la

KAUNO TECHNOLOGIJOS UNIVERSITETAS

Informatikos fakultetas

T120B169 App Development for Smart Mobile Systems

**Report of labaratory work no.2**

**Dėstytojas: Rytis Maskeliūnas**

**Studentas: Kasparas Giniūnas IFF-6/4**

KAUNAS, 2018

Table of contents

[Tasks 3](#_Toc530472624)

[Implementation 4](#_Toc530472625)

[ANNEX 7](#_Toc530472626)

[ *CircleView.java* file 7](#_Toc530472627)

[ *IndicatingView.java* file 7](#_Toc530472628)

[ *MainActivity.java* file 8](#_Toc530472629)

[ *ModelPost.java* file 9](#_Toc530472630)

[ *RequestOperator.java* file 10](#_Toc530472631)

[ *button.xml* file 11](#_Toc530472632)

[ *button\_default.xml* file 12](#_Toc530472633)

[ *button\_pressed.xml* file 12](#_Toc530472634)

[ circle.xml file 12](#_Toc530472635)

# Tasks

1. During the execution of the request on indicator should show the triangle
2. Work with URL-address of request of ….

The resulting Json should be JSONArray.

This array should be made of a list of publications

Create new indicator which show the number of publications. A quantity of publications should be indicated inside of the circle. Inner color must be black, the number inside must be white.

1. Change the design of “Send request” button so that the button (when not pressed), has a gradient color.
2. \*Add a progress animation function while the query is executing
3. \*Draw a progress indicator of gradually appearing different color squares (getting darker toward the end)

# Implementation

1. During the execution of the request on indicator should show the triangle

Method onDraw() in IndicatingView classs

**protected void** onDraw(Canvas canvas){  
 **super**.onDraw(canvas);  
 **int** width = getWidth();  
 **int** height = getHeight();  
 Paint paint;  
 **switch** (**state**){  
 **case *SUCCESS***:  
 paint = **new** Paint();  
 paint.setColor(Color.***GREEN***);  
 paint.setStrokeWidth(20f);  
 canvas.drawLine(0,0, width/2, height, paint);  
 canvas.drawLine(width/2, height, width, height/2, paint);  
 **break**;  
  
 **case *FAILED***:  
 paint = **new** Paint();  
 paint.setColor(Color.***RED***);  
 paint.setStrokeWidth(20f);  
 canvas.drawLine(0,0, width, height, paint);  
 canvas.drawLine(0, height, width, 0, paint);  
 **break**;  
  
 **case *INPROGRESS***:  
 paint = **new** Paint();  
 paint.setColor(Color.***WHITE***);  
 paint.setStrokeWidth(20f);  
 paint.setStyle(Paint.Style.***STROKE***);  
 paint.setPathEffect(**new** DashPathEffect(**new float**[]{40, 40,}, 0));  
 **mPath** = **new** Path();  
 **mPath**.moveTo(width/2, 5);  
 **mPath**.quadTo(5, height-5, 5, height-5);  
 **mPath**.quadTo(width -5, height-5, width-5, height-5);  
 **mPath**.quadTo(width/2, 5, width/2, 5);  
 canvas.drawPath(**mPath**, paint);  
 **break**;  
  
 **default**:  
 **break**;  
 }

1. Work with URL-address of request of ….

The resulting Json should be JSONArray.

This array should be made of a list of publications

Create new indicator which show the number of publications. A quantity of publications should be indicated inside of the circle. Inner color must be black, the number inside must be white.

Parsing publications from JSONArray object

JSONArray publications = **new** JSONArray(response);  
  
List<ModelPost> responses = **new** ArrayList<>();  
  
**for**(**int** i = 0; i< publications.length(); i++){  
 JSONObject object = publications.getJSONObject(i);  
 ModelPost post = **new** ModelPost();  
  
 post.setId(object.optInt(**"id"**, 0));  
 post.setUserId(object.optInt(**"userId"**, 0));  
  
 post.setTitle(object.getString(**"title"**));  
 post.setBodyText(object.getString(**"body"**));  
  
 responses.add(post);  
}

CircleView class for drawing circle indicator

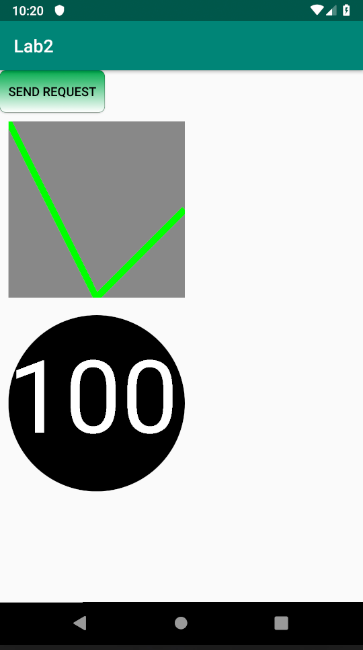
**package** edu.ktu.lab2;  
  
**import** android.content.Context;  
**import** android.graphics.Canvas;  
**import** android.graphics.Color;  
**import** android.graphics.Paint;  
**import** android.util.AttributeSet;  
**import** android.view.View;  
  
**public class** CircleView **extends** View {  
  
 String **text** = **""**;  
  
 **public** CircleView(Context context) {**super**(context);}  
  
 **public** CircleView(Context context, AttributeSet attrs) {**super**(context, attrs);}  
  
 **public** CircleView(Context context, AttributeSet attrs, **int** defStyleAttr) {  
 **super**(context, attrs, defStyleAttr);  
 }  
  
 **public void** setText(String text){  
 **this**.**text** = text;  
 }  
  
 **protected void** onDraw(Canvas canvas){  
 **super**.onDraw(canvas);  
 Paint paint = **new** Paint();  
  
 **float** width = getWidth();  
  
 **float** textWidth = paint.measureText(**text**);  
 System.***out***.println(textWidth);  
  
 paint.setColor(Color.***WHITE***);  
 paint.setTextSize(300);  
 paint.setTextAlign(Paint.Align.***CENTER***);  
 canvas.drawText(**text**, (width-textWidth)/2f, 350, paint);  
 }  
}

1. Change the design of “Send request” button so that the button (when not pressed), has a gradient color.

ButtonDefault.xml file for button design when its not pressed

*<?***xml version="1.0" encoding="utf-8"***?>*<**shape  
 xmlns:android="http://schemas.android.com/apk/res/android"**>  
  
 <**stroke  
 android:width="1px"  
 android:color="#006030"**/>  
  
 <**corners  
 android:radius="7dp"**/>  
  
 <**padding  
 android:left="1dp"  
 android:right="1dp"  
 android:top="1dp"  
 android:bottom="1dp"**/>  
  
 <**gradient  
 android:startColor="#00ab4c"  
 android:endColor="#FFFFFF"  
 android:angle="-180"** />  
</**shape**>

**Application result:**



# ANNEX

## *CircleView.java* file

**package** edu.ktu.lab2;  
  
**import** android.content.Context;  
**import** android.graphics.Canvas;  
**import** android.graphics.Color;  
**import** android.graphics.Paint;  
**import** android.util.AttributeSet;  
**import** android.view.View;  
  
**public class** CircleView **extends** View {  
  
 String **text** = **""**;  
  
 **public** CircleView(Context context) {**super**(context);}  
  
 **public** CircleView(Context context, AttributeSet attrs) {**super**(context, attrs);}  
  
 **public** CircleView(Context context, AttributeSet attrs, **int** defStyleAttr) {  
 **super**(context, attrs, defStyleAttr);  
 }  
  
 **public void** setText(String text){  
 **this**.**text** = text;  
 }  
  
 **protected void** onDraw(Canvas canvas){  
 **super**.onDraw(canvas);  
 Paint paint = **new** Paint();  
  
 **float** width = getWidth();  
  
 **float** textWidth = paint.measureText(**text**);  
 System.***out***.println(textWidth);  
  
 paint.setColor(Color.***WHITE***);  
 paint.setTextSize(300);  
 paint.setTextAlign(Paint.Align.***CENTER***);  
 canvas.drawText(**text**, (width-textWidth)/2f, 350, paint);  
 }  
}

## *IndicatingView.java* file

**package** edu.ktu.lab2;  
  
**import** android.content.Context;  
**import** android.graphics.Canvas;  
**import** android.graphics.Color;  
**import** android.graphics.DashPathEffect;  
**import** android.graphics.Paint;  
**import** android.graphics.Path;  
**import** android.util.AttributeSet;  
**import** android.view.View;  
  
**public class** IndicatingView **extends** View {  
 **public static final int *NOTEXECUTED*** = 0;  
 **public static final int *SUCCESS*** = 1;  
 **public static final int *FAILED*** = 2;  
 **public static final int *INPROGRESS*** = 3;  
  
 **int state** = ***NOTEXECUTED***;  
  
 **private** Path **mPath**;  
  
 **public** IndicatingView(Context context) {**super**(context);}  
  
 **public** IndicatingView(Context context, AttributeSet attrs) {**super**(context, attrs);}  
  
 **public** IndicatingView(Context context, AttributeSet attrs, **int** defStyleAttr) {  
 **super**(context, attrs, defStyleAttr);  
 }  
  
 **public int** getState(){  
 **return state**;  
 }  
  
 **public void** setState(**int** state) {  
 **this**.**state** = state;  
 }  
  
 **protected void** onDraw(Canvas canvas){  
 **super**.onDraw(canvas);  
 **int** width = getWidth();  
 **int** height = getHeight();  
 Paint paint;  
 **switch** (**state**){  
 **case *SUCCESS***:  
 paint = **new** Paint();  
 paint.setColor(Color.***GREEN***);  
 paint.setStrokeWidth(20f);  
 canvas.drawLine(0,0, width/2, height, paint);  
 canvas.drawLine(width/2, height, width, height/2, paint);  
 **break**;  
  
 **case *FAILED***:  
 paint = **new** Paint();  
 paint.setColor(Color.***RED***);  
 paint.setStrokeWidth(20f);  
 canvas.drawLine(0,0, width, height, paint);  
 canvas.drawLine(0, height, width, 0, paint);  
 **break**;  
  
 **case *INPROGRESS***:  
 paint = **new** Paint();  
 paint.setColor(Color.***WHITE***);  
 paint.setStrokeWidth(20f);  
 paint.setStyle(Paint.Style.***STROKE***);  
 paint.setPathEffect(**new** DashPathEffect(**new float**[]{40, 40,}, 0));  
 **mPath** = **new** Path();  
 **mPath**.moveTo(width/2, 5);  
 **mPath**.quadTo(5, height-5, 5, height-5);  
 **mPath**.quadTo(width -5, height-5, width-5, height-5);  
 **mPath**.quadTo(width/2, 5, width/2, 5);  
 canvas.drawPath(**mPath**, paint);  
 **break**;  
  
 **default**:  
 **break**;  
 }  
 }  
  
}

## *MainActivity.java* file

* **package** edu.ktu.lab2;  
    
  **import** android.graphics.Canvas;  
  **import** android.graphics.Color;  
  **import** android.graphics.Paint;  
  **import** android.graphics.drawable.Drawable;  
  **import** android.os.Bundle;  
  **import** android.support.v7.app.AppCompatActivity;  
  **import** android.view.View;  
  **import** android.widget.Button;  
  **import** android.widget.TextView;  
    
  **import** java.util.List;  
    
  **public class** MainActivity **extends** AppCompatActivity **implements** RequestOperator.RequestOperatorListener {  
    
   Button **sendRequestButton**;  
   TextView **title**;  
   TextView **bodyText**;  
   **private** List<ModelPost> **publication**;  
   **private** IndicatingView **indicator**;  
   **private** CircleView **indicator2**;  
   **private int size** = -1;  
    
   @Override  
   **protected void** onCreate(Bundle savedInstanceState){  
   **super**.onCreate(savedInstanceState);  
   setContentView(R.layout.***mainactivitydesign***);  
    
   **sendRequestButton** = (Button) findViewById(R.id.***send\_request***);  
   **sendRequestButton**.setOnClickListener(**requestButtonClicked**);  
    
    
   **title** = (TextView)findViewById(R.id.***title***);  
   **bodyText** = (TextView) findViewById(R.id.***body\_text***);  
    
   **indicator** = (IndicatingView)findViewById(R.id.***generated\_graphic***);  
   **indicator2** = (CircleView)findViewById(R.id.***circle\_view***);  
   }  
    
   View.OnClickListener **requestButtonClicked** = **new** View.OnClickListener() {  
   @Override  
   **public void** onClick(View v) {  
   setIndicatorStatus(IndicatingView.***INPROGRESS***);  
   **indicator2**.setText(**""**);  
   **indicator2**.invalidate();  
   sendRequest();  
    
   }  
   };  
    
   **public void** setIndicatorStatus(**final int** status){  
   runOnUiThread(**new** Runnable() {  
   @Override  
   **public void** run() {  
   **indicator**.setState(status);  
   **indicator**.invalidate();  
   }  
   });  
   }  
    
   **private void** sendRequest(){  
   RequestOperator ro = **new** RequestOperator();  
   ro.setListener(**this**);  
   ro.start();  
   }  
    
   **public void** updatePublication(){  
   runOnUiThread(**new** Runnable() {  
   @Override  
   **public void** run() {  
   **if**(**publication** != **null**){  
   **indicator2**.setText(Integer.*toString*(**publication**.size()));  
   **indicator2**.invalidate();  
   } **else** {  
   **title**.setText(**""**);  
   **bodyText**.setText(**""**);  
   }  
   }  
   });  
   }  
    
   @Override  
   **public void** success(List<ModelPost> publication){  
   **this**.**publication** = publication;  
   **this**.**size** = publication.size();  
   updatePublication();  
   setIndicatorStatus(IndicatingView.***SUCCESS***);  
   }  
    
   @Override  
   **public void** failed(**int** responseCode){  
   **this**.**publication** = **null**;  
   updatePublication();  
   setIndicatorStatus(IndicatingView.***FAILED***);  
   }  
    
  }

## *ModelPost.java* file

* **package** edu.ktu.lab2;  
    
  **import** android.graphics.ColorSpace;  
    
  **public class** ModelPost {  
    
   **int id**;  
   **int userId**;  
   String **title**;  
   String **bodyText**;  
    
   **public** ModelPost(){  
    
   }  
    
   **public** ModelPost(**int** id, **int** userId, String title, String bodyText){  
   **this**.**id** = id;  
   **this**.**userId** = userId;  
   **this**.**title** = title;  
   **this**.**bodyText** = bodyText;  
   }  
    
   **public int** getId() {  
   **return id**;  
   }  
    
   **public void** setId(**int** id) {  
   **this**.**id** = id;  
   }  
    
   **public int** getUserId() {  
   **return userId**;  
   }  
    
   **public void** setUserId(**int** userId) {  
   **this**.**userId** = userId;  
   }  
    
   **public** String getTitle() {  
   **return title**;  
   }  
    
   **public void** setTitle(String title) {  
   **this**.**title** = title;  
   }  
    
   **public** String getBodyText() {  
   **return bodyText**;  
   }  
    
   **public void** setBodyText(String bodyText) {  
   **this**.**bodyText** = bodyText;  
   }  
  }

## *RequestOperator.java* file

* **package** edu.ktu.lab2;  
    
  **import** org.json.JSONArray;  
  **import** org.json.JSONException;  
  **import** org.json.JSONObject;  
    
  **import** java.io.BufferedReader;  
  **import** java.io.IOException;  
  **import** java.io.InputStreamReader;  
  **import** java.net.HttpURLConnection;  
  **import** java.net.MalformedURLException;  
  **import** java.net.URL;  
  **import** java.util.ArrayList;  
  **import** java.util.List;  
    
  **import** javax.net.ssl.HttpsURLConnection;  
    
  **public class** RequestOperator **extends** Thread {  
    
   **public interface** RequestOperatorListener{  
   **void** success(List<ModelPost> publication);  
   **void** failed(**int** responseCode);  
   }  
    
   **private** RequestOperatorListener **listener**;  
   **private int responseCode**;  
    
   **public void** setListener(RequestOperatorListener listener) {**this**.**listener** = listener;}  
    
   @Override  
   **public void** run(){  
   **super**.run();  
    
   **try**{  
   List<ModelPost> publication = request();  
   **if**(publication!= **null**)  
   success(publication);  
   **else** failed(**responseCode**);  
   } **catch**(IOException e){  
   failed(-1);  
   } **catch**(JSONException e){  
   failed(-2);  
   }  
   }  
    
   **private** List<ModelPost> request() **throws** IOException, JSONException {  
    
   URL obj = **new** URL(**"https://jsonplaceholder.typicode.com/posts"**);  
    
   HttpsURLConnection con = (HttpsURLConnection)obj.openConnection();  
    
   con.setRequestMethod(**"GET"**);  
    
   con.setRequestProperty(**"Content-Type"**, **"application/json"**);  
    
   **responseCode** = con.getResponseCode();  
   System.***out***.println(**"Response Code: "** + **responseCode**);  
    
   InputStreamReader streamReader;  
    
   **if**(**responseCode** == 200){  
   streamReader = **new** InputStreamReader(con.getInputStream());  
   }**else** {  
   streamReader = **new** InputStreamReader(con.getErrorStream());  
   }  
    
   BufferedReader in = **new** BufferedReader(streamReader);  
   String inputLine;  
   StringBuffer response = **new** StringBuffer();  
    
   **while**((inputLine = in.readLine()) != **null**){  
   response.append(inputLine);  
   }  
    
   in.close();  
    
   System.***out***.println(response.toString());  
    
   **if**(**responseCode**==200) {  
   **return** parsingJsonObject(response.toString());  
   } **else return null**;  
   }  
    
   **public** List<ModelPost> parsingJsonObject(String response) **throws** JSONException{  
    
   JSONArray publications = **new** JSONArray(response);  
    
   List<ModelPost> responses = **new** ArrayList<>();  
    
   **for**(**int** i = 0; i< publications.length(); i++){  
   JSONObject object = publications.getJSONObject(i);  
   ModelPost post = **new** ModelPost();  
    
   post.setId(object.optInt(**"id"**, 0));  
   post.setUserId(object.optInt(**"userId"**, 0));  
    
   post.setTitle(object.getString(**"title"**));  
   post.setBodyText(object.getString(**"body"**));  
    
   responses.add(post);  
   }  
    
    
   **try** {  
   *sleep*(2000);  
   } **catch** (InterruptedException e) {  
   e.printStackTrace();  
   }  
    
    
   **return** responses;  
   }  
    
   **private void** failed(**int** code){  
   **if**(**listener** != **null**)  
   **listener**.failed(code);  
   }  
    
   **private void** success(List<ModelPost> publication){  
   **if**(**listener** != **null**)  
   **listener**.success(publication);  
   }  
  }

## *button.xml* file

*<?***xml version="1.0" encoding="utf-8"***?>*<**selector xmlns:android="http://schemas.android.com/apk/res/android"**>  
  
 <**item  
 android:state\_pressed="true"  
 android:drawable="@drawable/button\_pressed"**/>  
  
 <**item  
 android:drawable="@drawable/button\_default"**/>  
</**selector**>

## *button\_default.xml* file

*<?***xml version="1.0" encoding="utf-8"***?>*<**shape  
 xmlns:android="http://schemas.android.com/apk/res/android"**>  
  
 <**stroke  
 android:width="1px"  
 android:color="#006030"**/>  
  
 <**corners  
 android:radius="7dp"**/>  
  
 <**padding  
 android:left="1dp"  
 android:right="1dp"  
 android:top="1dp"  
 android:bottom="1dp"**/>  
  
 <**gradient  
 android:startColor="#00ab4c"  
 android:endColor="#FFFFFF"  
 android:angle="-180"** />  
</**shape**>

## *button\_pressed.xml* file

*<?***xml version="1.0" encoding="utf-8"***?>*<**shape  
 xmlns:android="http://schemas.android.com/apk/res/android"**>  
  
 <**stroke  
 android:width="1px"  
 android:color="#007035"**/>  
  
 <**corners  
 android:radius="7dp"**/>  
  
 <**padding  
 android:left="1dp"  
 android:right="1dp"  
 android:top="1dp"  
 android:bottom="1dp"**/>  
  
 <**solid android:color="#00ab4c"**/>  
</**shape**>

## circle.xml file

*<?***xml version="1.0" encoding="utf-8"***?>*<**selector xmlns:android="http://schemas.android.com/apk/res/android"**>  
  
 <**item**>  
 <**shape android:shape="oval"** >  
 <**solid android:color="#000000"**/>  
 </**shape**>  
 </**item**>  
</**selector**>