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 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Е.К. Борисов

(подпись)

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

(подпись)

“ ” 2021 г.

Санкт-Петербург

2021

Оглавление

[Цели: 3](#_Toc84169649)

[Введение 3](#_Toc84169650)

[LinearLayout 4](#_Toc84169651)

[Lab\_1\_5 4](#_Toc84169652)

[Lab\_1\_14\_1 10](#_Toc84169653)

[ConstraintLayout 15](#_Toc84169654)

[Lab\_2\_5 15](#_Toc84169655)

[Lab\_2\_14 18](#_Toc84169656)

[Lab\_3\_5 21](#_Toc84169657)

[Вывод 25](#_Toc84169658)

[Список источников: 25](#_Toc84169659)

# 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

# Введение

Ресурсы - один из основных компонентов, с которыми вам придется работать очень часто. В Android принято держать некоторые объекты - изображения, строковые константы, цвета, анимацию, стили и т.п. за пределами исходного кода. Система поддерживает хранение ресурсов в отдельных файлах. Ресурсы легче поддерживать, обновлять, редактировать.

Ресурсы в Android являются декларативными. В основном ресурсы хранятся в виде XML-файлов в каталоге *res* с подкаталогами *values, drawable-ldpi, drawable-mdpi, drawable-hdpi, layout*, но также бывают и другие типы ресурсов. Класс R содержит ссылки на все ресурсы проекта.

Ресурс разметки формы (layout resource) - это ключевой тип ресурсов, применяемый при программировании пользовательских интерфейсов в Android. Каждый ресурс, описывающий разметку, хранится в отдельном файле каталога *res/layout*. Имя файла без расширения выступает как идентификатор ресурса.

Все описываемые разметки являются подклассами ViewGroup и наследуют свойства, определённые в классе View.

Индивидуальный вариант: 5.

*View*

* View объекты являются основными строительными элементами элементов пользовательского интерфейса (UI) в Android.
* View - это простой прямоугольник, который реагирует на действия пользователя. Примерами являются EditText, Button, CheckBox и т.д.
* View относится к классу android.view.View, который является базовым классом всех классов пользовательского интерфейса.

*ViewGroup*

* ViewGroup - невидимый контейнер. Он содержит View и ViewGroup
* Например, LinearLayout - это ViewGroup, который содержит кнопку (вид) и другие макеты.
* ViewGroup - базовый класс для макетов.

# LinearLayout

Макет LinearLayout представлен двумя вариантами - Horizontal и Vertical. Макет LinearLayout выравнивает все дочерние объекты в одном направлении — вертикально или горизонтально. Направление задается при помощи атрибута ориентации android:orientation:

## Lab\_1\_5

Создать layout ресурс для следующего макета экрана

|  |
| --- |
|  |
| Рис. 1.1 Макет lab1\_1. |

Полученный ресурс:

|  |
| --- |
| Листинг 1.1 Ресурс разметки lab\_1\_5 |
| <?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:orientation="horizontal"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"   >   <RadioGroup  android:layout\_width="@dimen/in\_dp"  android:gravity="center"  android:layout\_height="match\_parent">   <RadioButton  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:text="@string/a" />   <RadioButton  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:text="@string/b" />   <RadioButton  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:text="@string/c" />  </RadioGroup>   <TextView  android:background="@drawable/gradient"  android:layout\_width="0dp"  android:layout\_height="match\_parent"  android:layout\_weight="1"  android:gravity="center"  android:text="@string/lab1\_1"  android:textColor="@color/black"  android:textSize="50sp" />   <Button  android:layout\_width="50dp"  android:layout\_height="match\_parent"  android:background=""  android:text="@string/button1"  android:textColor="@color/black"  android:textSize="20sp" />  </LinearLayout> |

|  |
| --- |
| Листинг 1.2 gradient.xml |
| <?xml version="1.0" encoding="utf-8"?> <shape xmlns:android="http://schemas.android.com/apk/res/android"  android:shape="rectangle" >   <gradient  android:angle="90"  android:endColor="#F8F8F8"  android:startColor="#888787" />  </shape> |

|  |
| --- |
| Дерево компонентов |
| LinearLayout  RadioButton1  RadioButton2  RadioButton3  Button  TextView  RadioGroup |

Корневой элемент Linear Layout:

*xmlns:tools=*[*http://schemas.android.com/tools*](http://schemas.android.com/tools) *-* позволяет среде разработки правильно отобразить компоненты для просмотра в режиме дизайна.

*android:orientation="horizontal"* - атрибута ориентации о котором было рассказано ранее.

*android:layout\_width="match\_parent"* и *android:layout\_height="match\_parent"* – ширина и высота элемента, могут задаваться в абсолютных значениях, а могут быть следующими:

* match\_parent - максимально возможная ширина или высота в пределах родителя
* wrap\_content - ширина или высота определяется по содержимому элемента

RadioGroup – компонент, использующийся в качестве контейнера для radioButton. Наследуется от Linear Layout. Может включать не только RadioButton, но и другие элементы, например, TextView. Переключатели работают в своём контейнере, т.е. если у вас два контейнера RadioGroup, то переключатели из одного контейнера не влияют на поведение переключателей второго контейнера.

*android:layout\_width="@dimen/in\_dp" –* Ширина заданная в соответствии с заданием 50dp.

*android:layout\_height="match\_parent"* – Максимально возможная высота в пределах родителя.

*android:gravity="center" –* Задает расположение содержимого внутри компонента. В данной случае в центре.

RadioButton – Переключатели, позволяют пользователю выбрать один вариант из набора. Ширина задана match\_parent, высота задана wrap-content.

TextView - предназначен для отображения текста без возможности редактирования его пользователем.

*android:background="@drawable/gradient"* – Задний фон компонента, заданный в виде прямоугольника с градиентом.

*android:layout\_width="0dp"* – Ширина match\_constraint, который выстраивает размер на основе компонентов, которые заданы по данной плоскости.

*android:layout\_height="match\_parent"* – Высота, заданная match\_parent.

*android:layout\_weight="1"* – Индивидуальный вес элемента, который помогает заполнить оставшееся пространство в родительском представлении.

*android:textColor="@color/black"* – Цвет текстка в элементе.

*android:textSize="50sp"* – Размер текста в элементе.

Button – кнопка, размеры заданы в соответствии с заданием, увеличен размер текста в кнопке и убран задний фон.

|  |
| --- |
|  |
| Рис.1.2 Получившийся layout. |

## Lab\_1\_14\_1

Создать layout ресурс для следующего макета экрана

|  |
| --- |
|  |
| Рис.1.3 Макет lab\_2 |

|  |
| --- |
| Листинг 1.3 Ресурс разметки lab\_1\_14\_1 |
| <?xml version="1.0" encoding="utf-8"?> <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  android:orientation="vertical"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent">   <ImageView  android:layout\_width="match\_parent"  android:layout\_height="0dp"  android:layout\_weight="1"  android:src="@drawable/logo"  android:contentDescription="@string/logo"/>   <EditText  android:layout\_width="match\_parent"  android:layout\_height="0dp"  android:layout\_weight="1"  android:layout\_marginTop="@dimen/margin"  android:inputType="textEmailAddress"  android:autofillHints="emailAddress"  android:hint="@string/type\_email"  android:gravity="start"  android:textColorHint="@color/black"/>   <EditText  android:layout\_width="match\_parent"  android:layout\_height="0dp"  android:layout\_weight="1"  android:autofillHints="password"  android:gravity="start"  android:hint="@string/type\_password"  android:textColorHint="@color/black"  android:inputType="textPassword" />  </LinearLayout> |

|  |
| --- |
| Дерево компонентов |
| LinearLayout  EditText (Password)  EditText (Email)  ImageView |

Чтобы расположить все 3 элемента в одинаковой пропорции, был использован атребут веса, в котором всем трем элементам соответствует одинаковое значение “1”. Высота всех элементов растянута с помощью match\_constraint, ширина с помощью match\_parent.

ImageView - предназначен для отображения изображений.

*android:src="@drawable/logo"* – Атрибут, использующийся для загрузки изображения в ImageView.

*android:contentDescription="@string/logo"* - Метка контента, позволяющая пользователям с ограниченными возможностями распознавать элементы интерфейса при помощи программы чтения с экрана.

EditText - это текстовое поле для пользовательского ввода, которое используется, если необходимо редактирование текста. Является наследником TextView.

Email:

У элемента E-mail используется атрибут *android:inputType="textEmailAddress"*. В этом случае на клавиатуре появляется дополнительная клавиша с символом @, который обязательно используется в любом электронном адресе.

*android:layout\_marginTop="@dimen/margin"* – Атрибут отступа от элемента сверху, заданный в соответствии с заданием.

*android:autofillHints="emailAddress"* – Атрибут, указывающий, какой тип контента должен быть помещен в данный EditText.

*android:hint="@string/type\_email"* – Добавляет в текстовое поле подсказку, что пользователю нужно вводить, которая пропадает, когда пользователь вводит любой символ.

*android:textColorHint="@color/black"* – Меняет цвет текста подсказки. В данном случае на черный.

Password:

При использовании Password в inputType используется значение textPassword. При вводе текста сначала показывается символ, который заменяется на звёздочку. В остальном используются те же аттрибуты, что и для поля email с измененными значениями.

Вместо атрибута marginTop у второго элемента, можно было бы использовать marginBottom у первого или использовать подкласс View “space”, использующийся для создания дополнительного пространства между компонентами.

|  |
| --- |
| Листинг 1.4 Ресурс разметки lab\_1\_14\_2 |
| <?xml version="1.0" encoding="utf-8"?> <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  android:orientation="vertical"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent">   <ImageView  android:layout\_width="match\_parent"  android:layout\_height="0dp"  android:layout\_weight="1"  android:src="@drawable/logo"  android:contentDescription="@string/logo"/>   <Space  android:layout\_width="match\_parent"  android:layout\_height="@dimen/margin"/>     <EditText  android:layout\_width="match\_parent"  android:layout\_height="0dp"  android:layout\_weight="1"  android:inputType="textEmailAddress"  android:autofillHints="emailAddress"  android:hint="@string/type\_email"  android:gravity="start"  android:textColorHint="@color/black"/>   <EditText  android:layout\_width="match\_parent"  android:layout\_height="0dp"  android:layout\_weight="1"  android:autofillHints="password"  android:gravity="start"  android:hint="@string/type\_password"  android:textColorHint="@color/black"  android:inputType="textPassword" />  </LinearLayout> |

|  |
| --- |
|  |
|  |
|  |
| Рис.1.4 Получившиеся layouts. |

# ConstraintLayout

ConstraintLayout – это макет, который позволяет создавать гибкие и масштабируемые визуальные интерфейсы. Для позиционирования элемента внутри ConstraintLayout необходимо указать ограничения.

Каждое ограничение устанавливает позиционирование элемента либо по горизонтали, либо по вертикали. И для определения позиции элемента в ConstraintLayout необходимо указать как минимум одно ограничение по горизонтали и одно ограничение по вертикали.

Позиционирования может производиться относительно границ самого контейнера ContentLayout (в этом случае ограничение имеет значение parent), либо же относительно любого другого элемента внутри ConstraintLayout, тогда в качестве значения ограничения указывается id этого элемента.

## Lab\_2\_5

Нужно реализовать макет, как в задании lab\_1\_5.

|  |
| --- |
| Листинг 2.1 Ресурс разметки lab\_2\_5 |
| <?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">   <RadioGroup  android:id="@+id/radioGroup1"  android:layout\_width="@dimen/in\_dp"  android:layout\_height="0dp"  android:gravity="center"  app:layout\_constraintStart\_toStartOf="parent"  app:layout\_constraintTop\_toTopOf="parent"  app:layout\_constraintBottom\_toBottomOf="parent">    <RadioButton  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:text="@string/a" />   <RadioButton  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:text="@string/b" />   <RadioButton  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:text="@string/c" />  </RadioGroup>   <TextView  android:id="@+id/textView1"  android:background="@drawable/gradient"  android:layout\_width="0dp"  android:layout\_height="0dp"  android:gravity="center"  android:text="@string/lab1\_1"  android:textColor="@color/black"  android:textSize="50sp"  app:layout\_constraintStart\_toEndOf="@id/radioGroup1"  app:layout\_constraintTop\_toTopOf="parent"  app:layout\_constraintEnd\_toStartOf="@id/button1"  app:layout\_constraintBottom\_toBottomOf="parent"  />   <Button  android:id="@+id/button1"  android:layout\_width="50dp"  android:layout\_height="0dp"  android:background=""  android:text="@string/button1"  android:textColor="@color/black"  android:textSize="20sp"  app:layout\_constraintEnd\_toEndOf="parent"  app:layout\_constraintTop\_toTopOf="parent"  app:layout\_constraintBottom\_toBottomOf="parent"/>   </androidx.constraintlayout.widget.ConstraintLayout> |

Корневым элементов является ConstraintLayout, мы располагаем все дочерние элементы относительно родительского и других дочерних элементов опираясь на их атрибут “id”.

RadioGroup – Вверх, начало и низ элемента совпадает с вверхом, началом и низом родителя.

TextView – Начало элемента совпадает с концом RadioGroup, вверх совпадает с верхом родителя, конец совпадает с началом Button, низ совпадает с низом родителя.

Button – Конец совпадает с концом родителя, вверх совпадает с верхом родителя, низ совпадает с низом родителя.

У всех элементов заменили match\_parent на match\_constraint, так как размер компонента не всегда должен совпадать с размером родителя, также больше не нужно использовать атрибута веса, так как при правильно прописанных ограничениях, компонент и так примет нужные размеры.

|  |
| --- |
|  |
| Рис.2.1 Получившийся layout. |

## Lab\_2\_14

Нужно реализовать макет, как в задании lab\_1\_14.

|  |
| --- |
| Листинг 2.2 Ресурс разметки lab\_2\_14 |
| <?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">   <ImageView  android:id="@+id/image"  android:layout\_width="0dp"  android:layout\_height="0dp"  android:src="@drawable/logo"   android:contentDescription="@string/logo"  app:layout\_constraintTop\_toTopOf="parent"  app:layout\_constraintStart\_toStartOf="parent"  app:layout\_constraintEnd\_toEndOf="parent"  app:layout\_constraintBottom\_toTopOf="@id/email"  />   <EditText  android:id="@+id/email"  android:layout\_width="0dp"  android:layout\_height="0dp"  android:inputType="textEmailAddress"  android:autofillHints="emailAddress"  android:hint="@string/type\_email"  android:gravity="start"  android:textColorHint="@color/black"  android:layout\_marginTop="@dimen/margin"  app:layout\_constraintTop\_toBottomOf="@id/image"  app:layout\_constraintStart\_toStartOf="parent"  app:layout\_constraintEnd\_toEndOf="parent"  app:layout\_constraintBottom\_toTopOf="@id/password"   />   <EditText  android:id="@+id/password"  android:layout\_width="0dp"  android:layout\_height="0dp"  android:autofillHints="password"  android:gravity="start"  android:hint="@string/type\_password"  android:textColorHint="@color/black"  android:inputType="textPassword"  app:layout\_constraintBottom\_toBottomOf="parent"  app:layout\_constraintTop\_toBottomOf="@id/email"  app:layout\_constraintStart\_toStartOf="parent"  app:layout\_constraintEnd\_toEndOf="parent"/>   </androidx.constraintlayout.widget.ConstraintLayout> |

У всех элементов начало и конец документа совпадает с началом и концом родителя.

ImageView – Вверх cовпадает с верхом родителя, низ совпадает с верхом email-а.

EditText(email) – верх совпадает с низом ImageView, низ совпадает с верхом EditText(password).

EditText(password) – верх совпадает с низом EditText(email), низ совпадает с низом документа.

Также, как в первом случае у всех элементов заменили ширину с match\_parent на match\_constraint. В качестве отступа использовали *android:layout\_marginTop="@dimen/margin"* у второго элемента.

|  |
| --- |
|  |
| Рис.2.2 Получившийся layout. |

## Lab\_3\_5

Создать layout ресурс для следующего макета:

|  |
| --- |
|  |
| Рис.3.1 Заданный макет. |

Полученный ресурс представлен ниже:

|  |
| --- |
| Листинг 2.2 Ресурс разметки lab\_3\_5 |
| <?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"  android:background="@drawable/gradient"  xmlns:app="http://schemas.android.com/apk/res-auto">    <androidx.constraintlayout.widget.ConstraintLayout  android:background="#EBE4E4"  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\_toEndOf="parent"  app:layout\_constraintDimensionRatio="1:1">   <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="vertical"  app:layout\_constraintGuide\_percent="0.4" />   <androidx.constraintlayout.widget.Guideline  android:id="@+id/line3"  android:layout\_width="0dp"  android:layout\_height="0dp"  android:orientation="vertical"  app:layout\_constraintGuide\_percent="0.6034063" />   <androidx.constraintlayout.widget.Guideline  android:id="@+id/line4"  android:layout\_width="0dp"  android:layout\_height="0dp"  android:orientation="vertical"  app:layout\_constraintGuide\_percent="0.8" />   <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" />    <ImageView  android:id="@+id/imageView1"  android:layout\_width="0dp"  android:layout\_height="0dp"  android:background="@color/white"  android:src="@drawable/logo"  android:contentDescription="@string/logo"  app:layout\_constraintDimensionRatio="2:1"  app:layout\_constraintStart\_toStartOf="@id/line1"  app:layout\_constraintBottom\_toBottomOf="parent"  app:layout\_constraintEnd\_toEndOf="@id/line3"/>   <ImageView  android:id="@+id/imageView2"  android:layout\_width="0dp"  android:layout\_height="0dp"   android:background="@color/white"  android:contentDescription="@string/logo"  android:src="@drawable/logo"  app:layout\_constraintDimensionRatio="2:1"  app:layout\_constraintEnd\_toStartOf="@id/line2"  app:layout\_constraintStart\_toStartOf="parent"  app:layout\_constraintTop\_toTopOf="parent" />   <ImageView  android:id="@+id/imageView3"  android:layout\_width="0dp"  android:layout\_height="0dp"   android:background="@color/white"  android:contentDescription="@string/logo"  android:src="@drawable/logo"  app:layout\_constraintDimensionRatio="2:1"   app:layout\_constraintEnd\_toEndOf="@id/line3"  app:layout\_constraintStart\_toStartOf="@+id/line1"  app:layout\_constraintTop\_toBottomOf="@id/imageView2" />   <ImageView  android:id="@+id/imageView4"  android:layout\_width="0dp"  android:layout\_height="0dp"   android:background="@color/white"  android:contentDescription="@string/logo"  android:src="@drawable/logo"  app:layout\_constraintDimensionRatio="2:1"   app:layout\_constraintEnd\_toEndOf="@id/line3"  app:layout\_constraintStart\_toStartOf="@+id/line1"  app:layout\_constraintTop\_toBottomOf="@id/imageView3" />    <ImageView  android:id="@+id/imageView5"  android:layout\_width="0dp"  android:layout\_height="0dp"   android:background="@color/white"  android:contentDescription="@string/logo"  android:src="@drawable/logo"   app:layout\_constraintDimensionRatio="1:2"  app:layout\_constraintEnd\_toEndOf="parent"  app:layout\_constraintStart\_toStartOf="@id/line4"  app:layout\_constraintTop\_toBottomOf="@id/imageView6" />   <ImageView  android:id="@+id/imageView6"  android:layout\_width="0dp"  android:layout\_height="0dp"   android:background="@color/white"  android:contentDescription="@string/logo"  android:src="@drawable/logo"  app:layout\_constraintDimensionRatio="1:2"   app:layout\_constraintEnd\_toEndOf="parent"  app:layout\_constraintStart\_toStartOf="@id/line4"  app:layout\_constraintTop\_toTopOf="parent" />      </androidx.constraintlayout.widget.ConstraintLayout> </androidx.constraintlayout.widget.ConstraintLayout> |

Для разработки данного макета были использованы виджеты GuideLine, представляющие собой вспомогательные объекты в виде горизонтальных или вертикальных линий. Расположении линий задается в виде процентов от размера ConstraintLayout, в котором они расположены с помощью атрибута *layout\_constraintGuide\_percent*.

Для получения квадрата, занимаемому максимальную площадь, был использован атрибут *layout\_constraintDimensionRatio,* который используется для задания соотношения сторон (высоты и ширины). Также, так как все дочерние элементы имеют одинаковую площадь и расположены либо вертикально, либо горизонтально, данный атрибут использовался и для всех прочих элементов.

|  |
| --- |
|  |
| Рис.3 Получившийся layout |

# Вывод

В данной лабораторной работе мы познакомились со средой разработки Android Studio. Была изучена работа с двумя с ресурсами разметки LinearLayout и ConstaintLayout путем разработки различных макетов. В процессе написания объектов были рассмотрены различные компоненты такие как: EditText, ImageView, TextView, RadioGroup, RadioButton. Также были использованы cтроковые ресурсы “string.xml” и ресурсы размеров “dimens.xml”, которые помогают ускорить разработку.

**В каких случаях целесообразно использовать LinearLayout, в каких ConstraintLayout?**

LinearLayout следует использовать, если нужно описать простой макет, где есть возможность добиться плоской иерархии layouts. В остальных случаях лучше использовать ConstaintLayout. Использовать ConstaintLayout для создания простого макета - не лучшая идея, так как он решает внутри себя системы уравнений и его использование может снизить производительность и усложнить читабельность.

**Список источников:**

* <https://github.com/andrei-kuznetsov/android-lectures>
* <http://developer.alexanderklimov.ru/android/>
* <https://startandroid.ru/>