Министерство образования и науки Российской Федерации

Федеральное государственное автономное образовательное

учреждение высшего образования

«Национальный исследовательский

Томский политехнический Университет»



Инженерная школа ядерных технологий

Направление 01.04.02 «Прикладная математика и информатика»

Специализация «Математическое моделирование и компьютерные вычисления»

**ОТЧЕТ**

по лабораторной работе №2

«Вёрстка UI. Сохранение состояния»

Вариант 1

по дисциплине:

**Программирование мобильных приложений**

|  |  |  |  |
| --- | --- | --- | --- |
| **Исполнитель:** |  | Е. В. Петрович | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| студент группы 0ВМ92 |  | Дата сдачи: | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
|  |  |  |  |
|  |  |  |  |
| **Руководитель:** |  | А. В. Погребной | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| ассистент |  | Дата проверки: | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
|  |  |  |  |
|  |  |  |  |

Томск - 2019

Оглавление

[**Цель работы** 2](#_Toc24050003)

[**Задание** 2](#_Toc24050004)

[Общее задание 2](#_Toc24050005)

[Задание варианта 1 2](#_Toc24050006)

[**Основная часть** 2](#_Toc24050007)

[**Заключение** 4](#_Toc24050008)

[**Приложение 1** 4](#_Toc24050009)

[Исходный код программы 4](#_Toc24050010)

[Модуль Lab2Activity 4](#_Toc24050011)

[Модуль контейнера Lab2Activity 9](#_Toc24050012)

[Модуль макета activity\_lab2 15](#_Toc24050013)

[**Приложение 2** 16](#_Toc24050014)

[Ссылка на проект 16](#_Toc24050015)

**Цель работы**

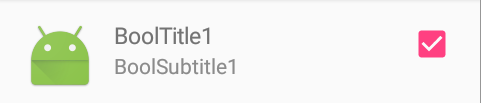
Реализовать описанный интерфейс пользователя и функционал сохранения состояния.

**Задание**

Общее задание

Ознакомиться с документацией модуля lab2. Реализовать описанный Custom View. Добавить на Activity результат в разных состояниях, либо контролы для изменения этого состояния. Восстановление состояния должно быть корректно обработано.

Задание варианта 1



При тапе по любой точке View, CheckBox должен переключаться. У View можно задать картинку, заголовок, подзаголовок и состояние (выбрано или нет). Если картинки нет, текст должен сдвигаться влево. Если подзаголовка нет, заголовок должен быть отцентрирован по вертикали. CheckBox находится в правом верхнем углу с задаваемым отступом справа и сверху. Для вывода картинки используется ImageView. Необходимо добавить несколько картинок разного размера в папку res/drawable для демонстрации результата.

**Основная часть**

Для изменения картинки и надписей были добавлены кнопки, запускающие соответствующие диалоговые окна. Приложение было запущено на реальном устройстве Samsung Galaxy Note 9. Скриншоты работы представлены на рисунках. Исходный код модулей представлен в Приложении 1. Ссылка на репозиторий проекта на Github представлена в Приложении 2.

|  |  |  |
| --- | --- | --- |
| D:\SharedFolder\A1\Screenshot_20191107-172741_Android Labs.jpg | D:\SharedFolder\A1\Screenshot_20191107-172807_Android Labs.jpg | D:\SharedFolder\A1\Screenshot_20191107-172843_Android Labs.jpg |
| Рис. 1. Стартовый вид экрана | Рис. 2. Экран с добавленной картинкой | Рис. 3. Экран без заголовка |

|  |  |  |  |
| --- | --- | --- | --- |
| D:\SharedFolder\A1\Screenshot_20191107-172901_Android Labs.jpg | D:\SharedFolder\A1\Screenshot_20191107-172935_Android Labs.jpg | | D:\SharedFolder\A1\Screenshot_20191107-173012_Android Labs.jpg |
| Рис. 4. Экран без надписей | Рис. 5. Экран без подзаголовка | | Рис. 6. Экран с картинкой меньшего размера |
| D:\SharedFolder\A1\Screenshot_20191107-172909_Android Labs.jpg | | D:\SharedFolder\A1\Screenshot_20191107-173041_Android Labs.jpg | |
| Рис. 7. Повернутый экран | | Рис. 8. Повернутый экран с картинкой меньшего размера | |

**Заключение**

В работе реализован интерфейс пользователя в соответствии с требованиями, изложенными в задании. Интерфейс также восстанавливается при повороте устройства и при возвращении к рабочему приложению после запуска и работы с другими приложениями. Проект протестирован на реальном устройстве и загружен на сайт «Github».

**Приложение 1**

Исходный код программы

Модуль Lab2Activity

package ru.tpu.courses.lab2;

import android.content.Context;

import android.content.DialogInterface;

import android.content.Intent;

import android.graphics.Bitmap;

import android.graphics.BitmapFactory;

import android.os.Bundle;

import android.text.InputType;

import android.view.View;

import android.view.View.OnClickListener;

import android.widget.Button;

import android.widget.EditText;

import androidx.annotation.NonNull;

import androidx.annotation.Nullable;

import androidx.appcompat.app.AlertDialog;

import androidx.appcompat.app.AppCompatActivity;

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

public class Lab2Activity extends AppCompatActivity implements OnClickListener {

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

public static Intent newIntent(@NonNull Context context) {

return new Intent(context, Lab2Activity.class);

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

@Override

protected void onCreate(@Nullable Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

// установка содержимого активности из ресурса макета

setContentView(R.layout.activity\_lab2);

// установка заголовка экрана

setTitle(getString(R.string.lab2\_app\_name));

// Определение кнопок и установка в них слушателя нажатий Lab2Activity

Button imgBtn = findViewById(R.id.lab2\_imageBtn);

imgBtn.setOnClickListener(this);

Button titleBtn = findViewById(R.id.lab2\_titleBtn);

titleBtn.setOnClickListener(this);

Button subtitleBtn = findViewById(R.id.lab2\_subtitleBtn);

subtitleBtn.setOnClickListener(this);

// восстанавление состояния экрана, если оно до этого было сохранено

if (savedInstanceState != null) {

Lab2ViewsContainer lab2ViewsContainer = getViewContainer();

// Передаем в контейнер Bundle c сохраненным состоянием

// и контейнер восстанавливает свое состояние сам

lab2ViewsContainer.setInstanceState(savedInstanceState);

}

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

@Override

protected void onSaveInstanceState(@NonNull Bundle instanceState) {

super.onSaveInstanceState(instanceState);

// сохранение состояние экрана

Lab2ViewsContainer lab2ViewsContainer = getViewContainer();

// Передаем в контейнер Lab2ViewsContainer Bundle instanceState

// и свое сотояние контейнер сохраняет сам

lab2ViewsContainer.saveInstanceState(instanceState);

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

@Override

public void onClick(View v) {

// обработка нажатий кнопок

int id = v.getId();

if (id == R.id.lab2\_imageBtn) {

// Запуск диалога для установки картинки

getPictureAndSetToView();

} else if (id == R.id.lab2\_titleBtn) {

// Запуск диалога для записи заголовка

getUserStringAndWriteToView(R.string.lab2\_title);

} else if (id == R.id.lab2\_subtitleBtn) {

// Запуск диалога для записи подзаголовка

getUserStringAndWriteToView(R.string.lab2\_subtitle);

}

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

protected Lab2ViewsContainer getViewContainer() {

Lab2ViewsContainer viewContainer = findViewById(R.id.container);

return viewContainer;

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

protected void getUserStringAndWriteToView(int strId) {

// Создание и настройка диалога ввода текста

AlertDialog.Builder builder = new AlertDialog.Builder(this);

builder.setTitle(getResources().getString(strId));

// Создание поля для ввода текста

EditText input = new EditText(this);

// настройка типа вводимого текста

input.setInputType(InputType.TYPE\_CLASS\_TEXT | InputType.TYPE\_TEXT\_VARIATION\_NORMAL);

// установка в диалог

builder.setView(input);

// Установка кнопки ОК и обработчик ее нажатия

builder.setPositiveButton("OK", new DialogInterface.OnClickListener() {

@Override

public void onClick(DialogInterface dialog, int which) {

// Получаем введенную строку и записываем в контейнер

// strId - для определения куда записывать строку - в заголовок или подзаголовок

Lab2ViewsContainer lab2ViewsContainer = getViewContainer();

if (strId == R.string.lab2\_title) {

lab2ViewsContainer.setTitle(input.getText().toString());

} else if (strId == R.string.lab2\_subtitle) {

lab2ViewsContainer.setSubtitle(input.getText().toString());

}

}

});

// Установка кнопки Cancel и обработчик ее нажатия

builder.setNegativeButton("Cancel", new DialogInterface.OnClickListener() {

@Override

public void onClick(DialogInterface dialog, int which) {

dialog.cancel();

}

});

// старт диалога

builder.show();

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

protected void getPictureAndSetToView() {

// R.drawable.class.getField("name\_of\_the\_resource").getInt(getResources())

// Создание и настройка диалога ввода текста

AlertDialog.Builder builder = new AlertDialog.Builder(this);

builder.setTitle(getResources().getString(R.string.lab2\_image));

// Создание списка картинок

final String[] pictureName = {"None", "Andriod", "GPS", "Teacher"};

// Установка списка картинок и обработчик нажатия

builder.setItems(pictureName, new DialogInterface.OnClickListener() {

@Override

public void onClick(DialogInterface dialog, int item) {

// Получаем введенную строку и загружаем картинку в контейнер

Bitmap bitmap;

bitmap = null;

switch (item) {

case 0:

// пустой битмап

break;

case 1: // Andriod

bitmap = BitmapFactory.decodeResource(getResources(), R.drawable.android\_games);

break;

case 2: // GPS

bitmap = BitmapFactory.decodeResource(getResources(), R.drawable.gps);

break;

case 3: // Teacher

bitmap = BitmapFactory.decodeResource(getResources(), R.drawable.teacher);

break;

}

Lab2ViewsContainer lab2ViewsContainer = getViewContainer();

lab2ViewsContainer.setImage(bitmap);

}

});

// Установка кнопки Cancel и обработчик ее нажатия

builder.setNegativeButton("Cancel", new DialogInterface.OnClickListener() {

@Override

public void onClick(DialogInterface dialog, int which) {

dialog.cancel();

}

});

// старт диалога

builder.show();

}

}

Модуль контейнера Lab2Activity

package ru.tpu.courses.lab2;

import android.content.Context;

import android.content.res.TypedArray;

import android.graphics.Bitmap;

import android.graphics.drawable.BitmapDrawable;

import android.os.Build;

import android.os.Bundle;

import android.util.AttributeSet;

import android.util.TypedValue;

import android.view.Gravity;

import android.view.View;

import android.view.View.OnClickListener;

import android.widget.CheckBox;

import android.widget.ImageView;

import android.widget.TextView;

import androidx.annotation.NonNull;

import androidx.annotation.RequiresApi;

import androidx.constraintlayout.widget.ConstraintLayout;

import androidx.constraintlayout.widget.ConstraintSet;

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

public class Lab2ViewsContainer extends ConstraintLayout implements OnClickListener {

ImageView imageView;

TextView titleView;

TextView subtitleView;

CheckBox checkBox;

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// Этот конструктор используется при создании View в коде.

@RequiresApi(api = Build.VERSION\_CODES.JELLY\_BEAN\_MR1)

public Lab2ViewsContainer(Context context) {

this(context, null);

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//Этот конструктор вызывается при создании View из XML.

@RequiresApi(api = Build.VERSION\_CODES.JELLY\_BEAN\_MR1)

public Lab2ViewsContainer(Context context, AttributeSet attrs) {

this(context, attrs, 0);

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// Конструктор, вызывается при инфлейте View, когда у View указан дополнительный стиль.

@RequiresApi(api = Build.VERSION\_CODES.JELLY\_BEAN\_MR1)

public Lab2ViewsContainer(Context context, AttributeSet attrs, int defStyleAttr) {

super(context, attrs, defStyleAttr);

// Свои атрибуты описываются в файле res/values/attrs.xml

// Эта строчка объединяет возможные применённые к View стили

TypedArray a = context.obtainStyledAttributes(attrs, R.styleable.Lab2ViewsContainer, defStyleAttr, 0);

// Полученный TypedArray необходимо обязательно очистить.

a.recycle();

// создание imageView для отображения картинки

imageView = new ImageView(getContext());

// генерируем и присваиваем id

imageView.setId(View.generateViewId());

// установка пустого bitmap

Bitmap bitmap = null;

imageView.setImageBitmap(bitmap);

imageView.setVisibility(View.GONE);

addView(imageView);

// создание textView для отображения заголовка

titleView = new TextView(getContext());

titleView.setId(View.generateViewId());

titleView.setGravity(Gravity.LEFT | Gravity.CENTER\_VERTICAL);

titleView.setLayoutParams(new LayoutParams(LayoutParams.WRAP\_CONTENT, LayoutParams.WRAP\_CONTENT));

titleView.setTextSize(TypedValue.COMPLEX\_UNIT\_PX, getResources().getDimensionPixelSize(R.dimen.lab2\_title\_size));

titleView.setText(getResources().getString(R.string.lab2\_title));

addView(titleView);

subtitleView = new TextView(getContext());

subtitleView.setId(View.generateViewId());

subtitleView.setGravity(Gravity.LEFT | Gravity.CENTER\_VERTICAL);

subtitleView.setLayoutParams(new LayoutParams(LayoutParams.WRAP\_CONTENT, LayoutParams.WRAP\_CONTENT));

subtitleView.setTextSize(TypedValue.COMPLEX\_UNIT\_PX, getResources().getDimensionPixelSize(R.dimen.lab2\_subtitle\_size));

subtitleView.setText(getResources().getString(R.string.lab2\_subtitle));

addView(subtitleView);

checkBox = new CheckBox(getContext());

checkBox.setId(View.generateViewId());

addView(checkBox);

// установка всем view "слушателя кликов" - класса Lab2ViewsContainer

// для переключения checkBox при клике по любому месту всего объекта класса Lab2ViewsContainer

imageView.setOnClickListener(this);

titleView.setOnClickListener(this);

subtitleView.setOnClickListener(this);

setOnClickListener(this);

// размещение всех view на макете

adjustViews();

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

@Override

public void onClick(View v) {

// переключение checkBox на противоположное

checkBox.setChecked(!checkBox.isChecked());

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

public void setInstanceState(@NonNull Bundle savedInstanceState) {

// восстановление состояний всех view

Bitmap bitmap = savedInstanceState.getParcelable(getResources().getString(R.string.lab2\_image));

setImage(bitmap);

setTitle(savedInstanceState.getString(getResources().getString(R.string.lab2\_title)));

setSubtitle(savedInstanceState.getString(getResources().getString(R.string.lab2\_subtitle)));

checkBox.setChecked(savedInstanceState.getBoolean(getResources().getString(R.string.lab2\_checkbox)));

adjustViews();

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

public void saveInstanceState(@NonNull Bundle outState) {

// сохранение состояний всех view

Bitmap bitmap = ((BitmapDrawable) imageView.getDrawable()).getBitmap();

outState.putParcelable(getResources().getString(R.string.lab2\_image), bitmap);

outState.putCharSequence(getResources().getString(R.string.lab2\_title), titleView.getText());

outState.putCharSequence(getResources().getString(R.string.lab2\_subtitle), subtitleView.getText());

outState.putBoolean(getResources().getString(R.string.lab2\_checkbox), checkBox.isChecked());

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

public void adjustViews() {

// горизонтальное расстояние между view

int padding = getResources().getDimensionPixelOffset(R.dimen.lab2\_view\_padding);

// набор настроек для макета СonstraintLayout

ConstraintSet set = new ConstraintSet();

set.clone(this);

// imageView

// привязка в левый верхний угол макета

set.connect(imageView.getId(), ConstraintSet.LEFT,

R.id.container, ConstraintSet.LEFT);

set.connect(imageView.getId(), ConstraintSet.TOP,

R.id.container, ConstraintSet.TOP);

// checkBox

// привязка в правый верхний угол макета

// с отступами по вертикали и горизонтали

// значения отступов - в ресурсах dimens

set.connect(checkBox.getId(), ConstraintSet.RIGHT,

R.id.container, ConstraintSet.RIGHT,

getResources().getDimensionPixelOffset(R.dimen.lab2\_checkBox\_h\_margin));

set.connect(checkBox.getId(), ConstraintSet.TOP,

R.id.container, ConstraintSet.TOP,

getResources().getDimensionPixelOffset(R.dimen.lab2\_checkBox\_v\_margin));

// titleView

// привязка левого края к правому краю картинки

// правого края к checkBox

// в ширину titleView занимает всё свободное место от картинки до checkBox

set.connect(titleView.getId(), ConstraintSet.LEFT,

imageView.getId(), ConstraintSet.RIGHT, padding);

set.connect(titleView.getId(), ConstraintSet.TOP,

R.id.container, ConstraintSet.TOP);

set.connect(titleView.getId(), ConstraintSet.RIGHT,

checkBox.getId(), ConstraintSet.LEFT, padding);

set.constrainWidth(titleView.getId(), ConstraintSet.MATCH\_CONSTRAINT);

// subtitle view

// привязка левого края к правому краю картинки

// правого края к checkBox

// в ширину titleView занимает всё свободное место от картинки до checkBox

set.connect(subtitleView.getId(), ConstraintSet.LEFT,

imageView.getId(), ConstraintSet.RIGHT, padding);

set.connect(subtitleView.getId(), ConstraintSet.RIGHT,

checkBox.getId(), ConstraintSet.LEFT, padding);

set.constrainWidth(subtitleView.getId(), ConstraintSet.MATCH\_CONSTRAINT);

// применение настроек к макету

set.applyTo(this);

adjustSubtitleView();

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

protected void adjustSubtitleView() {

// набор настроек для макета СonstraintLayout

ConstraintSet set = new ConstraintSet();

set.clone(this);

if ((titleView.getVisibility() == View.GONE) &

(subtitleView.getVisibility() != View.GONE)) {

// привязка центра textView подзаголовка к центру контейнера по вертикали

set.clear(subtitleView.getId(), ConstraintSet.TOP);

set.clear(subtitleView.getId(), ConstraintSet.BOTTOM);

set.connect(subtitleView.getId(), ConstraintSet.TOP,

R.id.container, ConstraintSet.TOP);

set.connect(subtitleView.getId(), ConstraintSet.BOTTOM,

R.id.container, ConstraintSet.BOTTOM);

} else {

// привязка верхнего края textView подзаголовка к нижнему краю textView заголовка

set.clear(subtitleView.getId(), ConstraintSet.TOP);

set.clear(subtitleView.getId(), ConstraintSet.BOTTOM);

set.connect(subtitleView.getId(), ConstraintSet.TOP,

titleView.getId(), ConstraintSet.BOTTOM);

}

// применение настроек к макету

set.applyTo(this);

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// установка картинки

public void setImage(Bitmap bitmap) {

imageView.setImageBitmap(bitmap);

if (bitmap == null) {

imageView.setVisibility(View.GONE);

} else {

imageView.setVisibility(View.VISIBLE);

}

adjustSubtitleView();

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// установка заголовка

public void setTitle(String title) {

if (titleView.getText().toString() != title) {

titleView.setText(title);

if (title.isEmpty()) {

// при пустой строке titleView удаляется с макета и не занимает места

titleView.setVisibility(View.GONE);

} else {

titleView.setVisibility(View.VISIBLE);

}

adjustSubtitleView();

}

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// установка подзаголовка

public void setSubtitle(String subtitle) {

if (subtitleView.getText().toString() != subtitle) {

subtitleView.setText(subtitle);

if (subtitle.isEmpty()) {

// при пустой строке subtitleView удаляется с макета и не занимает места

subtitleView.setVisibility(View.GONE);

} else {

subtitleView.setVisibility(View.VISIBLE);

}

adjustSubtitleView();

}

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Модуль макета activity\_lab2

<?xml version="1.0" encoding="utf-8"?>

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"

android:id="@+id/activity\_lab2\_mainLayout"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:orientation="vertical"

android:padding="16dp">

<!-- Lab2ViewsContainer - класс, содержащий все view-->

<ru.tpu.courses.lab2.Lab2ViewsContainer

android:id="@+id/container"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:orientation="vertical" />

<LinearLayout

android:id="@+id/activity\_lab2\_btnLayout"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_gravity="bottom"

android:orientation="horizontal"

android:padding="16dp">

<Button

android:id="@+id/lab2\_imageBtn"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_weight="1"

android:text="@string/lab2\_image" />

<Button

android:id="@+id/lab2\_titleBtn"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_weight="1"

android:text="@string/lab2\_title" />

<Button

android:id="@+id/lab2\_subtitleBtn"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_weight="1"

android:text="@string/lab2\_subtitle" />

</LinearLayout>

</LinearLayout>

**Приложение 2**

Ссылка на проект

<https://github.com/petrovicheugene/MobAppLabs.git>