МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное автономное образовательное учреждение

высшего образования

**«Национальный исследовательский**

**Нижегородский государственный университет им. Н.И. Лобачевского»**

**Институт информационных технологий, математики и механики**

Направление подготовки «Фундаментальная информатика и информационные технологии»

**ОТЧЕТ**

по лабораторной работе

**Практическое задание для лекции №4**

**Выполнил:** студент группы

381906-2 Углинский Б.С

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Подпись

**Проверил:** Карчков Д.А.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Подпись

Нижний Новгород

2022

Оглавление

[1. Цели работы. 3](#_Toc116060958)

[2. Постановка задачи. 4](#_Toc116060959)

[3. Решение задачи. 5](#_Toc116060960)

[4. Руководство пользователя. 6](#_Toc116060961)

[Приложение. 7](#_Toc116060962)

# 1. Цели работы.

Целью данной лабораторной работы является освоение принципов работы базовых элементов интерфейса Android OS. Необходимо научится правильно настраивать элементы, а также корректно обрабатывать пользовательский ввод с них.

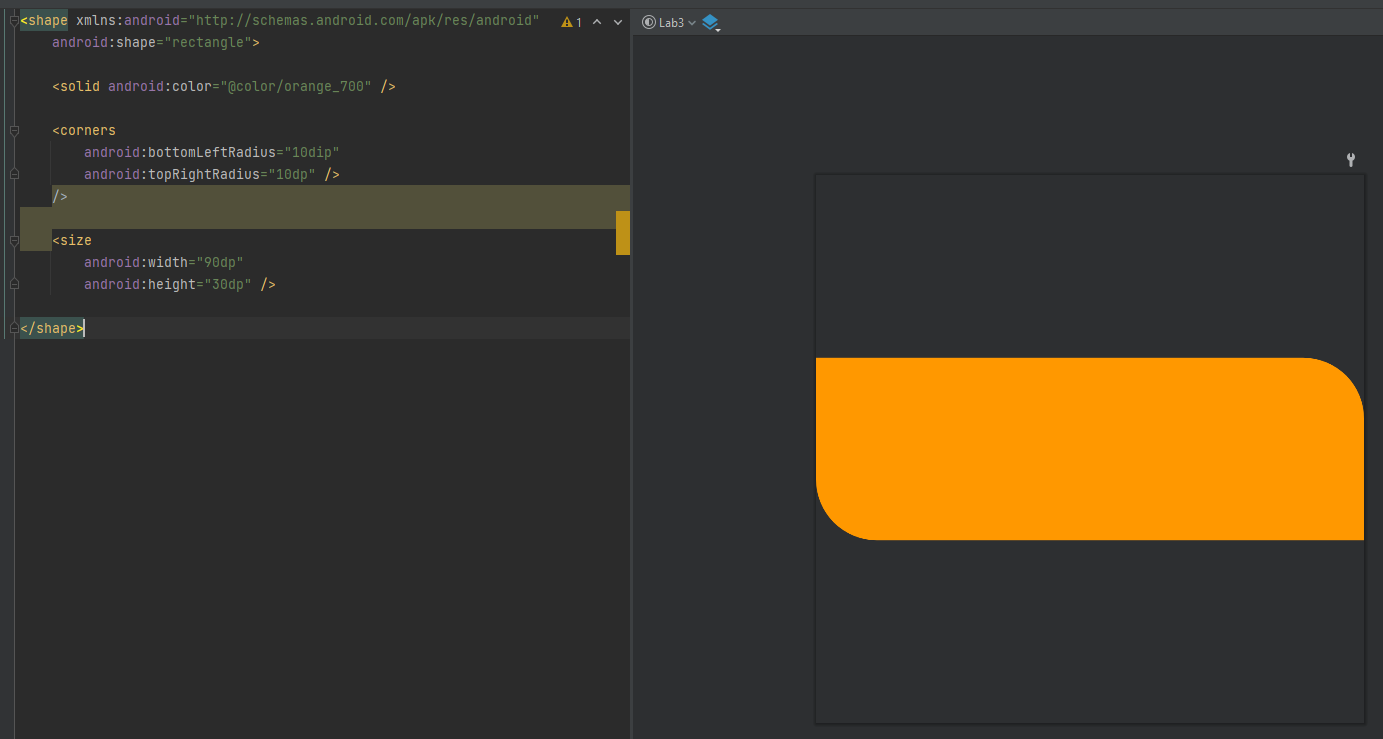
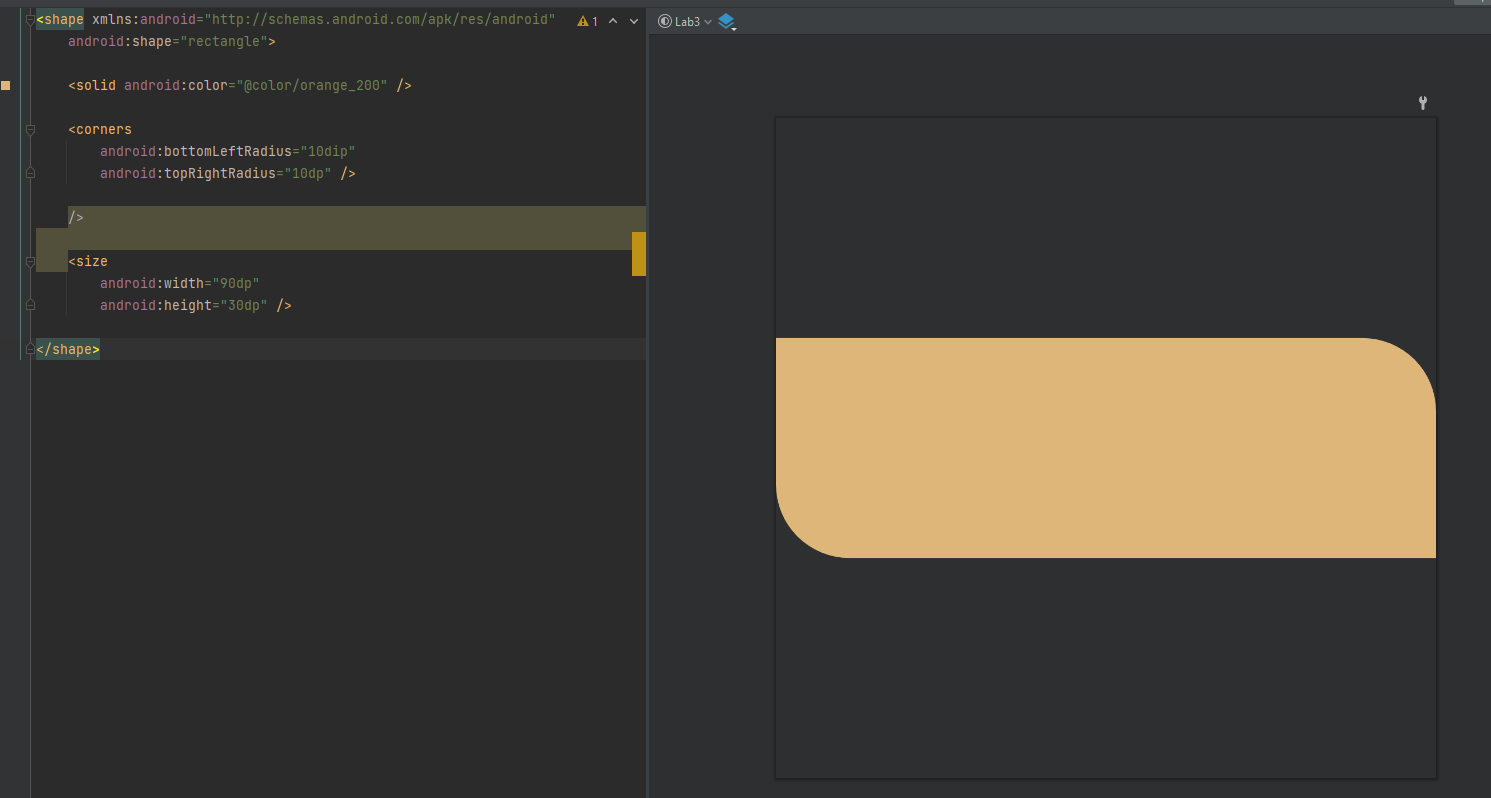
# 2. Постановка задачи.

Для выполнения лабораторной работы были поставлены следующие задачи:

1. Создать текстовое поле для ввода числа с подсказкой.
2. Создать кнопку, которая при нажатии будет изменяться и таким образом показывать пользователю, что она нажата. К кнопке также необходимо добавить текстовое поле, которое текстом дублирует состояние кнопки.
3. Необходимо сделать кнопку, которая будет отображать количество нажатий на неё.
4. Создать селектор даты, данные с которого будут повторятся на текстовое поле рядом.
5. Создать селектор времени, который будет отображать выбранное время в текстовое поле рядом на экране.
6. Сделать поле с выпадающим списком, которое позволяет выбрать один из вариантов списка. Результат выбора пользователя должен отображаться в текстовом поле.
7. Создать переключатель и текстовое поле, в котором будет написано включен ли выключатель или выключен.
8. Реализовать ползунок, который принимает только натуральные значения и при этом вывести его текущее значения в поле с текстом.

# 3. Решение задачи.

1. Чтобы создать поле для ввода числа с подсказкой, достаточно создать элемент EditText и в его свойствах указать android:inputType = ”numberDecimal” (включает ввод только чисел) и android:hint = ”hint” (текст подсказки).
2. Изменение внешнего вида кнопки происходит за счёт смены её заднего фона между заранее заготовленным шаблонами изображений. Для это необходимо создать два **Drawable Resource File** в папке ***drawables*** и настроить каждый из файлов под определённое состояние кнопки.

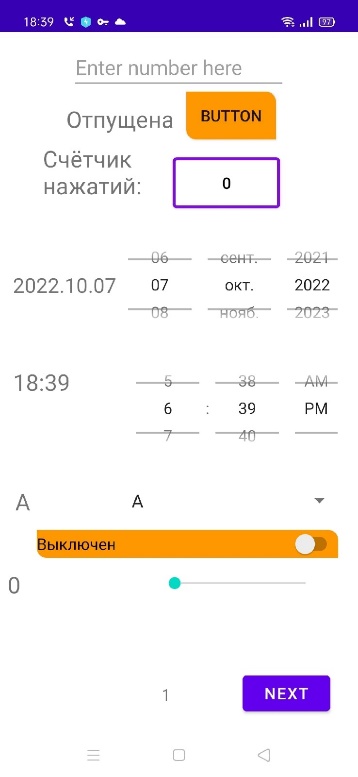


Когда пользователь нажимает на кнопку, то, в зависимости от её текущего состояния, задний фон заменяется на то или иное изображение.

1. Чтобы кнопка отображала количество нажатий на неё, нужно создать счётчик, который будет отслеживать сколько раз на кнопку нажали и просто менять соответственно текст самой кнопки через setText().
2. Селектор даты (DatePicker) обладает большим количество настроек, поэтому необходимо сначала определить какой конкретно внешний вид должен иметь выбор даты, после чего задать необходимые параметры. В нашем случае, тип селектора даты – “spinner”, отображение календаря - выключено, отображение спиннера - включено. Для того, чтобы считывать данные с селектора при изменении его значений необходимо создать event listener, который будет срабатывать каждый раз, когда происходит изменение состояние объекта. При его создании необходимо переопределить функции, отвечающие за обработку изменений, например, OnDateChanged(), после чего привязать этот listener, к соответствующему элементу пользовательского интерфейса.
3. Для таких элементов, как селектор даты (DatePicker), селектор врмени (TimePicker), выпадающий список (Spinner) и ползунок (SeekBar), механизм обработки одинаковый. Для каждого из вышеперечисленных элементов необходимо создать свой отдельный event listener, который будет отвечать за определённое изменение состояния объекта, а именно: OnDateChanged() для DatePicker, onTimeChanged() для TimePicker, OnItemSelected() для Spinner, OnProgressChanged() для SeekBar.
4. Нажатия на переключатель обрабатываются по аналогии с тем, как они обрабатываются в случае простой кнопки. В свойствах элемента в поле android:onClick необходимо указать какая функция будет вызываться при нажатии на переключатель. В данном случае это onSwticherPressed().

# 4. Руководство пользователя.

При запуске приложения пользователь увидит экран со всеми элементами сверху-вниз: поле для ввода числа, кнопка с надписью: “Нажата” или “Отпущена”, кнопка которая считает количество нажатий на неё, селектор даты, селектор времени, выпадающий список, переключатель и ползунок.



Для смены даты необходимо покрутить вниз или вверх, один или несколько, столбцов селектора. Для селектора времени аналогично. При нажатии на список появится выпадающее меню, где пользователь сможет выбрать один из предложенных вариантов значений.



# Приложение.

* MainActivity.java:

package com.example.lab3;  
  
import android.content.Intent;  
import android.os.Bundle;  
import android.view.View;  
import android.widget.AdapterView;  
import android.widget.ArrayAdapter;  
import android.widget.Button;  
import android.widget.DatePicker;  
import android.widget.SeekBar;  
import android.widget.Spinner;  
import android.widget.Switch;  
import android.widget.TextView;  
import android.widget.TimePicker;  
  
import androidx.appcompat.app.AppCompatActivity;  
  
import java.util.Calendar;  
  
public class MainActivity extends AppCompatActivity {  
 boolean buttonPressed = false;  
 boolean switcherOn = false;  
 int counter = 0;  
  
 private DatePicker mDatePicker;  
 private TextView mInfoDateTextView;  
  
  
 private TimePicker mTimePicker;  
 private TextView mInfoTimeTextView;  
  
 private Spinner spinner;  
 private TextView mInfoSpinnerTextView;  
  
 private SeekBar seekBar;  
 private TextView mInfoSeekbarTextView;  
 private SeekBar.OnSeekBarChangeListener seekBarChangeListener = new SeekBar.OnSeekBarChangeListener() {  
 @Override  
 public void onProgressChanged(SeekBar seekBar, int progress, boolean fromUser) {  
  
 mInfoSeekbarTextView.setText(String.*valueOf*(progress));  
 }  
  
 @Override  
 public void onStartTrackingTouch(SeekBar seekBar) {  
  
 }  
  
 @Override  
 public void onStopTrackingTouch(SeekBar seekBar) {  
  
 }  
 };  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
  
 //---------Date Picker---------//  
 mInfoDateTextView = (TextView) findViewById(R.id.*date\_txt*);  
 mDatePicker = (DatePicker) findViewById(R.id.*date\_picker*);  
  
 Calendar today = Calendar.*getInstance*();  
  
 mDatePicker.init(today.get(Calendar.*YEAR*), today.get(Calendar.*MONTH*),  
 today.get(Calendar.*DAY\_OF\_MONTH*), new DatePicker.OnDateChangedListener() {  
  
 @Override  
 public void onDateChanged(DatePicker view, int year,  
 int monthOfYear, int dayOfMonth) {  
 mInfoDateTextView.setText(  
 year + "."  
 + ((monthOfYear + 1) > 9 ? String.*valueOf*(monthOfYear + 1) : "0" + String.*valueOf*(monthOfYear + 1)) + "."  
 + ((dayOfMonth) > 9 ? String.*valueOf*(dayOfMonth) : "0" + String.*valueOf*(dayOfMonth))  
 );  
 }  
 });  
 int curYear = mDatePicker.getYear();  
 int curMonth = mDatePicker.getMonth();  
 int curDay = mDatePicker.getDayOfMonth();  
 mInfoDateTextView.setText(  
 curYear + "."  
 + ((curMonth + 1) > 9 ? String.*valueOf*(curMonth + 1) : "0" + String.*valueOf*(curMonth + 1)) + "."  
 + ((curDay) > 9 ? String.*valueOf*(curDay) : "0" + String.*valueOf*(curDay))  
 );  
 //---------Time Picker---------//  
 mInfoTimeTextView = findViewById(R.id.*time\_txt*);  
 mTimePicker = findViewById(R.id.*timePicker*);  
  
 mTimePicker.setOnTimeChangedListener(new TimePicker.OnTimeChangedListener() {  
 @Override  
 public void onTimeChanged(TimePicker view, int hourOfDay, int minute) {  
  
 mInfoTimeTextView.setText(  
 (hourOfDay > 9 ? String.*valueOf*(hourOfDay) : "0" + String.*valueOf*(hourOfDay)) + ":"  
 + ((minute) > 9 ? String.*valueOf*(minute) : "0" + String.*valueOf*(minute))  
  
 );  
 }  
 });  
 int curHour = mTimePicker.getHour();  
 int curMinute = mTimePicker.getMinute();  
 mInfoTimeTextView.setText(  
 (curHour > 9 ? String.*valueOf*(curHour) : "0" + String.*valueOf*(curHour)) + ":"  
 + ((curMinute) > 9 ? String.*valueOf*(curMinute) : "0" + String.*valueOf*(curMinute)));  
  
 //---------Spinner---------//  
 spinner = findViewById(R.id.*spinner*);  
 mInfoSpinnerTextView = (TextView) findViewById(R.id.*spinner\_txt*);  
 ArrayAdapter<String> adapter = new ArrayAdapter(this, android.R.layout.*simple\_spinner\_item*, getResources().getStringArray(R.array.*animals*));  
 adapter.setDropDownViewResource(android.R.layout.*simple\_spinner\_dropdown\_item*);  
 spinner.setAdapter(adapter);  
  
 AdapterView.OnItemSelectedListener itemSelectedListener = new AdapterView.OnItemSelectedListener() {  
 @Override  
 public void onItemSelected(AdapterView<?> parent, View view, int position, long id) {  
 String item = (String) parent.getItemAtPosition(position);  
 mInfoSpinnerTextView.setText(item);  
 }  
  
 @Override  
 public void onNothingSelected(AdapterView<?> parent) {  
 }  
 };  
 spinner.setOnItemSelectedListener(itemSelectedListener);  
  
 seekBar = findViewById(R.id.*seekBar*);  
 mInfoSeekbarTextView = findViewById(R.id.*seekbar\_txt*);  
 seekBar.setOnSeekBarChangeListener(seekBarChangeListener);  
  
 }  
  
 public void NextActivity(View v) {  
 startActivity(new Intent(this, Activity2.class));  
 }  
  
 public void onButtonPressed(View v) {  
 TextView txt = findViewById(R.id.*button\_pressed\_txt*);  
 if (!buttonPressed) {  
 v.setBackground(getDrawable(R.drawable.*button\_back\_pressed*));  
 buttonPressed = true;  
 txt.setText("Нажата");  
 } else {  
 v.setBackground(getDrawable(R.drawable.*button\_back\_not\_pressed*));  
 buttonPressed = false;  
 txt.setText("Отпущена");  
 }  
 }  
  
 public void onSwitcherPressed(View v) {  
 if (!switcherOn) {  
 switcherOn = true;  
 ((Switch) v).setText("Включен");  
 } else {  
 switcherOn = false;  
 ((Switch) v).setText("Выключен");  
 }  
 }  
  
 public void onButtonCounterPressed(View v) {  
 ((Button) v).setText((String.*valueOf*(++counter)));  
 }  
  
}

* activity\_main.xml:

<?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"  
 tools:context=".MainActivity">  
  
  
 <Button  
 android:id="@+id/next2"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginEnd="28dp"  
 android:layout\_marginBottom="16dp"  
 android:onClick="NextActivity"  
 android:text="Next"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent" />  
  
  
 <TextView  
 android:id="@+id/textView"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginEnd="16dp"  
 android:layout\_marginBottom="28dp"  
 android:text="1"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toStartOf="@+id/next2"  
 app:layout\_constraintHorizontal\_bias="0.741"  
 app:layout\_constraintStart\_toStartOf="parent" />  
  
  
 <EditText  
 android:id="@+id/editTextTextPersonName"  
 android:layout\_width="216dp"  
 android:layout\_height="48dp"  
 android:layout\_marginTop="12dp"  
 android:ems="10"  
 android:hint="Enter number here"  
 android:inputType="numberDecimal"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.497"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent" />  
  
 <androidx.appcompat.widget.AppCompatButton  
 android:id="@+id/press\_button"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="60dp"  
 android:background="@drawable/button\_back\_not\_pressed"  
 android:onClick="onButtonPressed"  
 android:text="Button"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.694"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent" />  
  
 <TextView  
 android:id="@+id/button\_pressed\_txt"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="72dp"  
 android:text="Отпущена"  
 android:textSize="20sp"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.265"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent" />  
  
 <TextView  
 android:id="@+id/date"  
 android:layout\_width="133dp"  
 android:layout\_height="66dp"  
  
 android:text="Счётчик нажатий:"  
 android:textSize="20sp"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.19"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintVertical\_bias="0.175" />  
  
 <Button  
 android:id="@+id/button"  
 android:layout\_width="108dp"  
 android:layout\_height="63dp"  
  
 android:layout\_marginTop="120dp"  
 android:backgroundTint="#FFFFFF"  
 android:onClick="onButtonCounterPressed"  
 android:text="0"  
 android:textColor="#000000"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.689"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:strokeColor="#800EE3"  
 app:strokeWidth="3dp" />  
  
 <DatePicker  
 android:id="@+id/date\_picker"  
 android:layout\_width="235dp"  
 android:layout\_height="115dp"  
 android:layout\_marginTop="196dp"  
 android:calendarViewShown="false"  
 android:datePickerMode="spinner"  
 android:spinnersShown="true"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.966"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent" />  
  
 <TextView  
 android:id="@+id/date\_txt"  
 android:layout\_width="141dp"  
 android:layout\_height="27dp"  
  
 android:layout\_marginTop="240dp"  
 android:text="1970.01.01"  
 android:textSize="18sp"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.059"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent" />  
  
 <TextView  
 android:id="@+id/time\_txt"  
 android:layout\_width="128dp"  
 android:layout\_height="52dp"  
  
 android:layout\_marginTop="336dp"  
 android:text="15:50"  
 android:textSize="20sp"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.056"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent" />  
  
 <TextView  
 android:id="@+id/spinner\_txt"  
 android:layout\_width="130dp"  
 android:layout\_height="30dp"  
  
 android:layout\_marginTop="456dp"  
 android:text="фыв"  
 android:textSize="20sp"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.067"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent" />  
  
 <TextView  
 android:id="@+id/seekbar\_txt"  
 android:layout\_width="130dp"  
 android:layout\_height="30dp"  
  
 android:layout\_marginTop="8dp"  
 android:layout\_marginEnd="8dp"  
 android:text="0"  
 android:textSize="20sp"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toStartOf="@+id/seekBar"  
 app:layout\_constraintHorizontal\_bias="0.365"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/switch1"  
 app:layout\_constraintVertical\_bias="0.025" />  
  
 <TimePicker  
 android:id="@+id/timePicker"  
 android:layout\_width="235dp"  
 android:layout\_height="115dp"  
 android:layout\_marginTop="320dp"  
 android:timePickerMode="spinner"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.965"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent" />  
  
 <Spinner  
 android:id="@+id/spinner"  
 android:layout\_width="221dp"  
 android:layout\_height="29dp"  
 android:layout\_marginTop="456dp"  
 android:entries="@array/animals"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.894"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent" />  
  
 <Switch  
 android:id="@+id/switch1"  
 android:layout\_width="303dp"  
 android:layout\_height="28dp"  
 android:layout\_marginTop="500dp"  
 android:layout\_marginEnd="20dp"  
 android:background="@drawable/button\_back\_not\_pressed"  
 android:checked="false"  
 android:onClick="onSwitcherPressed"  
 android:text="Выключен"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 tools:checked="false" />  
  
 <SeekBar  
 android:id="@+id/seekBar"  
 android:layout\_width="164dp"  
 android:layout\_height="31dp"  
 android:layout\_marginTop="8dp"  
 android:layout\_marginBottom="116dp"  
 android:max="777"  
 android:progress="1"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.813"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/switch1"  
 app:layout\_constraintVertical\_bias="0.088" />  
  
</androidx.constraintlayout.widget.ConstraintLayout>

* Button\_back\_not\_pressed.xml:

<shape xmlns:android="http://schemas.android.com/apk/res/android"  
 android:shape="rectangle">  
  
 <solid android:color="@color/orange\_700" />  
  
 <corners  
 android:bottomLeftRadius="10dip"  
 android:topRightRadius="10dp" />  
 />  
  
 <size  
 android:width="90dp"  
 android:height="30dp" />  
  
</shape>

* Button\_back\_pressed.xml:

<shape xmlns:android="http://schemas.android.com/apk/res/android"  
 android:shape="rectangle">  
  
 <solid android:color="@color/orange\_200" />  
  
 <corners  
 android:bottomLeftRadius="10dip"  
 android:topRightRadius="10dp" />  
  
 />  
  
 <size  
 android:width="90dp"  
 android:height="30dp" />  
   
</shape>