout\_constraintRight\_toRightOf="@id/guideline\_vertical\_80"  app:layout\_constraintBottom\_toBottomOf="@id/guideline\_horizontal\_40" />   <ImageView  android:layout\_width="0dp"  android:layout\_height="0dp"  android:contentDescription="Hello World!"  android:src="@mipmap/ic\_launcher"  android:scaleType="centerCrop"  app:layout\_constraintLeft\_toRightOf="@id/guideline\_vertical\_40"  app:layout\_constraintTop\_toTopOf="parent"  app:layout\_constraintRight\_toRightOf="@id/guideline\_vertical\_80"  app:layout\_constraintBottom\_toBottomOf="@id/guideline\_horizontal\_20" />   <CheckBox  android:layout\_width="0dp"  android:layout\_height="0dp"  android:text="Hello World!"  android:background="#ddffff"  app:layout\_constraintLeft\_toRightOf="@id/guideline\_vertical\_20"  app:layout\_constraintTop\_toBottomOf="@id/guideline\_horizontal\_60"  app:layout\_constraintRight\_toRightOf="@id/guideline\_vertical\_60"  app:layout\_constraintBottom\_toBottomOf="@id/guideline\_horizontal\_80" />   <androidx.appcompat.widget.SwitchCompat  android:layout\_width="0dp"  android:layout\_height="0dp"  android:text="Hello World!"  android:background="#ddffdd"  app:layout\_constraintLeft\_toRightOf="@id/guideline\_vertical\_60"  app:layout\_constraintTop\_toBottomOf="@id/guideline\_horizontal\_80"  app:layout\_constraintRight\_toRightOf="parent"  app:layout\_constraintBottom\_toBottomOf="parent" />   <ImageView  android:layout\_width="0dp"  android:layout\_height="0dp"  android:contentDescription="Hello World!"  android:src="@mipmap/ic\_launcher\_round"  android:scaleType="fitXY"  android:background="#000"  app:layout\_constraintLeft\_toLeftOf="parent"  app:layout\_constraintTop\_toBottomOf="@id/guideline\_horizontal\_60"  app:layout\_constraintRight\_toRightOf="@id/guideline\_vertical\_20"  app:layout\_constraintBottom\_toBottomOf="parent" />   <Button  android:layout\_width="0dp"  android:layout\_height="0dp"  android:text="Hello World!"  app:layout\_constraintLeft\_toRightOf="@id/guideline\_vertical\_80"  app:layout\_constraintTop\_toBottomOf="@id/guideline\_horizontal\_20"  app:layout\_constraintRight\_toRightOf="parent"  app:layout\_constraintBottom\_toBottomOf="@id/guideline\_horizontal\_60" />   </androidx.constraintlayout.widget.ConstraintLayout>  </androidx.constraintlayout.widget.ConstraintLayout> |

Листинг ‑ Куча guideline’ов

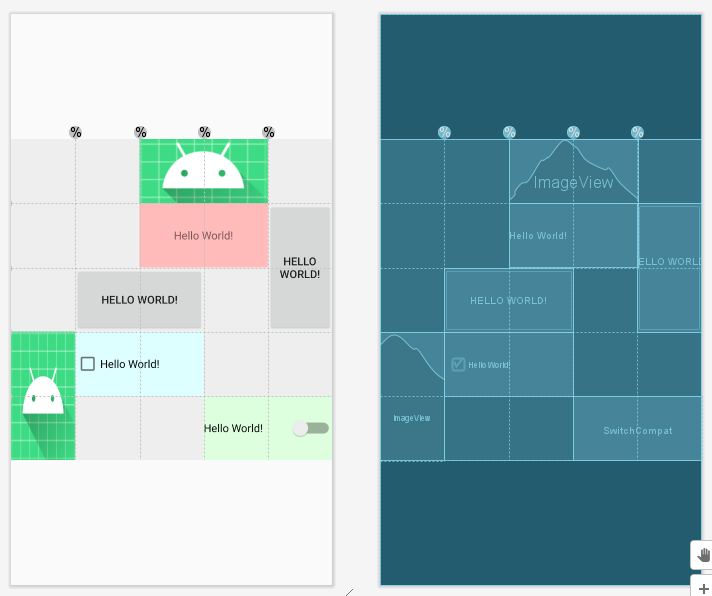


Рис. 5‑2 Визуализация

Хитрость с квадратом заключается в использовании app:layout\_constraintDimensionRatio="1:1" и задании ограничений по краям экрана. Ширина и высота заданы как 0dp, что заставляет их быть вычисленными динамически.

Разбиение на зоны я сделал, добавив guideline’ы на места разделов и ограничив границы каждого View руками.

## 06.png

И последнее.

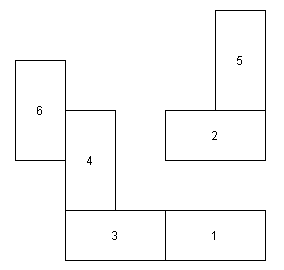


Рис. 5‑3 labs/01/constraint/lab01\_constraint\_v06.png

|  |
| --- |
| *<?*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">   <androidx.constraintlayout.widget.ConstraintLayout  app:layout\_constraintDimensionRatio="1:1"  android:layout\_width="0dp"  android:layout\_height="0dp"  app:layout\_constraintTop\_toTopOf="parent"  app:layout\_constraintLeft\_toLeftOf="parent"  app:layout\_constraintRight\_toRightOf="parent"  app:layout\_constraintBottom\_toBottomOf="parent"  android:background="#eeeeee">   <androidx.constraintlayout.widget.Guideline  android:id="@+id/guideline\_horizontal\_20"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:orientation="horizontal"  app:layout\_constraintGuide\_percent="0.2"/>  <androidx.constraintlayout.widget.Guideline  android:id="@+id/guideline\_horizontal\_40"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:orientation="horizontal"  app:layout\_constraintGuide\_percent="0.4"/>  <androidx.constraintlayout.widget.Guideline  android:id="@+id/guideline\_horizontal\_60"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:orientation="horizontal"  app:layout\_constraintGuide\_percent="0.6"/>  <androidx.constraintlayout.widget.Guideline  android:id="@+id/guideline\_horizontal\_80"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:orientation="horizontal"  app:layout\_constraintGuide\_percent="0.8"/>   <androidx.constraintlayout.widget.Guideline  android:id="@+id/guideline\_vertical\_20"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:orientation="vertical"  app:layout\_constraintGuide\_percent="0.2"/>  <androidx.constraintlayout.widget.Guideline  android:id="@+id/guideline\_vertical\_40"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:orientation="vertical"  app:layout\_constraintGuide\_percent="0.4"/>  <androidx.constraintlayout.widget.Guideline  android:id="@+id/guideline\_vertical\_60"  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/guideline\_vertical\_80"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:orientation="vertical"  app:layout\_constraintGuide\_percent="0.8"/>   <Button  android:layout\_width="0dp"  android:layout\_height="0dp"  android:text="Hello World!"  app:layout\_constraintLeft\_toRightOf="@id/guideline\_vertical\_60"  app:layout\_constraintTop\_toBottomOf="@id/guideline\_horizontal\_80"  app:layout\_constraintRight\_toRightOf="parent"  app:layout\_constraintBottom\_toBottomOf="parent" />   <TextView  android:layout\_width="0dp"  android:layout\_height="0dp"  android:text="Hello World!"  android:background="#ffbaba"  android:gravity="center"  app:layout\_constraintLeft\_toRightOf="@id/guideline\_vertical\_60"  app:layout\_constraintTop\_toBottomOf="@id/guideline\_horizontal\_40"  app:layout\_constraintRight\_toRightOf="parent"  app:layout\_constraintBottom\_toBottomOf="@id/guideline\_horizontal\_60" />   <ImageView  android:layout\_width="0dp"  android:layout\_height="0dp"  android:contentDescription="Hello World!"  android:src="@mipmap/ic\_launcher"  android:scaleType="centerCrop"  app:layout\_constraintLeft\_toRightOf="@id/guideline\_vertical\_20"  app:layout\_constraintTop\_toBottomOf="@id/guideline\_horizontal\_80"  app:layout\_constraintRight\_toRightOf="@id/guideline\_vertical\_60"  app:layout\_constraintBottom\_toBottomOf="parent"/>   <CheckBox  android:layout\_width="0dp"  android:layout\_height="0dp"  android:text="Hello World!"  android:background="#ddffff"  app:layout\_constraintLeft\_toRightOf="@id/guideline\_vertical\_20"  app:layout\_constraintTop\_toBottomOf="@id/guideline\_horizontal\_40"  app:layout\_constraintRight\_toRightOf="@id/guideline\_vertical\_40"  app:layout\_constraintBottom\_toBottomOf="@id/guideline\_horizontal\_80"/>   <androidx.appcompat.widget.SwitchCompat  android:layout\_width="0dp"  android:layout\_height="0dp"  android:text="Hello World!"  android:background="#ddffdd"  app:layout\_constraintLeft\_toRightOf="@id/guideline\_vertical\_80"  app:layout\_constraintTop\_toTopOf="parent"  app:layout\_constraintRight\_toRightOf="parent"  app:layout\_constraintBottom\_toBottomOf="@id/guideline\_horizontal\_40" />   <ImageView  android:layout\_width="0dp"  android:layout\_height="0dp"  android:contentDescription="Hello World!"  android:src="@mipmap/ic\_launcher\_round"  android:scaleType="fitXY"  android:background="#000"  app:layout\_constraintLeft\_toLeftOf="parent"  app:layout\_constraintTop\_toBottomOf="@id/guideline\_horizontal\_20"  app:layout\_constraintRight\_toRightOf="@id/guideline\_vertical\_20"  app:layout\_constraintBottom\_toBottomOf="@id/guideline\_horizontal\_60"/>   </androidx.constraintlayout.widget.ConstraintLayout>  </androidx.constraintlayout.widget.ConstraintLayout> |

Листинг 5‑2 Куча guideline’ов

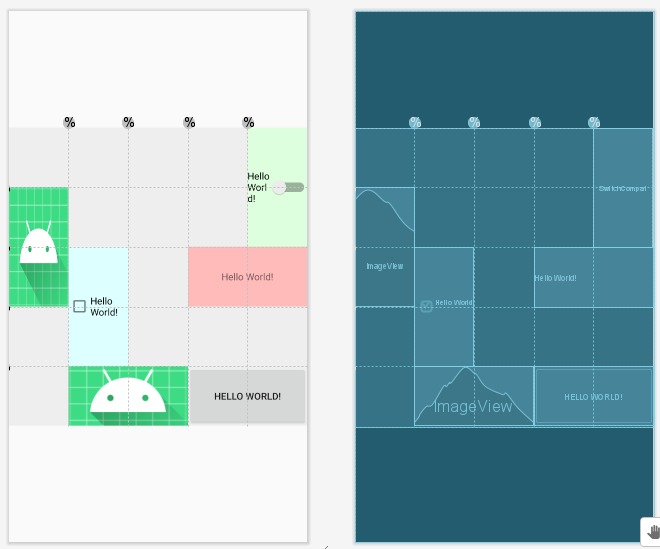


Рис. 5‑4 Визуализация

# Выводы

Что ж, как нам сказали писать выводы на лекции… Мы:

* Ознакомились со средой разработки Android Studio
* Изучили основные принципы верстки layout с использованием View и ViewGroup
* Изучили основные возможности и свойства LinearLayout
* Изучили основные возможности и свойства ConstraintLayout

И теперь при необходимости вывести ряд из нескольких View будем пользоваться LinearLayout, а для случаев более хитрых размещений будем прибегать к помощи ConstraintLayout.