|  |  |
| --- | --- |
| Подпись и дата |  |
| Инв. № дубл. |  |
| Взам. инв. № |  |
| Подпись и дата |  |
| Инв. № подл. | RU. 17701729. 502620-01 12 1-1ЛУ |

ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ

НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ

«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»

|  |  |  |
| --- | --- | --- |
| СОГЛАСОВАНО  Старший преподаватель департамента программной инженерии Факультета компьютерных наук  Национального исследовательского университета «Высшая школа экономики»  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ О.В.Максименкова  «\_\_» \_\_\_\_\_\_\_\_\_\_\_ 2016 г. | УТВЕРЖДАЮ   |  | | --- | | Академический руководитель  Образовательной программы  «Программная инженерия» |   \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ В.В.Шилов  «\_\_» \_\_\_\_\_\_\_\_\_\_\_ 2016 г. |

**Android приложение для создания мультимедиа-экскурсий**

**Текст программы**

**ЛИСТ УТВЕРЖДЕНИЯ**

RU. 17701729. 502620-01 12 1-1ЛУ

Исполнитель: студентка группы БПИ142

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/А.А.Репина/

«\_\_» \_\_\_\_\_\_\_\_\_\_\_ 2016 г.

2016

|  |  |
| --- | --- |
| Подпись и дата |  |
| Инв. № дубл. |  |
| Взам. инв. № |  |
| Подпись и дата |  |
| Инв. № подл. | RU. 17701729. 502620-0112 |

**УТВЕРЖДЕНО**

**RU. 17701729. 502620-01 12 1-1-ЛУ**

|  |  |
| --- | --- |
|  |  |

**Android приложение для создания мультимедиа-экскурсий**

**Текст программы**

RU. 17701729. 502620-0112 1-1

Листов 56

2016

Содержание

[Содержание 2](#_Toc441771757)

[Класс VideoActivity 3](#_Toc441771758)

[Класс Utilities 7](#_Toc441771759)

[Класс Upload 8](#_Toc441771760)

[Класс ViewHolder 10](#_Toc441771761)

[Класс TextActivity 11](#_Toc441771762)

[Класс SwipeListView 13](#_Toc441771763)

[Класс Settings 15](#_Toc441771764)

[Класс PlayActivity 16](#_Toc441771765)

[Класс PhotoActivity 18](#_Toc441771766)

[Класс ObjectActivity 22](#_Toc441771767)

[Класс MapsActivity 27](#_Toc441771768)

[Класс MainActivity 29](#_Toc441771769)

[Класс ConnectionDetector 37](#_Toc441771770)

[Класс LoadActivity 37](#_Toc441771771)

[Класс DataClass 43](#_Toc441771772)

[Класс HAA 43](#_Toc441771773)

[Класс Constants 47](#_Toc441771774)

[Класс Excursion 47](#_Toc441771775)

[Класс ExcursionObject 47](#_Toc441771776)

[Интерфейс DropboxData 48](#_Toc441771777)

[Класс AudioActivity 48](#_Toc441771778)

[Интерфейс API\_Listener 54](#_Toc441771779)

[Лист регистрации изменений 55](#_Toc441771780)

[Список используемой литературы 56](#_Toc441771782)

# Класс VideoActivity

**package** com.example.anast.app;  
  
**import** android.content.Context;  
**import** android.content.Intent;  
**import** android.content.pm.ActivityInfo;  
**import** android.os.Bundle;  
**import** android.os.Handler;  
**import** android.support.v4.app.FragmentActivity;  
**import** android.view.KeyEvent;  
**import** android.view.LayoutInflater;  
**import** android.view.Menu;  
**import** android.view.MenuItem;  
**import** android.view.MotionEvent;  
**import** android.view.View;  
**import** android.view.ViewGroup;  
**import** android.widget.BaseAdapter;  
**import** android.widget.Button;  
**import** android.widget.CheckBox;  
**import** android.widget.ListView;  
**import** android.widget.TextView;  
**import** android.widget.Toast;  
**import** android.widget.VideoView;  
  
**import** java.util.ArrayList;  
**import** java.util.List;  
  
**public class** VideoActivity **extends** FragmentActivity {  
  
 **private** VideoView **player**;*//видеоплеер* **private boolean bVideoIsBeingTouched** = **false**;*//нажал ли пользователь на видеоплеер* **private** Handler **mHandler** = **new** Handler();*//работа с видео на плеере* **private int length**;*//время, когда видео было остановлено* **public static** List<VideoNum> *checkboxes* = **new** ArrayList<>();*//номера выбранных чекбоксов у видео* **public static** List<ViewHolder> *viewholders* = **new** ArrayList<>();*//список всех видео-хранилищ* **private int obpos**, **pos**;*//коды экскурсии и объекта* @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 setRequestedOrientation(ActivityInfo.***SCREEN\_ORIENTATION\_LANDSCAPE***);  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_video***);  
 Toast.*makeText*(getApplicationContext(), **"Choose video files now"**, Toast.***LENGTH\_LONG***).show();  
 **try** {  
 **pos** = Integer.*parseInt*(getIntent().getExtras().getString(**"position"**));  
 } **catch** (Exception e) {  
 **pos** = 0;  
 }  
 **try** {  
 **obpos** = Integer.*parseInt*(getIntent().getExtras().getString(**"obposition"**));  
 } **catch** (Exception e) {  
 **obpos** = 0;  
 }  
 **player** = (VideoView) findViewById(R.id.***videofromlist***);  
 **player**.setOnTouchListener(**new** View.OnTouchListener() {  
 @Override  
 **public boolean** onTouch(View v, MotionEvent event) {  
 **if** (!**bVideoIsBeingTouched**) {  
 **bVideoIsBeingTouched** = **true**;  
 **if** (**player**.isPlaying()) {  
 **player**.pause();  
 **length** = **player**.getCurrentPosition();  
 } **else** {  
 **player**.seekTo(**length**);  
 **player**.start();  
 }  
 **mHandler**.postDelayed(**new** Runnable() {  
 **public void** run() {  
 **bVideoIsBeingTouched** = **false**;  
 }  
 }, 100);  
 }  
 **return true**;  
 }  
 });  
 Button back = (Button) findViewById(R.id.***backbutstep2***);  
 back.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View view) {  
 MainActivity.*exlist*.get(**pos**).**objects**.get(**obpos**).**obselectedvideos**.clear();  
 *checkboxes*.clear();  
 *viewholders*.clear();  
 finish();  
 finishActivity(2);  
 Intent intent = **new** Intent(VideoActivity.**this**, PhotoActivity.**class**);  
 intent.putExtra(**"position"**, Integer.*toString*(**pos**));  
 intent.putExtra(**"obposition"**, Integer.*toString*(**obpos**));  
 startActivity(intent);  
 *// set the animation to move once the button is clicked* overridePendingTransition(R.anim.***slide\_in***, R.anim.***slide\_out***);  
 }  
 });  
 ListView listView = (ListView) **this**.findViewById(R.id.***videolist***);  
 listView.setAdapter(**new** VideoGalleryAdapter());  
 **final** Button selectBtn = (Button) findViewById(R.id.***selectBtnstep2***);  
 selectBtn.setOnClickListener(**new** View.OnClickListener() {  
 **public void** onClick(View v) {  
 Intent in = **new** Intent(VideoActivity.**this**, AudioActivity.**class**);  
 in.putExtra(**"position"**, Integer.*toString*(**pos**));  
 in.putExtra(**"obposition"**, Integer.*toString*(**obpos**));  
 startActivityForResult(in, 3);  
 *// set the animation to move once the button is clicked* overridePendingTransition(R.anim.***slide\_forw***, R.anim.***slide\_back***);  
 }  
 });  
 }  
  
 *//класс нужный для добавления номера конкретного видео в список выбранных пользователем* **class** VideoNum {  
 **int num**;  
 }  
  
 *//класс, отвечающий за отображения списка видео на gridview, нажатие на элементы, возврат исходных значений при возврате из последующей активности* **class** VideoGalleryAdapter **extends** BaseAdapter {  
 LayoutInflater **inflater**;  
  
 **public** VideoGalleryAdapter() {  
 **inflater** = (LayoutInflater) getSystemService(Context.***LAYOUT\_INFLATER\_SERVICE***);  
 }  
  
 **public int** getCount() {  
 **return** LoadActivity.*videosList*.size();  
 }  
  
 **public** Object getItem(**int** position) {  
 **return** LoadActivity.*videosList*.get(position);  
 }  
  
 **public long** getItemId(**int** position) {  
 **return** position;  
 }  
  
 *//инициализация, загрузка видео с карты памяти* **public** View getView(**int** position, View convertView, ViewGroup parent) {  
 **final** ViewHolder holder;  
 **if** (convertView == **null**) {  
 LayoutInflater vi = (LayoutInflater) getSystemService(Context.***LAYOUT\_INFLATER\_SERVICE***);  
 convertView = vi.inflate(R.layout.***video\_info***, **null**);  
 holder = **new** ViewHolder();  
 holder.**code** = (TextView) convertView.findViewById(R.id.***code***);  
 holder.**name** = (CheckBox) convertView.findViewById(R.id.***checkBox1***);  
 convertView.setTag(holder);  
 } **else** holder = (ViewHolder) convertView.getTag();  
 DataClass vid = LoadActivity.*videosList*.get(position);  
 **boolean** flag = **false**;  
 **for** (**int** q = 0; q < *checkboxes*.size(); q++)  
 **if** (*checkboxes*.get(q).**num** == position) {  
 vid.setSelected(**true**);  
 flag = **true**;  
 }  
 **if** (!flag)  
 vid.setSelected(**false**);  
 holder.**name**.setOnClickListener(**new** View.OnClickListener() {  
 **public void** onClick(View v) {  
 CheckBox cb = (CheckBox) v;  
 DataClass w = (DataClass) cb.getTag();  
 **int** id = 0;  
 **for** (**int** i = 0; i < LoadActivity.*videosList*.size(); i++) {  
 **if** (LoadActivity.*videosList*.get(i).**filePath**.equals(w.**filePath**)) {  
 id = i;  
 **break**;  
 }  
 }  
 **if** (LoadActivity.*videosList*.get(id).isSelected()) {  
 **player**.pause();  
 **length** = **player**.getCurrentPosition();  
 **for** (**int** k = 0; k < MainActivity.*exlist*.get(**pos**).**objects**.get(**obpos**).**obselectedvideos**.size(); k++)  
 **if** (MainActivity.*exlist*.get(**pos**).**objects**.get(**obpos**).**obselectedvideos**.get(k).**id** == LoadActivity.*videosList*.get(id).**id**) {  
 MainActivity.*exlist*.get(**pos**).**objects**.get(**obpos**).**obselectedvideos**.remove(k);  
 **break**;  
 }  
 cb.setChecked(**false**);  
 LoadActivity.*videosList*.get(id).setSelected(**false**);  
 **for** (**int** q = 0; q < *checkboxes*.size(); q++)  
 **if** (*checkboxes*.get(q).**num** == id) {  
 *checkboxes*.remove(q);  
 q--;  
 }  
 } **else** {  
 cb.setChecked(**true**);  
 **player**.setVideoPath(w.**filePath**);  
 **try** {  
 **player**.start();  
 } **catch** (Exception e) {  
 Toast.*makeText*(getApplicationContext(), **"Sorry, it is impossible to play your video"**, Toast.***LENGTH\_LONG***).show();  
 }  
 LoadActivity.*videosList*.get(id).setSelected(**true**);  
 MainActivity.*exlist*.get(**pos**).**objects**.get(**obpos**).**obselectedvideos**.add(LoadActivity.*videosList*.get(id));  
 VideoNum e = **new** VideoNum();  
 e.**num** = id;  
 *checkboxes*.add(e);  
 }  
 }  
 });  
 holder.**code**.setText(vid.**title**);  
 holder.**name**.setChecked(vid.isSelected());  
 holder.**name**.setTag(vid);  
 *viewholders*.add(holder);  
 **return** convertView;  
 }  
 }  
  
 @Override  
 **public boolean** onCreateOptionsMenu(Menu menu) {  
 getMenuInflater().inflate(R.menu.***menu\_video***, menu);  
 **return true**;  
 }  
  
 @Override  
 **public boolean** onOptionsItemSelected(MenuItem item) {  
 **int** id = item.getItemId();  
 **if** (id == R.id.***action\_settings***) {  
 **return true**;  
 }  
 **return super**.onOptionsItemSelected(item);  
 }  
  
 *//обработка нажатия на аппаратную кнопку назад* @Override  
 **public boolean** onKeyDown(**int** keyCode, KeyEvent event) {  
 **if** (keyCode == KeyEvent.***KEYCODE\_BACK***) {  
 MainActivity.*exlist*.get(**pos**).**objects**.get(**obpos**).**obselectedvideos**.clear();  
 *checkboxes*.clear();  
 *viewholders*.clear();  
 finish();  
 finishActivity(2);  
 Intent intent = **new** Intent(**this**, PhotoActivity.**class**);  
 intent.putExtra(**"position"**, Integer.*toString*(**pos**));  
 intent.putExtra(**"obposition"**, Integer.*toString*(**obpos**));  
 startActivity(intent);  
 *// set the animation to move once the button is clicked* overridePendingTransition(R.anim.***slide\_in***, R.anim.***slide\_out***);  
 **return true**;  
 }  
 **return false**;  
 }  
}

# Класс Utilities

**package** com.example.anast.app;  
  
  
**public class** Utilities {  
  
 *//милисекунды в формат таймера Hours:Minutes:Seconds* **public** String milliSecondsToTimer(**long** milliseconds) {  
 String finalTimerString = **""**;  
 String secondsString = **""**;  
 **int** hours = (**int**) (milliseconds / (Constants.*MILLISEC* \* Constants.*MINUTE* \* Constants.*MINUTE*));  
 **int** minutes = (**int**) (milliseconds % (Constants.*MILLISEC* \* Constants.*MINUTE* \* Constants.*MINUTE*)) / (Constants.*MILLISEC* \* Constants.*MINUTE*);  
 **int** seconds = (**int**) ((milliseconds % (Constants.*MILLISEC* \* Constants.*MINUTE* \* Constants.*MINUTE*)) % (Constants.*MILLISEC* \* Constants.*MINUTE*) / Constants.*MILLISEC*);  
 **if** (hours > 0)  
 finalTimerString = hours + **":"**;  
 **if** (seconds < 10)  
 secondsString = **"0"** + seconds;  
 **else** secondsString = **""** + seconds;  
 finalTimerString = finalTimerString + minutes + **":"** + secondsString;  
 *// строка с временем для таймера* **return** finalTimerString;  
 }  
  
 *//получаем прогресс в процентах* **public int** getProgressPercentage(**long** currentDuration, **long** totalDuration) {  
 Double percentage = (**double**) 0;  
 **long** currentSeconds = (**int**) (currentDuration / Constants.*MILLISEC*);  
 **long** totalSeconds = (**int**) (totalDuration / Constants.*MILLISEC*);  
 percentage = (((**double**) currentSeconds) / totalSeconds) \* Constants.*PROC*;  
 **return** percentage.intValue();  
 }  
  
 *//возвращаем текущее время таймера в милисекундах* **public int** progressToTimer(**int** progress, **int** totalDuration) {  
 **int** currentDuration = 0;  
 totalDuration = (**int**) (totalDuration / Constants.*MILLISEC*);  
 currentDuration = (**int**) ((((**double**) progress) / Constants.*PROC*) \* totalDuration);  
 **return** currentDuration \* Constants.*MILLISEC*;  
 }  
}

# Класс Upload

**package** com.example.anast.app;  
  
**import** android.app.ProgressDialog;  
**import** android.os.AsyncTask;  
**import** android.widget.Toast;  
  
**import** com.dropbox.client2.DropboxAPI;  
  
**import** java.io.File;  
**import** java.io.FileInputStream;  
**import** java.io.FileWriter;  
  
**public class** Upload **extends** AsyncTask<Void, Long, Boolean> {  
  
 **private** ProgressDialog **mDialog**;*//прогресс процесса загрузки* Settings **set**;*//перемеенная настроек дропбокса* **public** Upload(**int** request\_num, MainActivity activity, DropboxAPI dropboxApi, DropboxAPI<?> api, String dropboxPath) {  
 **set** = **new** Settings(request\_num, activity, dropboxApi, api, dropboxPath);  
 **mDialog** = **new** ProgressDialog(activity);  
 **int** number\_of\_ob\_to\_load = 0;  
 **for** (**int** i = 0; i < MainActivity.*exlist*.size(); i++)  
 **if** (MainActivity.*exlist*.get(i).**selected**)  
 number\_of\_ob\_to\_load++;  
 **mDialog**.setMax(number\_of\_ob\_to\_load);  
 **mDialog**.setMessage(**"Uploading... Please be patient, it will take some time"**);  
 **mDialog**.setProgressStyle(ProgressDialog.***STYLE\_HORIZONTAL***);  
 **mDialog**.setProgress(0);  
 **mDialog**.setCancelable(**false**);  
 **mDialog**.show();  
 }  
  
 *//Подготовка к загрузке* @Override  
 **protected void** onPreExecute() {  
 **super**.onPreExecute();  
  
 }  
  
 *//загрузка объектов в DropBox* @Override  
 **protected** Boolean doInBackground(Void... params) {  
 **final** File tempDropboxDirectory = **set**.**mContext**.getCacheDir();  
 File tempFileToUpload;  
 FileWriter fileWriter;  
 FileInputStream fis;  
 **int** k = 0;  
 **try** {  
 **for** (**int** j = 0; j < MainActivity.*exlist*.size(); j++)*//загружаем все экскурсии* {  
 **if** (MainActivity.*exlist*.get(j).**selected**) {  
 k++;  
 **set**.**mPath** = **"/"** + MainActivity.*exlist*.get(j).**exname** + **"/"**;  
 **for** (**int** q = 0; q < MainActivity.*exlist*.get(j).**objects**.size(); q++) {*//загружаем все объекты 1 экскурсии* tempFileToUpload = File.*createTempFile*(**"file"**, **".txt"**, tempDropboxDirectory);*//загрузка текстового файла* fileWriter = **new** FileWriter(tempFileToUpload);  
 fileWriter.write(MainActivity.*exlist*.get(j).**objects**.get(q).**obobjectText**);  
 fileWriter.close();  
 FileInputStream text = **new** FileInputStream(tempFileToUpload);  
 **set**.**dropboxApi**.putFile(**set**.**mPath** + MainActivity.*exlist*.get(j).**objects**.get(q).**obname** + **"/text.txt"**, text, tempFileToUpload.length(), **null**, **null**);  
 tempFileToUpload = File.*createTempFile*(**"geo"**, **".txt"**, tempDropboxDirectory);*//загрузка geo файла* fileWriter = **new** FileWriter(tempFileToUpload);  
 String s = MainActivity.*exlist*.get(j).**objects**.get(q).**oblatitude** + **" : "** + MainActivity.*exlist*.get(j).**objects**.get(q).**oblongitude**;  
 fileWriter.write(s);  
 fileWriter.close();  
 FileInputStream text1 = **new** FileInputStream(tempFileToUpload);  
 **set**.**dropboxApi**.putFile(**set**.**mPath** + MainActivity.*exlist*.get(j).**objects**.get(q).**obname** + **"/geo.txt"**, text1, tempFileToUpload.length(), **null**, **null**);  
 **for** (**int** i = 0; i < MainActivity.*exlist*.get(j).**objects**.get(q).**obselectedphotos**.size(); i++)*//загружаем фото* {  
 **set**.**mCurrentFileIndex** = i;  
 File file = **new** File(MainActivity.*exlist*.get(j).**objects**.get(q).**obselectedphotos**.get(i).**filePath**);  
 fis = **new** FileInputStream(file);  
 String path = **set**.**mPath** + MainActivity.*exlist*.get(j).**objects**.get(q).**obname** + **"/photo"** + i + **".jpg"**;  
 **set**.**mRequest** = **set**.**mApi**.putFileOverwriteRequest(path, fis, file.length(), **null**);  
 **set**.**mRequest**.upload();  
 **if** (!isCancelled()) {  
 **set**.**mFilesUploaded**++;  
 } **else** {  
 **return false**;  
 }  
 }  
 **for** (**int** i = 0; i < MainActivity.*exlist*.get(j).**objects**.get(q).**obselectedvideos**.size(); i++)*//загружаем видео* {  
 **set**.**mCurrentFileIndex** = i;  
 File file = **new** File(MainActivity.*exlist*.get(j).**objects**.get(q).**obselectedvideos**.get(i).**filePath**);  
 fis = **new** FileInputStream(file);  
 String path = **set**.**mPath** + MainActivity.*exlist*.get(j).**objects**.get(q).**obname** + **"/video"** + i + **".mp4"**;  
 **set**.**mRequest** = **set**.**mApi**.putFileOverwriteRequest(path, fis, file.length(), **null**);  
 **set**.**mRequest**.upload();  
 **if** (!isCancelled()) {  
 **set**.**mFilesUploaded**++;  
 } **else** {  
 **return false**;  
 }  
 }  
 **for** (**int** i = 0; i < MainActivity.*exlist*.get(j).**objects**.get(q).**obselectedaudios**.size(); i++)*//загружаем аудио* {  
 **set**.**mCurrentFileIndex** = i;  
 File file = **new** File(MainActivity.*exlist*.get(j).**objects**.get(q).**obselectedaudios**.get(i).**filePath**);  
 fis = **new** FileInputStream(file);  
 String path = **set**.**mPath** + MainActivity.*exlist*.get(j).**objects**.get(q).**obname** + **"/audio"** + i + **".mp3"**;  
 **set**.**mRequest** = **set**.**mApi**.putFileOverwriteRequest(path, fis, file.length(), **null**);  
 **set**.**mRequest**.upload();  
 **if** (!isCancelled()) {  
 **set**.**mFilesUploaded**++;  
 } **else** {  
 **return false**;  
 }  
 }  
 }  
 publishProgress(Long.*parseLong*(**""** + k));  
 }  
 }  
 **return true**;  
 } **catch** (Exception e) {  
 **set**.**mErrorMsg** = **"The was an error during the upload!"**;  
 **return false**;  
 }  
 }  
  
 *//апдейт прогресса загрузки объектов в DropBox* @Override  
 **protected void** onProgressUpdate(Long... progress) {  
 **mDialog**.setProgress(Integer.*parseInt*(**""** + progress[0]));  
 **super**.onProgressUpdate(progress);  
 }  
  
 *//загрузка завершена, проверяем ее успех* @Override  
 **protected void** onPostExecute(Boolean result) {  
 **mDialog**.dismiss();  
 **if** (result)  
 **set**.**api\_Listener**.onSuccess(**set**.**requestNumber**, result);  
 **else** showToast(**set**.**mErrorMsg**);  
 }  
  
 *//вывод сообщения об ошибке на экран для пользователя* **private void** showToast(String msg) {  
 Toast error = Toast.makeText(set.mContext, msg, Toast.LENGTH\_LONG);  
 error.show();  
 }  
}

# Класс ViewHolder

**package** com.example.anast.app;  
  
**import** android.widget.CheckBox;  
**import** android.widget.TextView;  
  
*//каждый отдельный объект в листах на страницах видео и аудио***public class** ViewHolder {  
 CheckBox **name**;  
 TextView **code**;  
}

# Класс TextActivity

**package** com.example.anast.app;  
  
**import** android.content.Intent;  
**import** android.content.pm.ActivityInfo;  
**import** android.os.Bundle;  
**import** android.support.v7.app.AppCompatActivity;  
**import** android.view.KeyEvent;  
**import** android.view.Menu;  
**import** android.view.MenuItem;  
**import** android.view.View;  
**import** android.widget.Button;  
**import** android.widget.EditText;  
**import** android.widget.Toast;  
  
**public class** TextActivity **extends** AppCompatActivity {  
  
 **private** EditText **ed**;*//текст на экране* **public static** String *text* = **""**;*//хранение введенного пользователем текста* **private int obpos**, **pos**;*//коды экскурсии и объекта* @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 setRequestedOrientation(ActivityInfo.***SCREEN\_ORIENTATION\_LANDSCAPE***);  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_text***);  
 Toast.*makeText*(getApplicationContext(), **"Write text now"**, Toast.***LENGTH\_LONG***).show();  
 **try** {  
 **pos** = Integer.*parseInt*(getIntent().getExtras().getString(**"position"**));  
 } **catch** (Exception e) {  
 **pos** = 0;  
 }  
 **try** {  
 **obpos** = Integer.*parseInt*(getIntent().getExtras().getString(**"obposition"**));  
 } **catch** (Exception e) {  
 **obpos** = 0;  
 }  
 **ed** = (EditText) findViewById(R.id.***editText***);  
 **if** (*text*.length() != 0) {  
 **ed**.setText(*text*);  
 **ed**.setSelection(**ed**.getText().length());  
 }  
 Button back = (Button) findViewById(R.id.***backbutstep5***);  
 back.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View view) {  
 MainActivity.*exlist*.get(**pos**).**objects**.get(**obpos**).**obobjectText** = **""**;  
 *text* = **""**;  
 finish();  
 finishActivity(4);  
 Intent intent = **new** Intent(TextActivity.**this**, AudioActivity.**class**);  
 intent.putExtra(**"position"**, Integer.*toString*(**pos**));  
 intent.putExtra(**"obposition"**, Integer.*toString*(**obpos**));  
 startActivity(intent);  
 *// set the animation to move once the button is clicked* overridePendingTransition(R.anim.***slide\_in***, R.anim.***slide\_out***);  
 }  
 });  
 **final** Button selectBtn = (Button) findViewById(R.id.***selectBtnstep5***);  
 selectBtn.setOnClickListener(**new** View.OnClickListener() {  
 **public void** onClick(View v) {  
 **if** (Conn()) {  
 *text* = **ed**.getText().toString();  
 **if** (*text*.length() == 0) {  
 Toast.*makeText*(getApplicationContext(), **"Please write at least one word"**, Toast.***LENGTH\_LONG***).show();  
 } **else** {  
 MainActivity.*exlist*.get(**pos**).**objects**.get(**obpos**).**obobjectText** = *text*;  
 Intent in = **new** Intent(TextActivity.**this**, MapsActivity.**class**);  
 in.putExtra(**"position"**, Integer.*toString*(**pos**));  
 in.putExtra(**"obposition"**, Integer.*toString*(**obpos**));  
 startActivityForResult(in, 5);  
 *// set the animation to move once the button is clicked* overridePendingTransition(R.anim.***slide\_forw***, R.anim.***slide\_back***);  
 }  
 }  
 }  
 });  
 }  
  
 *//проверка наличия соединения с интернетом* **public boolean** Conn() {  
 *//Создаем пример класса connection detector:* ConnectionDetector cd = **new** ConnectionDetector(getApplicationContext());  
 *//Получаем статус Интернет* **boolean** isInternetPresent = cd.ConnectingToInternet();  
 *//Проверяем Интернет статус:* **if** (isInternetPresent) {  
 **return true**;  
 } **else** {  
 *//Интернет соединения нет просим пользователя подключить Интернет:* Toast.*makeText*(getApplicationContext(), **"You need to turn on the Internet connection!"**, Toast.***LENGTH\_SHORT***).show();  
 **return false**;  
 }  
 }  
  
 @Override  
 **public boolean** onCreateOptionsMenu(Menu menu) {  
 getMenuInflater().inflate(R.menu.***menu\_text***, menu);  
 **return true**;  
 }  
  
 @Override  
 **public boolean** onOptionsItemSelected(MenuItem item) {  
 **int** id = item.getItemId();  
 **if** (id == R.id.***action\_settings***) {  
 **return true**;  
 }  
 **return super**.onOptionsItemSelected(item);  
 }  
  
 *//обработка нажатия на аппаратную кнопку назад* @Override  
 **public boolean** onKeyDown(**int** keyCode, KeyEvent event) {  
 **if** (keyCode == KeyEvent.***KEYCODE\_BACK***) {  
 MainActivity.*exlist*.get(**pos**).**objects**.get(**obpos**).**obobjectText** = **""**;  
 *text* = **""**;  
 finish();  
 finishActivity(4);  
 Intent intent = **new** Intent(**this**, AudioActivity.**class**);  
 intent.putExtra(**"position"**, Integer.*toString*(**pos**));  
 intent.putExtra(**"obposition"**, Integer.*toString*(**obpos**));  
 startActivity(intent);  
 *// set the animation to move once the button is clicked* overridePendingTransition(R.anim.***slide\_in***, R.anim.***slide\_out***);  
 **return true**;  
 }  
 **return false**;  
 }  
}

# Класс SwipeListView

**package** com.example.anast.app;  
  
**import** android.content.Context;  
**import** android.util.DisplayMetrics;  
**import** android.view.GestureDetector;  
**import** android.view.GestureDetector.SimpleOnGestureListener;  
**import** android.view.MotionEvent;  
**import** android.view.View;  
**import** android.widget.ListAdapter;  
**import** android.widget.ListView;  
  
**public class** SwipeListView {  
  
 **private** ListView **list**;  
 **private int REL\_SWIPE\_MIN\_DISTANCE**;  
 **private int REL\_SWIPE\_MAX\_OFF\_PATH**;  
 **private int REL\_SWIPE\_THRESHOLD\_VELOCITY**;  
 **private** Context **m\_Context**;  
 **private** SwipeListViewCallback **m\_Callback**;  
  
 *//свайп-интерфейс для листа объектов* **public interface** SwipeListViewCallback {  
 ListView getListView();  
 **void** onSwipeItem(**boolean** isRight, **int** position);*//свайп отдельного пункта списка* **void** onItemClickListener(ListAdapter adapter, **int** position);*//единичный клик* }  
  
 *//вызываем инициализацию объектов* **public** SwipeListView(Context mContext, SwipeListViewCallback callback) {  
 **if** (callback == **null**) {  
 **try** {  
 **throw new** Exception(**"Activity must be implement SwipeListViewCallback"**);  
 } **catch** (Exception e) {  
 e.printStackTrace();  
 }  
 }  
 init(mContext, callback);  
 }  
  
 *//вызываем инициализацию объектов* **public** SwipeListView(Context mContext) **throws** Exception {  
 **if** (!(mContext **instanceof** SwipeListViewCallback)) {  
 **throw new** Exception(**"Activity must be implement SwipeListViewCallback"**);  
 }  
 init(mContext, (SwipeListViewCallback) mContext);  
 }  
  
 *//инициализация* **protected void** init(Context mContext, SwipeListViewCallback mCallback) {  
 **m\_Context** = mContext;  
 **m\_Callback** = mCallback;  
 }  
  
 *//устанавливаем в качестве обработчика нажатий и свайпов наш созданный класс GestureDetector* **public void** exec() {  
 DisplayMetrics dm = **m\_Context**.getResources().getDisplayMetrics();  
 **REL\_SWIPE\_MIN\_DISTANCE** = (**int**) (120.0f \* dm.**densityDpi** / 160.0f + 0.5);  
 **REL\_SWIPE\_MAX\_OFF\_PATH** = (**int**) (250.0f \* dm.**densityDpi** / 160.0f + 0.5);  
 **REL\_SWIPE\_THRESHOLD\_VELOCITY** = (**int**) (200.0f \* dm.**densityDpi** / 160.0f + 0.5);  
 **list** = **m\_Callback**.getListView();  
 @SuppressWarnings(**"deprecation"**)  
 **final** GestureDetector gestureDetector = **new** GestureDetector(**new** MyGestureDetector());  
 View.OnTouchListener gestureListener = **new** View.OnTouchListener() {  
 **public boolean** onTouch(View v, MotionEvent event) {  
 **return** gestureDetector.onTouchEvent(event);  
 }  
 };  
 **list**.setOnTouchListener(gestureListener);  
 }  
  
 *//нажатие на объект* **private void** myOnItemClick(**int** position) {  
 **if** (position < 0)  
 **return**;  
 **m\_Callback**.onItemClickListener(**list**.getAdapter(), position);  
  
 }  
  
 *//класс-обработчик нажатий и свайпов на объект* **class** MyGestureDetector **extends** SimpleOnGestureListener {  
  
 **private int temp\_position** = -1;  
  
 *//единичное нажатие на объект* @Override  
 **public boolean** onSingleTapUp(MotionEvent e) {  
 **int** pos = **list**.pointToPosition((**int**) e.getX(), (**int**) e.getY());  
 myOnItemClick(pos);  
 **return true**;  
 }  
  
 *//палец опущен на объект* @Override  
 **public boolean** onDown(MotionEvent e) {  
 **temp\_position** = **list**.pointToPosition((**int**) e.getX(), (**int**) e.getY());  
 **return super**.onDown(e);  
 }  
  
 *//произведен свайп по объекту* @Override  
 **public boolean** onFling(MotionEvent e1, MotionEvent e2, **float** velocityX, **float** velocityY) {  
 **if** (Math.*abs*(e1.getY() - e2.getY()) > **REL\_SWIPE\_MAX\_OFF\_PATH**)  
 **return false**;  
 **if** (e1.getX() - e2.getX() > **REL\_SWIPE\_MIN\_DISTANCE** && Math.*abs*(velocityX) > **REL\_SWIPE\_THRESHOLD\_VELOCITY**) {  
 **int** pos = **list**.pointToPosition((**int**) e1.getX(), (**int**) e2.getY());  
 **if** (pos >= 0 && **temp\_position** == pos)  
 **m\_Callback**.onSwipeItem(**false**, pos);  
 } **else if** (e2.getX() - e1.getX() > **REL\_SWIPE\_MIN\_DISTANCE** && Math.*abs*(velocityX) > **REL\_SWIPE\_THRESHOLD\_VELOCITY**) {  
 **int** pos = **list**.pointToPosition((**int**) e1.getX(), (**int**) e2.getY());  
 **if** (pos >= 0 && **temp\_position** == pos)  
 **m\_Callback**.onSwipeItem(**true**, pos);  
 }  
 **return false**;  
 }  
 }  
}

# Класс Settings

**package** com.example.anast.app;  
  
**import** android.content.Context;  
**import** com.dropbox.client2.DropboxAPI;  
  
*//Класс настроек дропбокса***class** Settings  
{  
 **public** DropboxAPI<?> **mApi**;*//Dropbox API* **public** String **mPath**;*//путь к объекту* **public** DropboxAPI.UploadRequest **mRequest**;*//запрос на загрузку* **public** Context **mContext**;*//контекст* **public** String **mErrorMsg**;*//вывод ошибки* **public int mFilesUploaded**;*//кол-во загружаемых файлов* **public int mCurrentFileIndex**;*//номер загружаемого файла* **public** API\_Listener **api\_Listener**;*//обработчик API* **public int requestNumber**;*//номер запроса на загрузку* **public** DropboxAPI **dropboxApi**;*//DropBox API* **public** Settings(**int** request\_num, MainActivity activity, DropboxAPI dropboxApi, DropboxAPI<?> api, String dropboxPath)  
 {  
 **this**.**dropboxApi** = dropboxApi;  
 **mContext** = activity;  
 **api\_Listener** = activity;  
 **requestNumber** = request\_num;  
 **mApi** = api;  
 **mPath** = dropboxPath;  
 **mFilesUploaded** = 0;  
 **mCurrentFileIndex** = 0;  
 }  
}

# Класс PlayActivity

**package** com.example.anast.app;  
  
**import** android.content.Intent;  
**import** android.content.pm.ActivityInfo;  
**import** android.os.Bundle;  
**import** android.support.v7.app.AppCompatActivity;  
**import** android.text.Html;  
**import** android.view.KeyEvent;  
**import** android.view.Menu;  
**import** android.view.MenuItem;  
**import** android.view.View;  
**import** android.widget.Button;  
**import** android.widget.TextView;  
**import** android.widget.Toast;  
  
**import** com.google.android.gms.ads.AdListener;  
  
**public class** PlayActivity **extends** AppCompatActivity {  
  
 **private int** obpos, pos;*//коды экскурсии и объекта* @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 setRequestedOrientation(ActivityInfo.SCREEN\_ORIENTATION\_LANDSCAPE);  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.activity\_play);  
 Toast.makeText(getApplicationContext(), **"That's the final step"**, Toast.LENGTH\_LONG).show();  
 **try** {  
 pos = Integer.parseInt(getIntent().getExtras().getString(**"position"**));  
 } **catch** (Exception e) {  
 pos = 0;  
 }  
 **try** {  
 obpos = Integer.parseInt(getIntent().getExtras().getString(**"obposition"**));  
 } **catch** (Exception e) {  
 obpos = 0;  
 }  
 TextView text = (TextView) findViewById(R.id.***finaltext***);  
 String str = **"<font color='#1b9900'>Your object's name: </font>"** + MainActivity.*exlist*.get(**pos**).**objects**.get(**obpos**).**obname** + **"<br/>"**;  
 String str1 = **"<font color='#1b9900'>The number of photos you've chosen for your object is </font>"** + MainActivity.*exlist*.get(**pos**).**objects**.get(**obpos**).**obselectedphotos**.size() + **"<br/>"**;  
 String str2 = **"<font color='#1b9900'>The number of videos you've chosen for your object is </font>"** + MainActivity.*exlist*.get(**pos**).**objects**.get(**obpos**).**obselectedvideos**.size() + **"<br/>"**;  
 String str3 = **"<font color='#1b9900'>The number of audios you've chosen for your object is </font>"** + MainActivity.*exlist*.get(**pos**).**objects**.get(**obpos**).**obselectedaudios**.size() + **"<br/>"**;  
 String str4 = **"<font color='#1b9900'>The text you've written for your object is </font>"** + MainActivity.*exlist*.get(**pos**).**objects**.get(**obpos**).**obobjectText** + **"<br/>"**;  
 String str5 = **"<font color='#1b9900'>The geolocation you've chosen for your object is </font>"** + String.*format*(**"%.7f"**, MainActivity.*exlist*.get(**pos**).**objects**.get(**obpos**).**oblatitude**) + **" : "** + String.*format*(**"%.7f"**, MainActivity.*exlist*.get(**pos**).**objects**.get(**obpos**).**oblongitude**) + **"<br/><br/>"**;  
 String str6 = **"<font color='white'><b>If you want to save your object press the button below</b></font>"** + **"<br/>"**;  
 String s = str + str1 + str2 + str3 + str4 + str5 + str6;  
 text.setText(Html.*fromHtml*(s), TextView.BufferType.***SPANNABLE***);  
 Button back = (Button) findViewById(R.id.***backbutstep4***);  
 back.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View view) {  
 finish();  
 finishActivity(6);  
 Intent intent = **new** Intent(PlayActivity.**this**, MapsActivity.**class**);  
 intent.putExtra(**"position"**, Integer.*toString*(**pos**));  
 intent.putExtra(**"obposition"**, Integer.*toString*(**obpos**));  
 startActivity(intent);  
 *// set the animation to move once the button is clicked* overridePendingTransition(R.anim.***slide\_in***, R.anim.***slide\_out***);  
 }  
 });  
 **final** Button selectBtn = (Button) findViewById(R.id.***selectBtnstep4***);  
 selectBtn.setOnClickListener(**new** View.OnClickListener() {  
 **public void** onClick(View v) {  
 **try** {  
 LoadActivity.*checkboxes*.clear();  
 VideoActivity.*checkboxes*.clear();  
 VideoActivity.*viewholders*.clear();  
 AudioActivity.*checkboxes*.clear();  
 AudioActivity.*viewholders*.clear();  
 TextActivity.*text* = **""**;  
 MapsActivity.*flag* = **false**;  
 finish();  
 finishActivity(5);  
 ObjectActivity.mInterstitialAd.setAdListener(**new** AdListener() {  
 @Override  
 **public void** onAdClosed() {  
 Intent intent = **new** Intent(PlayActivity.**this**, ObjectActivity.**class**);  
 intent.putExtra(**"mode"**, **"-2"**);  
 intent.putExtra(**"position"**, Integer.toString(pos));  
 startActivity(intent);  
 *// set the animation to move once the button is clicked* overridePendingTransition(R.anim.slide\_forw, R.anim.slide\_back);  
 }  
 });  
 **if** (ObjectActivity.mInterstitialAd.isLoaded())  
 ObjectActivity.mInterstitialAd.show();  
 **else** {  
 Intent intent = **new** Intent(PlayActivity.**this**, ObjectActivity.**class**);  
 intent.putExtra(**"mode"**, **"-2"**);  
 intent.putExtra(**"position"**, Integer.toString(pos));  
 startActivity(intent);  
 *// set the animation to move once the button is clicked* overridePendingTransition(R.anim.slide\_forw, R.anim.slide\_back);  
 }  
 } **catch** (Exception e) {  
 e.printStackTrace();  
 }  
 }  
 });  
 }  
  
 @Override  
 **public boolean** onCreateOptionsMenu(Menu menu) {  
 getMenuInflater().inflate(R.menu.menu\_play, menu);  
 **return true**;  
 }  
  
 @Override  
 **public boolean** onOptionsItemSelected(MenuItem item) {  
 **int** id = item.getItemId();  
 **if** (id == R.id.action\_settings) {  
 **return true**;  
 }  
 **return super**.onOptionsItemSelected(item);  
 }  
  
 *//обработка нажатия на аппаратную кнопку назад* @Override  
 **public boolean** onKeyDown(**int** keyCode, KeyEvent event) {  
 **if** (keyCode == KeyEvent.KEYCODE\_BACK) {  
 finish();  
 finishActivity(6);  
 Intent intent = **new** Intent(**this**, MapsActivity.**class**);  
 intent.putExtra(**"position"**, Integer.toString(pos));  
 intent.putExtra(**"obposition"**, Integer.toString(obpos));  
 startActivity(intent);  
 *// set the animation to move once the button is clicked* overridePendingTransition(R.anim.slide\_in, R.anim.slide\_out);  
 **return true**;  
 }  
 **return false**;  
 }  
}

# Класс PhotoActivity

**package** com.example.anast.app;  
  
**import** android.content.Context;  
**import** android.content.Intent;  
**import** android.content.pm.ActivityInfo;  
**import** android.graphics.Bitmap;  
**import** android.graphics.BitmapFactory;  
**import** android.net.Uri;  
**import** android.os.Bundle;  
**import** android.support.v7.app.AppCompatActivity;  
**import** android.view.KeyEvent;  
**import** android.view.LayoutInflater;  
**import** android.view.View;  
**import** android.view.ViewGroup;  
**import** android.widget.BaseAdapter;  
**import** android.widget.Button;  
**import** android.widget.CheckBox;  
**import** android.widget.GridView;  
**import** android.widget.ImageView;  
**import** android.widget.Toast;  
  
**import** com.squareup.picasso.Picasso;  
  
**import** java.io.File;  
  
**public class** PhotoActivity **extends** AppCompatActivity {  
  
 **public** ImageView **showphoto**;*//большое фото* **private int obpos**, **pos**;*//коды экскурсии и объекта* @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setRequestedOrientation(ActivityInfo.***SCREEN\_ORIENTATION\_LANDSCAPE***);  
 setContentView(R.layout.***activity\_photo***);  
 Toast.*makeText*(getApplicationContext(), **"Choose photo files now"**, Toast.***LENGTH\_SHORT***).show();  
 **showphoto** = (ImageView) findViewById(R.id.***imageViewlook***);  
 **try** {  
 **pos** = Integer.*parseInt*(getIntent().getExtras().getString(**"position"**));  
 } **catch** (Exception e) {  
 **pos** = 0;  
 }  
 **try** {  
 **obpos** = Integer.*parseInt*(getIntent().getExtras().getString(**"obposition"**));  
 } **catch** (Exception e) {  
 **obpos** = 0;  
 }  
 GridView gridview = (GridView) findViewById(R.id.***PhoneImageGrid***);  
 gridview.setAdapter(**new** ImageAdapter(**this**));  
 Button back = (Button) findViewById(R.id.***backbut***);  
 back.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View view) {  
 MainActivity.*exlist*.get(**pos**).**objects**.remove(**obpos**);  
 LoadActivity.*checkboxes*.clear();  
 finish();  
 finishActivity(22);  
 Intent intent = **new** Intent(PhotoActivity.**this**, ObjectActivity.**class**);  
 intent.putExtra(**"mode"**, **"-2"**);  
 intent.putExtra(**"position"**, Integer.*toString*(**pos**));  
 startActivity(intent);  
 *// set the animation to move once the button is clicked* overridePendingTransition(R.anim.***slide\_in***, R.anim.***slide\_out***);  
 }  
 });  
 **final** Button selectBtn = (Button) findViewById(R.id.***selectBtn***);  
 selectBtn.setOnClickListener(**new** View.OnClickListener() {  
 **public void** onClick(View v) {  
 Intent intent = **new** Intent(PhotoActivity.**this**, VideoActivity.**class**);  
 intent.putExtra(**"position"**, Integer.*toString*(**pos**));  
 intent.putExtra(**"obposition"**, Integer.*toString*(**obpos**));  
 startActivityForResult(intent, 2);  
 *// set the animation to move once the button is clicked* overridePendingTransition(R.anim.***slide\_forw***, R.anim.***slide\_back***);  
 }  
 });  
 }  
  
 *//класс - номер фото* **class** PhotoNum {  
 **int num**;  
 }  
  
 *//кастомный адаптер* **private class** ImageAdapter **extends** BaseAdapter {  
 **private** Context **mContext**;  
 **private** LayoutInflater **mInflater**;  
  
 **public** ImageAdapter(Context context) {  
 **mContext** = context;  
 **mInflater** = (LayoutInflater) getSystemService(Context.***LAYOUT\_INFLATER\_SERVICE***);  
 }  
  
 @Override  
 **public int** getCount() {  
 **return** LoadActivity.*imagesList*.size();  
 }  
  
 @Override  
 **public** Object getItem(**int** position) {  
 **return null**;  
 }  
  
 @Override  
 **public long** getItemId(**int** position) {  
 **return** 0;  
 }  
  
 @Override  
 **public** View getView(**int** position, View convertView, ViewGroup parent) {  
 **final** ViewHolder holder;  
 **if** (convertView == **null**) {  
 holder = **new** ViewHolder();  
 convertView = **mInflater**.inflate(R.layout.***galeryitem***, **null**);  
 holder.**imageview** = (ImageView) convertView.findViewById(R.id.***thumbImage***);  
 holder.**checkbox** = (CheckBox) convertView.findViewById(R.id.***itemCheckBox***);  
 convertView.setTag(holder);  
 } **else** {  
 holder = (ViewHolder) convertView.getTag();  
 }  
 DataClass item = LoadActivity.*imagesList*.get(position);  
 **boolean** flag = **false**;  
 **for** (**int** q = 0; q < LoadActivity.*checkboxes*.size(); q++)  
 **if** (LoadActivity.*checkboxes*.get(q).**num** == position) {  
 item.setSelected(**true**);  
 flag = **true**;  
 }  
 **if** (!flag) {  
 item.setSelected(**false**);  
 }  
 Uri uri = Uri.*fromFile*(**new** File(LoadActivity.*imagesList*.get(position).**filePath**));  
 holder.**checkbox**.setId(position);  
 holder.**imageview**.setId(position);  
 holder.**checkbox**.setOnClickListener(**new** View.OnClickListener() {  
 **public void** onClick(View v) {  
 CheckBox cb = (CheckBox) v;  
 **int** id = cb.getId();  
 **if** (LoadActivity.*imagesList*.get(id).isSelected()) {  
 **for** (**int** k = 0; k < MainActivity.*exlist*.get(**pos**).**objects**.get(**obpos**).**obselectedphotos**.size(); k++)  
 **if** (MainActivity.*exlist*.get(**pos**).**objects**.get(**obpos**).**obselectedphotos**.get(k).**id** == LoadActivity.*imagesList*.get(id).**id**) {  
 MainActivity.*exlist*.get(**pos**).**objects**.get(**obpos**).**obselectedphotos**.remove(k);  
 **break**;  
 }  
 cb.setChecked(**false**);  
 LoadActivity.*imagesList*.get(id).setSelected(**false**);  
 **for** (**int** q = 0; q < LoadActivity.*checkboxes*.size(); q++)  
 **if** (LoadActivity.*checkboxes*.get(q).**num** == id) {  
 LoadActivity.*checkboxes*.remove(q);  
 q--;  
 }  
 } **else** {  
 cb.setChecked(**true**);  
 LoadActivity.*imagesList*.get(id).setSelected(**true**);  
 MainActivity.*exlist*.get(**pos**).**objects**.get(**obpos**).**obselectedphotos**.add(LoadActivity.*imagesList*.get(id));  
 PhotoNum e = **new** PhotoNum();  
 e.**num** = id;  
 LoadActivity.*checkboxes*.add(e);  
 }  
 Bitmap myBitmap = BitmapFactory.*decodeFile*(LoadActivity.*imagesList*.get(id).**filePath**);  
 **showphoto**.setImageBitmap(myBitmap);  
 }  
 });  
 Picasso.*with*(PhotoActivity.**this**)  
 .load(uri)  
 .resize(120, 120)  
 .centerCrop()  
 .into(holder.**imageview**);  
 holder.**checkbox**.setChecked(item.isSelected());  
 **return** convertView;  
 }  
 }  
  
 *//конкретное фото с его checkboxом на экране* **class** ViewHolder {  
 ImageView **imageview**;  
 CheckBox **checkbox**;  
 }  
  
 *//обработка нажатия на аппаратную кнопку назад* @Override  
 **public boolean** onKeyDown(**int** keyCode, KeyEvent event) {  
 **if** (keyCode == KeyEvent.***KEYCODE\_BACK***) {  
 MainActivity.*exlist*.get(**pos**).**objects**.remove(**obpos**);  
 LoadActivity.*checkboxes*.clear();  
 finish();  
 finishActivity(22);  
 Intent intent = **new** Intent(**this**, ObjectActivity.**class**);  
 intent.putExtra(**"mode"**, **"-2"**);  
 intent.putExtra(**"position"**, Integer.*toString*(**pos**));  
 startActivity(intent);  
 *// set the animation to move once the button is clicked* overridePendingTransition(R.anim.***slide\_in***, R.anim.***slide\_out***);  
 **return true**;  
 }  
 **return false**;  
 }  
}

# Класс ObjectActivity

**package** com.example.anast.app;  
  
**import** android.app.AlertDialog;  
**import** android.content.Intent;  
**import** android.content.pm.ActivityInfo;  
**import** android.os.Bundle;  
**import** android.support.v7.app.AppCompatActivity;  
**import** android.util.SparseArray;  
**import** android.view.KeyEvent;  
**import** android.view.LayoutInflater;  
**import** android.view.Menu;  
**import** android.view.MenuItem;  
**import** android.view.View;  
**import** android.view.ViewGroup;  
**import** android.widget.AdapterView;  
**import** android.widget.BaseAdapter;  
**import** android.widget.Button;  
**import** android.widget.EditText;  
**import** android.widget.ListAdapter;  
**import** android.widget.ListView;  
**import** android.widget.TextView;  
**import** android.widget.Toast;  
  
**import** com.example.anast.app.SwipeListView.SwipeListViewCallback;  
**import** com.google.android.gms.ads.AdRequest;  
**import** com.google.android.gms.ads.InterstitialAd;  
  
**import** java.util.ArrayList;  
**import** java.util.List;  
  
**public class** ObjectActivity **extends** AppCompatActivity **implements** SwipeListViewCallback {  
  
 **private** EditText **propname**;*//имя нового объекта* **private** ListView **list**;*//объекты на экране* **private** MyAdapter **m\_Adapter**;*//адаптер списка всех объектов на экране, чтобы была возможность их удалить* **private** String **code**;*//код страницы, откуда пришли* **private int pos**;*//код экскурсии* **public static** InterstitialAd *mInterstitialAd*;*//рекламная переменная* @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 setRequestedOrientation(ActivityInfo.***SCREEN\_ORIENTATION\_LANDSCAPE***);*//горизонтальная ориентация экрана* **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_object***);  
 **code** = getIntent().getExtras().getString(**"mode"**);  
 **try** {  
 **pos** = Integer.*parseInt*(getIntent().getExtras().getString(**"position"**));  
 } **catch** (Exception e) {  
 **pos** = 0;  
 }  
 *mInterstitialAd* = **new** InterstitialAd(**this**);*//создание рекламы  
 mInterstitialAd*.setAdUnitId(**"ca-app-pub-3197289332528502/5379248076"**);*//номер рекламного полотна  
 //AdRequest.Builder builder = new AdRequest.Builder();//создаем баннер  
 //AdRequest ad = builder.build();//билдим баннер//TODO: CHANGE to normal* AdRequest ad = **new** com.google.android.gms.ads.AdRequest.Builder()  
 .addTestDevice(AdRequest.***DEVICE\_ID\_EMULATOR***)  
 .addTestDevice(**"96F89B24B609ED5DFE69922744B289BF"**).build();  
 *mInterstitialAd*.loadAd(ad);*//загружаем рекламы* TextView exname = (TextView) findViewById(R.id.***exName***);  
 **if** (**code**.equals(**"-1"**)) {*//новая экскурсия, новые объекты* exname.setText(MainActivity.*exlist*.get(**pos**).**exname**);  
 } **else** {*//старая экскурсия, новые объекты* **if** (**code**.equals(**"-2"**)) {*//пришли из PlayActivity* exname.setText(MainActivity.*exlist*.get(**pos**).**exname**);  
 } **else** { *//пришли из MainActivity со старая экскурсия* exname.setText(MainActivity.*exlist*.get(Integer.*parseInt*(**code**)).**exname**);  
 }  
 }  
 **list** = (ListView) findViewById(R.id.***objectslist1***);  
 **list**.setOnItemClickListener(**new** AdapterView.OnItemClickListener() {  
 @Override  
 **public void** onItemClick(AdapterView<?> a, View v, **int** position, **long** id) {  
 String str1 = **"The number of photos you've chosen for your object is: "** + MainActivity.*exlist*.get(**pos**).**objects**.get(position).**obselectedphotos**.size() + **"\n"**;  
 String str2 = **"The number of videos you've chosen for your object is: "** + MainActivity.*exlist*.get(**pos**).**objects**.get(position).**obselectedvideos**.size() + **"\n"**;  
 String str3 = **"The number of audios you've chosen for your object is: "** + MainActivity.*exlist*.get(**pos**).**objects**.get(position).**obselectedaudios**.size() + **"\n"**;  
 String str4 = **"The geolocation you've chosen for your object is: "** + MainActivity.*exlist*.get(**pos**).**objects**.get(position).**oblatitude** + **" : "** + MainActivity.*exlist*.get(**pos**).**objects**.get(position).**oblongitude** + **"\n"**;  
 String str5 = **"The text you've written for your object is: "** + MainActivity.*exlist*.get(**pos**).**objects**.get(position).**obobjectText** + **"\n"**;  
 String s = str1 + str2 + str3 + str4 + str5;  
 AlertDialog.Builder adb = **new** AlertDialog.Builder(ObjectActivity.**this**);  
 adb.setTitle(MainActivity.*exlist*.get(**pos**).**objects**.get(position).**obname**);  
 adb.setMessage(s);  
 adb.setPositiveButton(**"Ok"**, **null**);  
 adb.show();  
 }  
 });  
 SwipeListView l = **new** SwipeListView(**this**, **this**);  
 l.exec();  
 **try** {  
 **if** (MainActivity.*exlist*.get(**pos**).**objects**.size() != 0) {  
 **m\_Adapter** = **new** MyAdapter();  
 **for** (**int** i = 0; i < MainActivity.*exlist*.get(**pos**).**objects**.size(); i++) {  
 **m\_Adapter**.addItem(MainActivity.*exlist*.get(**pos**).**objects**.get(i).**obname**);  
 }  
 **list**.setAdapter(**m\_Adapter**);  
 }  
 } **catch** (Exception e) {  
 e.printStackTrace();  
 }  
 Button del = (Button) findViewById(R.id.***delobbut***);  
 del.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View view) {  
 Toast.*makeText*(getApplicationContext(), **"Tap on object you want to delete and then swipe from right to left"**, Toast.***LENGTH\_SHORT***).show();  
 }  
 });  
 **propname** = (EditText) findViewById(R.id.***obName***);  
 Button new\_ob = (Button) findViewById(R.id.***newobbut***);  
 new\_ob.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View view) {  
 **if** (Conn()) {  
 **if** (**propname**.getText().length() == 0)  
 Toast.*makeText*(getApplicationContext(), **"Enter the object's name"**, Toast.***LENGTH\_SHORT***).show();  
 **else** {  
 ExcursionObject eo = **new** ExcursionObject();  
 eo.**obname** = **propname**.getText().toString();  
 MainActivity.*exlist*.get(**pos**).**objects**.add(eo);  
 Intent intent = **new** Intent(ObjectActivity.**this**, PhotoActivity.**class**);  
 intent.putExtra(**"position"**, Integer.*toString*(**pos**));  
 intent.putExtra(**"obposition"**, Integer.*toString*(MainActivity.*exlist*.get(**pos**).**objects**.size() - 1));  
 startActivityForResult(intent, 22);  
 *// set the animation to move once the button is clicked* overridePendingTransition(R.anim.***slide\_forw***, R.anim.***slide\_back***);  
 }  
 }  
 }  
 });  
 Button menu = (Button) findViewById(R.id.***menubut***);  
 menu.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View view) {  
 finish();  
 finishActivity(21);  
 Intent intent = **new** Intent(ObjectActivity.**this**, MainActivity.**class**);  
 startActivity(intent);  
 *// set the animation to move once the button is clicked* overridePendingTransition(R.anim.***slide\_in***, R.anim.***slide\_out***);  
 }  
 });  
 }  
  
  
 @Override  
 **public boolean** onCreateOptionsMenu(Menu menu) {  
 getMenuInflater().inflate(R.menu.***menu\_ob***, menu);  
 **return true**;  
 }  
  
 @Override  
 **public boolean** onOptionsItemSelected(MenuItem item) {  
 **int** id = item.getItemId();  
 **if** (id == R.id.***action\_settings***) {  
 **return true**;  
 }  
 **return super**.onOptionsItemSelected(item);  
 }  
  
 @Override  
 **public** ListView getListView() {  
 **return list**;  
 }  
  
 *//свайп по элементу списка объектов* @Override  
 **public void** onSwipeItem(**boolean** isRight, **int** position) {  
 **m\_Adapter**.onSwipeItem(isRight, position);  
 }  
  
 *//нажатие на элемент списка объектов* @Override  
 **public void** onItemClickListener(ListAdapter adapter, **int** position) {  
 }  
  
 *//проверка наличия соединения с интернетом* **public boolean** Conn() {  
 *//Создаем пример класса connection detector:* ConnectionDetector cd = **new** ConnectionDetector(getApplicationContext());  
 *//Получаем статус Интернет* **boolean** isInternetPresent = cd.ConnectingToInternet();  
 *//Проверяем Интернет статус:* **if** (isInternetPresent) {  
 **return true**;  
 } **else** {  
 *//Интернет соединения нет просим пользователя подключить Интернет:* Toast.*makeText*(getApplicationContext(), **"You need to turn on the Internet connection!"**, Toast.***LENGTH\_SHORT***).show();  
 **return false**;  
 }  
 }  
  
 *//класс, реализующий удаление элемента из списка по свайпу* **public class** MyAdapter **extends** BaseAdapter {  
  
 **private final int invalid** = -1;  
 **protected int delete\_pos** = -1;  
  
 **protected** List<String> **m\_List**;  
  
 **public** MyAdapter() {  
 **m\_List** = **new** ArrayList<>();  
 }  
  
 **public void** addItem(String item) {  
 **m\_List**.add(item);  
 notifyDataSetChanged();  
 }  
  
 **public void** onSwipeItem(**boolean** isRight, **int** position) {  
 **if** (!isRight) {  
 **delete\_pos** = position;  
 } **else if** (**delete\_pos** == position) {  
 **delete\_pos** = **invalid**;  
 }  
 notifyDataSetChanged();  
 }  
  
 **public void** deleteItem(**int** posi) {  
 **m\_List**.remove(posi);  
 MainActivity.*exlist*.get(**pos**).**objects**.remove(posi);  
 **delete\_pos** = **invalid**;  
 notifyDataSetChanged();  
 }  
  
 @Override  
 **public int** getCount() {  
 **return m\_List**.size();  
 }  
  
 @Override  
 **public** String getItem(**int** position) {  
 **return m\_List**.get(position);  
 }  
  
 @Override  
 **public long** getItemId(**int** position) {  
 **return** 0;  
 }  
  
 @Override  
 **public** View getView(**final int** position, View convertView, ViewGroup parent) {  
 **if** (convertView == **null**) {  
 convertView = LayoutInflater.*from*(ObjectActivity.**this**).inflate(R.layout.***obitem***, **null**);  
 }  
 TextView text = ViewHolderPattern.*get*(convertView, R.id.***text***);  
 Button delete = ViewHolderPattern.*get*(convertView, R.id.delete);  
 **if** (delete\_pos == position) {  
 delete.setVisibility(View.VISIBLE);  
 } **else** delete.setVisibility(View.GONE);  
 delete.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 deleteItem(position);  
 }  
 });  
 text.setText(getItem(position));  
 **return** convertView;  
 }  
 }  
  
 *//класс, свзанный с удалением элемента из списка из определенного holdera* **public static class** ViewHolderPattern {  
 @SuppressWarnings(**"unchecked"**)  
 **public static** <T **extends** View> T get(View view, **int** id) {  
 SparseArray<View> viewHolder = (SparseArray<View>) view.getTag();  
 **if** (viewHolder == **null**) {  
 viewHolder = **new** SparseArray<>();  
 view.setTag(viewHolder);  
 }  
 View childView = viewHolder.get(id);  
 **if** (childView == **null**) {  
 childView = view.findViewById(id);  
 viewHolder.put(id, childView);  
 }  
 **return** (T) childView;  
 }  
 }  
  
 *//анализатор нажатия на аппаратную кнопку назад* @Override  
 **public boolean** onKeyDown(**int** keyCode, KeyEvent event) {  
 **if** (keyCode == KeyEvent.KEYCODE\_BACK) {  
 finish();  
 finishActivity(21);  
 Intent intent = **new** Intent(ObjectActivity.**this**, MainActivity.**class**);  
 startActivity(intent);  
 *// set the animation to move once the button is clicked* overridePendingTransition(R.anim.slide\_in, R.anim.slide\_out);  
 **return true**;  
 }  
 **return false**;  
 }  
}

# Класс MapsActivity

**package** com.example.anast.app;  
  
**import** android.content.Intent;  
**import** android.content.pm.ActivityInfo;  
**import** android.os.Bundle;  
**import** android.support.v4.app.FragmentActivity;  
**import** android.view.KeyEvent;  
**import** android.view.Menu;  
**import** android.view.View;  
**import** android.widget.Button;  
**import** android.widget.Toast;  
  
**import** com.google.android.gms.maps.CameraUpdateFactory;  
**import** com.google.android.gms.maps.GoogleMap;  
**import** com.google.android.gms.maps.SupportMapFragment;  
**import** com.google.android.gms.maps.model.LatLng;  
**import** com.google.android.gms.maps.model.MarkerOptions;  
  
**public class** MapsActivity **extends** FragmentActivity {  
  
 **private** GoogleMap **googleMap**;*//карта* **public static boolean** *flag*;*//было ли нажатие на карту* **private int** obpos, **pos**;*//коды экскурсии и объекта* @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 setRequestedOrientation(ActivityInfo.***SCREEN\_ORIENTATION\_LANDSCAPE***);  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_maps***);  
 **try** {  
 **pos** = Integer.*parseInt*(getIntent().getExtras().getString(**"position"**));  
 } **catch** (Exception e) {  
 **pos** = 0;  
 }  
 **try** {  
 obpos = Integer.*parseInt*(getIntent().getExtras().getString(**"obposition"**));  
 } **catch** (Exception e) {  
 obpos = 0;  
 }  
 Toast.*makeText*(getApplicationContext(), **"Tap on the map to choose the location"**, Toast.***LENGTH\_LONG***).show();  
 SupportMapFragment supportMapFragment = (SupportMapFragment) getSupportFragmentManager().findFragmentById(R.id.***map***);  
 **googleMap** = supportMapFragment.getMap();*//создаем карту* **if** (*flag*) {  
 LatLng latLng = **new** LatLng(MainActivity.*exlist*.get(**pos**).**objects**.get(obpos).**oblatitude**, MainActivity.*exlist*.get(**pos**).**objects**.get(obpos).**oblongitude**);*//метка уже была размещена* MarkerOptions markerOptions = **new** MarkerOptions();*//создаем маркер* markerOptions.position(latLng);*//задаем позицию* markerOptions.title(MainActivity.*exlist*.get(**pos**).**objects**.get(obpos).**obname** + **" "** + latLng.**latitude** + **" : "** + latLng.**longitude**);*//задаем название* **googleMap**.clear();*//очищаем позицию, которая была до* **googleMap**.animateCamera(CameraUpdateFactory.*newLatLng*(latLng));*//анимируем переход к новой метке* **googleMap**.addMarker(markerOptions);*//помещаем метку на карту* } **else** *flag* = **false**;  
 Button back = (Button) findViewById(R.id.***backbutstep6***);  
 back.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View view) {  
 MainActivity.*exlist*.get(**pos**).**objects**.get(obpos).**oblatitude** = 0.0;  
 MainActivity.*exlist*.get(**pos**).**objects**.get(obpos).**oblongitude** = 0.0;  
 *flag* = **false**;  
 finish();  
 finishActivity(5);  
 Intent intent = **new** Intent(MapsActivity.**this**, TextActivity.**class**);  
 intent.putExtra(**"position"**, Integer.*toString*(**pos**));  
 intent.putExtra(**"obposition"**, Integer.*toString*(obpos));  
 startActivity(intent);  
 *// set the animation to move once the button is clicked* overridePendingTransition(R.anim.***slide\_in***, R.anim.***slide\_out***);  
 }  
 });  
 **final** Button selectBtn = (Button) findViewById(R.id.***selectBtnstep6***);  
 selectBtn.setOnClickListener(**new** View.OnClickListener() {  
 **public void** onClick(View v) {  
 **if** (!*flag*) {  
 Toast.*makeText*(getApplicationContext(), **"Please choose the location"**, Toast.***LENGTH\_LONG***).show();  
 } **else** {  
 Intent in = **new** Intent(MapsActivity.**this**, PlayActivity.**class**);  
 in.putExtra(**"position"**, Integer.*toString*(**pos**));  
 in.putExtra(**"obposition"**, Integer.*toString*(obpos));  
 startActivityForResult(in, 6);  
 *// set the animation to move once the button is clicked* overridePendingTransition(R.anim.***slide\_forw***, R.anim.***slide\_back***);  
 }  
 }  
 });  
 *//нажатие на карту* **googleMap**.setOnMapClickListener(**new** GoogleMap.OnMapClickListener() {  
 @Override  
 **public void** onMapClick(LatLng latLng) {  
 *flag* = **true**;  
 MarkerOptions markerOptions = **new** MarkerOptions();*//создаем маркер* markerOptions.position(latLng);*//задаем позицию* markerOptions.title(MainActivity.*exlist*.get(**pos**).**objects**.get(obpos).**obname** + **" "** + latLng.**latitude** + **" : "** + latLng.**longitude**);*//задаем название* MainActivity.*exlist*.get(**pos**).**objects**.get(obpos).**oblatitude** = latLng.**latitude**;*//широта* MainActivity.*exlist*.get(**pos**).**objects**.get(obpos).**oblongitude** = latLng.**longitude**;*//долгота* **googleMap**.clear();*//очищаем позицию, которая была до* **googleMap**.animateCamera(CameraUpdateFactory.*newLatLng*(latLng));*//анимируем переход к новой метке* **googleMap**.addMarker(markerOptions);*//помещаем метку на карту* }  
 });  
 }  
  
 @Override  
 **public boolean** onCreateOptionsMenu(Menu menu) {  
 getMenuInflater().inflate(R.menu.***menu\_maps***, menu);  
 **return true**;  
 }  
  
 *//обработка нажатия на аппаратную кнопку назад* @Override  
 **public boolean** onKeyDown(**int** keyCode, KeyEvent event) {  
 **if** (keyCode == KeyEvent.***KEYCODE\_BACK***) {  
 *flag* = **false**;  
 MainActivity.*exlist*.get(**pos**).**objects**.get(obpos).**oblatitude** = 0.0;  
 MainActivity.*exlist*.get(**pos**).**objects**.get(obpos).**oblongitude** = 0.0;  
 finish();  
 finishActivity(5);  
 Intent intent = **new** Intent(**this**, TextActivity.**class**);  
 intent.putExtra(**"position"**, Integer.*toString*(**pos**));  
 intent.putExtra(**"obposition"**, Integer.*toString*(obpos));  
 startActivity(intent);  
 *// set the animation to move once the button is clicked* overridePendingTransition(R.anim.***slide\_in***, R.anim.***slide\_out***);  
 **return true**;  
 }  
 **return false**;  
 }  
}

# Класс MainActivity

**package** com.example.anast.app;  
  
**import** android.app.AlertDialog;  
**import** android.content.Context;  
**import** android.content.DialogInterface;  
**import** android.content.Intent;  
**import** android.content.SharedPreferences;  
**import** android.content.pm.ActivityInfo;  
**import** android.net.ConnectivityManager;  
**import** android.net.NetworkInfo;  
**import** android.os.Bundle;  
**import** android.support.v7.app.AppCompatActivity;  
**import** android.util.SparseArray;  
**import** android.view.KeyEvent;  
**import** android.view.LayoutInflater;  
**import** android.view.Menu;  
**import** android.view.MenuItem;  
**import** android.view.View;  
**import** android.view.ViewGroup;  
**import** android.widget.AdapterView;  
**import** android.widget.BaseAdapter;  
**import** android.widget.Button;  
**import** android.widget.CheckBox;  
**import** android.widget.EditText;  
**import** android.widget.ListAdapter;  
**import** android.widget.ListView;  
**import** android.widget.TextView;  
**import** android.widget.Toast;  
  
**import** com.dropbox.client2.DropboxAPI;  
**import** com.dropbox.client2.android.AndroidAuthSession;  
**import** com.dropbox.client2.session.AccessTokenPair;  
**import** com.dropbox.client2.session.AppKeyPair;  
**import** com.dropbox.client2.session.TokenPair;  
**import** com.example.anast.app.SwipeListView.SwipeListViewCallback;  
  
**import** java.util.ArrayList;  
**import** java.util.List;  
  
**public class** MainActivity **extends** AppCompatActivity **implements** SwipeListViewCallback, API\_Listener {  
  
 **private** EditText **propname**;*//имя нового объекта* **private** ListView **list**;*//объекты на экране* **private** Button **upload**;*//загрузка имеющихся объектов в dropbox* **private** MyAdapter **m\_Adapter**;*//адаптер списка всех объектов на экране, чтобы была возможность их удалить* **public static** List<Excursion> *exlist*;*//все экскурсии  
 //Переменные связанные с dropbox* **private** DropboxAPI<AndroidAuthSession> **mApi**;*//api dropbox extra* **private boolean mLoggedIn**;*//пользователь залогинился или нет* **private** DropboxAPI **dropboxApi**;*//api dropbox* **private boolean isUserLoggedIn**;*//пользователь залогинился или нет* @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 setRequestedOrientation(ActivityInfo.***SCREEN\_ORIENTATION\_LANDSCAPE***);*//горизонтальная ориентация экрана* **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_main***);  
 **boolean** q = Constants.*FIRST\_TIME\_FLAG*;  
 **if** (!q) {  
 **if** (*exlist* == **null**) {  
 *exlist* = **new** ArrayList<>();  
 }  
 Constants.*FIRST\_TIME\_FLAG* = **true**;  
 }  
 **propname** = (EditText) findViewById(R.id.***propname***);  
 **list** = (ListView) findViewById(R.id.***objectslist***);  
 **list**.setOnItemClickListener(**new** AdapterView.OnItemClickListener() {  
 @Override  
 **public void** onItemClick(AdapterView<?> a, View v, **int** position, **long** id) {  
 **for** (**int** i = 0; i < *exlist*.size(); i++) {  
 *exlist*.get(i).**selected** = **false**;  
 }  
 **int** index = 0;  
 **for** (**int** i = 0; i < *exlist*.size(); i++) {  
 **if** (*exlist*.get(i).**exname**.equals(*exlist*.get(position).**exname**)) {  
 index = i;  
 **break**;  
 }  
 }  
 Intent intent = **new** Intent(MainActivity.**this**, ObjectActivity.**class**);  
 intent.putExtra(**"mode"**, Integer.*toString*(position));  
 intent.putExtra(**"position"**, Integer.*toString*(index));  
 startActivityForResult(intent, 21);  
 *// set the animation to move once the button is clicked* overridePendingTransition(R.anim.***slide\_forw***, R.anim.***slide\_back***);  
 }  
 });  
 SwipeListView l = **new** SwipeListView(**this**, **this**);  
 l.exec();  
 **try** {  
 **if** (*exlist*.size() != 0) {  
 **m\_Adapter** = **new** MyAdapter();  
 **for** (**int** i = 0; i < *exlist*.size(); i++) {  
 **m\_Adapter**.addItem(*exlist*.get(i).**exname**);  
 }  
 **list**.setAdapter(**m\_Adapter**);  
 }  
 } **catch** (Exception e) {  
 e.printStackTrace();  
 }  
 Button del = (Button) findViewById(R.id.***delbut***);  
 del.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View view) {  
 Toast.*makeText*(getApplicationContext(), **"Tap on excursion you want to delete and then swipe from right to left"**, Toast.***LENGTH\_SHORT***).show();  
 }  
 });  
 Button new\_ex = (Button) findViewById(R.id.***in\_button***);  
 Button haa = (Button) findViewById(R.id.***haabut***);  
 new\_ex.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View view) {  
 **if** (!Conn()) {  
 Toast.*makeText*(getApplicationContext(), **"No internet connection!"**, Toast.***LENGTH\_SHORT***).show();  
 } **else** {  
 Excursion ex = **new** Excursion();  
 **if** (**propname**.getText().length() == 0)  
 Toast.*makeText*(getApplicationContext(), **"Enter the excursion's name"**, Toast.***LENGTH\_SHORT***).show();  
 **else** {  
 **boolean** flag = **false**;  
 ex.**exname** = **propname**.getText().toString();  
 **for** (**int** i = 0; i < *exlist*.size(); i++) {  
 **if** (*exlist*.get(i).**exname**.equals(ex.**exname**)) {  
 Toast.*makeText*(getApplicationContext(), **"You already have an excursion with this name"**, Toast.***LENGTH\_SHORT***).show();  
 flag = **true**;  
 **break**;  
 }  
 }  
 **if** (!flag) {  
 **for** (**int** i = 0; i < *exlist*.size(); i++) {  
 *exlist*.get(i).**selected** = **false**;  
 }  
 *exlist*.add(ex);  
 Intent intent = **new** Intent(MainActivity.**this**, ObjectActivity.**class**);  
 intent.putExtra(**"mode"**, **"-1"**);  
 intent.putExtra(**"position"**, Integer.*toString*(*exlist*.size() - 1));  
 startActivityForResult(intent, 21);  
 *// set the animation to move once the button is clicked* overridePendingTransition(R.anim.***slide\_forw***, R.anim.***slide\_back***);  
 }  
 }  
 }  
 }  
 });  
 haa.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View view) {  
 **if** (!Conn()) {  
 Toast.*makeText*(getApplicationContext(), **"No internet connection!"**, Toast.***LENGTH\_SHORT***).show();  
 } **else** {  
 **for** (**int** i = 0; i < *exlist*.size(); i++) {  
 *exlist*.get(i).**selected** = **false**;  
 }  
 Intent intent = **new** Intent(MainActivity.**this**, HAA.**class**);  
 startActivityForResult(intent, 10);  
 *// set the animation to move once the button is clicked* overridePendingTransition(R.anim.***slide\_forw***, R.anim.***slide\_back***);  
 }  
 }  
 });  
 Button logindropbox = (Button) findViewById(R.id.***conuploadbut***);  
 logindropbox.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View view) {  
 **if** (**isUserLoggedIn**) {  
 **dropboxApi**.getSession().unlink();  
 loggedIn(**false**);  
 Toast.*makeText*(MainActivity.**this**, **"You have just logged out from the Dropbox"**, Toast.***LENGTH\_LONG***).show();  
 } **else** {  
 ((AndroidAuthSession) **dropboxApi**.getSession()).startAuthentication(MainActivity.**this**);  
 }  
 }  
 });  
 **upload** = (Button) findViewById(R.id.***uploadbut***);  
 **upload**.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View view) {  
 **if** (!**mLoggedIn**)  
 Toast.*makeText*(MainActivity.**this**, **"This app wasn't authenticated properly. Please link with the Dropbox to upload your excursions"**, Toast.***LENGTH\_LONG***).show();  
 **if** (**mLoggedIn**) {  
 **if** (!Conn()) {  
 Toast.*makeText*(getApplicationContext(), **"No internet connection!"**, Toast.***LENGTH\_SHORT***).show();  
 } **else** {  
 **boolean** sel = **false**;  
 **boolean** size\_ob = **false**;  
 **for** (**int** i = 0; i < *exlist*.size(); i++) {  
 **if** (*exlist*.get(i).**selected**) {*//экскурсия выбрана для загрузки* sel = **true**;  
 **if** (*exlist*.get(i).**objects**.size() == 0)*//но объектов нет* {  
 size\_ob = **true**;  
 }  
 }  
 }  
 **if** (!sel)  
 Toast.*makeText*(MainActivity.**this**, **"You should choose at least one excursion to upload"**, Toast.***LENGTH\_LONG***).show();  
 **else** {  
 **if** (size\_ob) {  
 Toast.*makeText*(MainActivity.**this**, **"All the excursions you want to upload must have at least 1 object"**, Toast.***LENGTH\_LONG***).show();  
 } **else** {  
 Upload upload = **new** Upload(Constants.*UPLOAD\_CODE*, MainActivity.**this**, **dropboxApi**, **mApi**, DropboxData.***DROPBOX\_FILE\_DIR***);  
 upload.execute();  
 }  
 }  
 }  
 }  
 }  
 });  
 loggedIn(**false**);  
 AppKeyPair appKeyPair = **new** AppKeyPair(DropboxData.***ACCESS\_KEY***, DropboxData.***ACCESS\_SECRET***);  
 AndroidAuthSession session;  
 SharedPreferences prefs = getSharedPreferences(DropboxData.***DROPBOX\_NAME***, 0);  
 String key = prefs.getString(DropboxData.***ACCESS\_KEY***, **null**);  
 String secret = prefs.getString(DropboxData.***ACCESS\_SECRET***, **null**);  
 **if** (key != **null** && secret != **null**) {  
 AccessTokenPair token = **new** AccessTokenPair(key, secret);  
 session = **new** AndroidAuthSession(appKeyPair, DropboxData.***ACCESS\_TYPE***, token);  
 } **else** {  
 session = **new** AndroidAuthSession(appKeyPair, DropboxData.***ACCESS\_TYPE***);  
 }  
 **mApi** = **new** DropboxAPI<>(session);  
 **dropboxApi** = **new** DropboxAPI(session);  
 }  
  
 *//logging in to dropbox* **public void** loggedIn(**boolean** userLoggedIn) {  
 **mLoggedIn** = userLoggedIn;  
 **isUserLoggedIn** = userLoggedIn;  
 }  
  
 *//файлы успешно загружены* @Override  
 **public void** onSuccess(**int** requestnumber, Object obj) {  
 Toast.*makeText*(MainActivity.**this**, **"All excursions were successfully uploaded"**, Toast.***LENGTH\_LONG***).show();  
 }  
  
 *//в процессе загрузки были ошибки* @Override  
 **public void** onFail(String errormessage) {  
 Toast.makeText(MainActivity.**this**, **"Sorry, we have some problems:("**, Toast.LENGTH\_LONG).show();  
 }  
  
 *//при повторном запуске не требуется заново логиниться, т.к данные сохранены в shared prefs* @Override  
 **protected void** onResume() {  
 **super**.onResume();  
 AndroidAuthSession session = (AndroidAuthSession) dropboxApi.getSession();  
 **if** (session.authenticationSuccessful()) {  
 **try** {  
 session.finishAuthentication();  
 TokenPair tokens = session.getAccessTokenPair();  
 SharedPreferences prefs = getSharedPreferences(DropboxData.DROPBOX\_NAME, 0);  
 SharedPreferences.Editor editor = prefs.edit();  
 editor.putString(DropboxData.ACCESS\_KEY, tokens.key);  
 editor.putString(DropboxData.ACCESS\_SECRET, tokens.secret);  
 editor.commit();  
 loggedIn(**true**);  
 } **catch** (IllegalStateException e) {  
 Toast.makeText(**this**, **"Error during Dropbox auth"**, Toast.LENGTH\_SHORT).show();  
 }  
 }  
 }  
  
 *//проверка наличия соединения с интернетом* **public boolean** Conn() {  
 *//Создаем пример класса connection detector:* ConnectionDetector cd = **new** ConnectionDetector(getApplicationContext());  
 *//Получаем статус Интернет* Boolean isInternetPresent = cd.ConnectingToInternet();  
 *//Проверяем Интернет статус:* **if** (isInternetPresent) {  
 **return true**;  
 } **else** {  
 *//Интернет соединения нет просим пользователя подключить Интернет:* Toast.makeText(getApplicationContext(), **"No Internet connection!"**, Toast.LENGTH\_SHORT).show();  
 **return false**;  
 }  
 }  
  
 @Override  
 **public boolean** onCreateOptionsMenu(Menu menu) {  
 getMenuInflater().inflate(R.menu.menu\_main, menu);  
 **return true**;  
 }  
  
 @Override  
 **public boolean** onOptionsItemSelected(MenuItem item) {  
 **int** id = item.getItemId();  
 **if** (id == R.id.action\_settings) {  
 **return true**;  
 }  
 **return super**.onOptionsItemSelected(item);  
 }  
  
 @Override  
 **public** ListView getListView() {  
 **return** list;  
 }  
  
 *//свайп по элементу списка объектов* @Override  
 **public void** onSwipeItem(**boolean** isRight, **int** position) {  
 m\_Adapter.onSwipeItem(isRight, position);  
 }  
  
 *//нажатие на элемент списка объектов* @Override  
 **public void** onItemClickListener(ListAdapter adapter, **int** position) {  
 }  
  
 *//класс, реализующий удаление элемента из списка по свайпу* **public class** MyAdapter **extends** BaseAdapter {  
  
 **private final int** invalid = -1;  
 **protected int** delete\_pos = -1;  
  
 **protected** List<String> m\_List;  
  
 **public** MyAdapter() {  
 m\_List = **new** ArrayList<>();  
 }  
  
 **public void** addItem(String item) {  
 m\_List.add(item);  
 notifyDataSetChanged();  
 }  
  
 **public void** onSwipeItem(**boolean** isRight, **int** position) {  
 **if** (!isRight) {  
 delete\_pos = position;  
 } **else if** (delete\_pos == position) {  
 delete\_pos = invalid;  
 }  
 notifyDataSetChanged();  
 }  
  
 **public void** deleteItem(**int** pos) {  
 m\_List.remove(pos);  
 exlist.remove(pos);  
 delete\_pos = invalid;  
 notifyDataSetChanged();  
 }  
  
 @Override  
 **public int** getCount() {  
 **return** m\_List.size();  
 }  
  
 @Override  
 **public** String getItem(**int** position) {  
 **return** m\_List.get(position);  
 }  
  
 @Override  
 **public long** getItemId(**int** position) {  
 **return** 0;  
 }  
  
 @Override  
 **public** View getView(**final int** position, View convertView, ViewGroup parent) {  
 **if** (convertView == **null**) {  
 convertView = LayoutInflater.from(MainActivity.**this**).inflate(R.layout.item, **null**);  
 }  
 TextView text = ViewHolderPattern.get(convertView, R.id.text);  
 Button delete = ViewHolderPattern.get(convertView, R.id.delete);  
 CheckBox check = ViewHolderPattern.get(convertView, R.id.checkBox11);  
 **if** (delete\_pos == position) {  
 delete.setVisibility(View.VISIBLE);  
 } **else** delete.setVisibility(View.GONE);  
 delete.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 deleteItem(position);  
 }  
 });  
 check.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 exlist.get(position).selected = !exlist.get(position).selected;  
 }  
 });  
 text.setText(getItem(position));  
 **return** convertView;  
 }  
 }  
  
 *//класс, свзанный с удалением элемента из списка из определенного holdera* **public static class** ViewHolderPattern {  
 @SuppressWarnings(**"unchecked"**)  
 **public static** <T **extends** View> T get(View view, **int** id) {  
 SparseArray<View> viewHolder = (SparseArray<View>) view.getTag();  
 **if** (viewHolder == **null**) {  
 viewHolder = **new** SparseArray<>();  
 view.setTag(viewHolder);  
 }  
 View childView = viewHolder.get(id);  
 **if** (childView == **null**) {  
 childView = view.findViewById(id);  
 viewHolder.put(id, childView);  
 }  
 **return** (T) childView;  
 }  
 }  
  
 *//анализатор нажатия на аппаратную кнопку назад* @Override  
 **public boolean** onKeyDown(**int** keyCode, KeyEvent event) {  
 **if** (keyCode == KeyEvent.KEYCODE\_BACK) {  
 **for** (**int** i = 0; i < exlist.size(); i++)  
 exlist.get(i).selected = **false**;  
 AlertDialog.Builder alertDialogBuilder = **new** AlertDialog.Builder(MainActivity.**this**);  
 alertDialogBuilder.setTitle(**"Are you sure you want to exit the ExcurionMaker?"**);  
 alertDialogBuilder.setMessage(**"We highly recommend to upload all your excursions to DropBox. Otherwise you will lost all your data"**);  
 alertDialogBuilder.setPositiveButton(**"Yes, I'm sure"**, **new** DialogInterface.OnClickListener() {*// set positive button: Yes message* **public void** onClick(DialogInterface dialog, **int** id) {  
 finishAffinity();  
 }  
 });  
 alertDialogBuilder.setNegativeButton(**"No, let me save my data"**, **new** DialogInterface.OnClickListener() {*// set negative button: No message* **public void** onClick(DialogInterface dialog, **int** id) {  
 dialog.cancel();  
 }  
 });  
 AlertDialog alertDialog = alertDialogBuilder.create();  
 alertDialog.show();  
 **return true**;  
 }  
 **return false**;  
 }  
}

Класс ConnectionDetector  
  
*//класс связанный с наличие подключения к интернету***class** ConnectionDetector {  
 **private** Context **\_context**;  
  
 **public** ConnectionDetector(Context context)  
 {  
 **this**.**\_context** = context;  
 }  
  
 *//подключаемся к интернету, проверяем соединение* **public boolean** ConnectingToInternet(){  
 ConnectivityManager connectivity = (ConnectivityManager) **\_context**.getSystemService(Context.***CONNECTIVITY\_SERVICE***);  
 **if** (connectivity != **null**)  
 {  
 NetworkInfo[] info = connectivity.getAllNetworkInfo();  
 **if** (info != **null**)  
 **for** (**int** i = 0; i < info.**length**; i++)  
 **if** (info[i].getState() == NetworkInfo.State.***CONNECTED***)  
 **return true**;  
 }  
 **return false**;  
 }  
}

# Класс LoadActivity

**package** com.example.anast.app;  
  
**import** android.content.ContentUris;  
**import** android.content.Context;  
**import** android.content.Intent;  
**import** android.content.pm.ActivityInfo;  
**import** android.database.Cursor;  
**import** android.net.Uri;  
**import** android.os.AsyncTask;  
**import** android.os.Bundle;  
**import** android.os.PowerManager;  
**import** android.provider.MediaStore;  
**import** android.support.v7.app.AppCompatActivity;  
**import** android.util.Log;  
**import** android.view.KeyEvent;  
**import** android.view.Menu;  
**import** android.view.MenuItem;  
**import** android.widget.ProgressBar;  
**import** android.widget.TextView;  
  
**import** com.google.android.gms.ads.AdRequest;  
**import** com.google.android.gms.ads.AdView;  
  
**import** java.util.ArrayList;  
**import** java.util.List;  
**import** java.util.concurrent.TimeUnit;  
  
*//класс, отвечающий за загрузку медиафайлов в приложение***public class** LoadActivity **extends** AppCompatActivity {  
  
 **public static** List<DataClass> *imagesList* = **new** ArrayList<>();*//список всех фото* **public static** List<DataClass> *videosList* = **new** ArrayList<>();*//список всех видео* **public static** List<DataClass> *audiosList* = **new** ArrayList<>();*//список всех аудио* **public static** List<String> *filepaths* = **new** ArrayList<>();*//все фото-пути* **public static** List<PhotoActivity.PhotoNum> *checkboxes* = **new** ArrayList<>();*//номера выбранных чекбоксов у фото* **private** Cursor **cursor1**;*//бегунок по списку на карте памяти* **private** TextView **info**;*//информация о загрузке* **private** ProgressBar **horizontalprogress**;*//прогресс загрузки(заполняющийся столбец)* **public** Context **context**;*//нужен для подгрузки аудио* **protected** PowerManager.WakeLock **mWakeLock**;*//экран всегда включен пока идет загрузка* @Override  
 **public void** onDestroy() {  
 **this**.**mWakeLock**.release();  
 **super**.onDestroy();  
 }  
  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 setRequestedOrientation(ActivityInfo.***SCREEN\_ORIENTATION\_LANDSCAPE***);*//горизонтальная ориентация экрана* **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_load***);  
 **final** PowerManager pm = (PowerManager) getSystemService(Context.***POWER\_SERVICE***);  
 **this**.**mWakeLock** = pm.newWakeLock(PowerManager.***SCREEN\_DIM\_WAKE\_LOCK***, **"My Tag"**);  
 **this**.**mWakeLock**.acquire();  
 AdView mAdView = (AdView) findViewById(R.id.***adView***);*//создание рекламы* **context** = **this**;  
 */\*AdRequest adRequest = new AdRequest.Builder().build();//****TODO: CHANGE to normal*** *mAdView.loadAd(adRequest);\*/* AdRequest adRequest = **new** com.google.android.gms.ads.AdRequest.Builder()  
 .addTestDevice(AdRequest.***DEVICE\_ID\_EMULATOR***)  
 .addTestDevice(**"96F89B24B609ED5DFE69922744B289BF"**).build();*//запуск рекламы на утройстве* mAdView.loadAd(adRequest);*//подгрузка рекламы* **info** = (TextView) findViewById(R.id.***progressView***);*//информация о загрузке* **horizontalprogress** = (ProgressBar) findViewById(R.id.***progressload***);*//прогресс загрузки(заполняющийся столбец)* **horizontalprogress**.setMax(100);*//мксимальный процент загрузки* LoadTask load = **new** LoadTask();  
 load.execute();*//запуск подгрузки* }  
  
 @Override  
 **public boolean** onCreateOptionsMenu(Menu menu) {  
 getMenuInflater().inflate(R.menu.***menu\_load***, menu);  
 **return true**;  
 }  
  
 @Override  
 **public boolean** onOptionsItemSelected(MenuItem item) {  
 **int** id = item.getItemId();  
 **if** (id == R.id.***action\_settings***) {  
 **return true**;  
 }  
 **return super**.onOptionsItemSelected(item);  
 }  
  
  
 **class** LoadTask **extends** AsyncTask<Void, String, Void> {  
  
 **boolean im** = **true**;*//были загружены фото* **boolean vid** = **true**;*//были загружены видео* **boolean aud** = **true**;*//были загуржены аудио  
  
 //подготовка к загрузке* @Override  
 **protected void** onPreExecute() {  
 **super**.onPreExecute();  
 **info**.setText(**"Loading...Please wait, you have a lot of media files"**);  
 }  
  
 *//загрузка всех необходимых файлов* @Override  
 **protected** Void doInBackground(Void... params) {  
 **try** {  
 **if** (*imagesList*.size() == 0) {  
 **try** {  
 PhotoFilePaths();*//загружаем фото пути* PhotoInic();*//загружаем сами фото* Log.*d*(**"LoadActivity"**, **"loaded images"**);  
 } **catch** (Exception e) {  
 e.printStackTrace();  
 }  
 }  
 **if** (*videosList*.size() == 0) {  
 **try** {  
 VideoInic();*//загружаем видео* Log.*d*(**"LoadActivity"**, **"loaded videos"**);  
 } **catch** (Exception e) {  
 e.printStackTrace();  
 }  
 }  
 **if** (*audiosList*.size() == 0) {  
 **try** {  
 AudioInic();*//загружаем аудио* Log.*d*(**"LoadActivity"**, **"loaded audios"**);  
 } **catch** (Exception e) {  
 e.printStackTrace();  
 }  
 }  
 **if** (isCancelled())*//что-то пошло не так* **return null**;  
 TimeUnit.***SECONDS***.sleep(1);*// разъединяемся* } **catch** (Exception e) {  
 e.printStackTrace();  
 }  
 **return null**;  
 }  
  
 *//загрузка фото путей* **void** PhotoFilePaths() {  
 **if** (*filepaths*.size() == 0) {  
 String[] mediaColumns = {MediaStore.Images.Media.***\_ID***, MediaStore.Images.Media.***DATA***, MediaStore.Images.Media.***TITLE***, MediaStore.Images.Media.***MIME\_TYPE***};  
 **final** String orderBy = MediaStore.Images.Media.***\_ID***;  
 **cursor1** = managedQuery(MediaStore.Images.Media.***EXTERNAL\_CONTENT\_URI***, mediaColumns, **null**, **null**, orderBy);  
 **if** (**cursor1**.moveToFirst()) {  
 **do** {  
 *filepaths*.add(**cursor1**.getString(**cursor1**.getColumnIndexOrThrow(MediaStore.Images.Media.***DATA***)));  
 } **while** (**cursor1**.moveToNext());  
 }  
 }  
 }  
  
 *//загрузка непосредственно самих фото* **void** PhotoInic() {  
 **if** (*imagesList*.size() == 0) {  
 **final** String[] columns = {MediaStore.Images.Thumbnails.***\_ID***};  
 **final** String orderBy = MediaStore.Images.Media.***\_ID***;  
 Cursor imagecursor = managedQuery(MediaStore.Images.Media.***EXTERNAL\_CONTENT\_URI***, columns, **null**, **null**, orderBy);  
 **if** (imagecursor != **null**) {  
 **int** image\_column\_index = imagecursor.getColumnIndex(MediaStore.Images.Media.***\_ID***);  
 **int** count = imagecursor.getCount();  
 **for** (**int** i = 0; i < count; i++) {  
 imagecursor.moveToPosition(i);  
 **int** id = imagecursor.getInt(image\_column\_index);  
 DataClass imageItem = **new** DataClass();  
 imageItem.**id** = id;  
 imageItem.**filePath** = *filepaths*.get(i);  
 PhotoMakeCheck(imageItem, i);  
 *imagesList*.add(imageItem);  
 Log.*d*(**"LoadActivity"**, **"images added "** + i);  
 publishProgress(**"Images loaded "** + i + **"/"** + count);  
 }  
 imagecursor.close();  
 **im** = **false**;  
 }  
 }  
 }  
  
 *//загурзка видео* **void** VideoInic() {  
 **if** (*videosList*.size() == 0) {  
 String[] thumbColumns = {MediaStore.Video.Thumbnails.***DATA***, MediaStore.Video.Thumbnails.***VIDEO\_ID***};  
 String[] mediaColumns = {MediaStore.Video.Media.***\_ID***,  
 MediaStore.Video.Media.***DATA***, MediaStore.Video.Media.***TITLE***,  
 MediaStore.Video.Media.***MIME\_TYPE***};  
 **final** String orderBy = MediaStore.Video.Media.***\_ID***;  
 **cursor1** = managedQuery(MediaStore.Video.Media.***EXTERNAL\_CONTENT\_URI***, mediaColumns, **null**, **null**, orderBy);  
 **int** z = 0;  
 **int** count = **cursor1**.getCount();  
 **if** (**cursor1**.moveToFirst()) {  
 **do** {  
 DataClass newVVI = **new** DataClass();  
 **int** id = **cursor1**.getInt(**cursor1**.getColumnIndex(MediaStore.Video.Media.***\_ID***));  
 Cursor thumbCursor = managedQuery(MediaStore.Video.Thumbnails.***EXTERNAL\_CONTENT\_URI***, thumbColumns, MediaStore.Video.Thumbnails.***VIDEO\_ID*** + **"="** + id, **null**, **null**);  
 **if** (thumbCursor.moveToFirst()) {  
 newVVI.**thumbPath** = thumbCursor.getString(thumbCursor.getColumnIndex(MediaStore.Video.Thumbnails.***DATA***));  
 }  
 newVVI.**filePath** = **cursor1**.getString(**cursor1**.getColumnIndexOrThrow(MediaStore.Video.Media.***DATA***));  
 newVVI.**title** = **cursor1**.getString(**cursor1**.getColumnIndexOrThrow(MediaStore.Video.Media.***TITLE***));  
 newVVI.**mimeType** = **cursor1**.getString(**cursor1**.getColumnIndexOrThrow(MediaStore.Video.Media.***MIME\_TYPE***));  
 newVVI.**id** = id;  
 *videosList*.add(newVVI);  
 publishProgress(**"Videos loaded "** + z + **"/"** + count);  
 Log.*d*(**"LoadActivity"**, **"videos added"**);  
 z++;  
 } **while** (**cursor1**.moveToNext());  
 **vid** = **false**;  
 }  
 }  
 }  
  
 *//загрузка аудио* **void** AudioInic() {  
 **final** Uri uri = MediaStore.Audio.Media.***EXTERNAL\_CONTENT\_URI***;  
 **final** String[] cursor\_cols = {MediaStore.Audio.Media.***\_ID***, MediaStore.Audio.Media.***TITLE***, MediaStore.Audio.Media.***DATA***,  
 MediaStore.Audio.Media.***ALBUM\_ID***};  
 **final** String where = MediaStore.Audio.Media.***IS\_MUSIC*** + **"=1"**;  
 **final** Cursor cursor = **context**.getContentResolver().query(uri, cursor\_cols, where, **null**, **null**);  
 **if** (cursor != **null**) {  
 **if** (cursor.getCount() == 0)  
 **return**;  
 }  
 **int** z = 1;  
 **int** count = cursor.getCount();  
 **while** (cursor.moveToNext()) {  
 String track = cursor.getString(cursor.getColumnIndexOrThrow(MediaStore.Audio.Media.***TITLE***));  
 String data = cursor.getString(cursor.getColumnIndexOrThrow(MediaStore.Audio.Media.***DATA***));  
 Long albumId = cursor.getLong(cursor.getColumnIndexOrThrow(MediaStore.Audio.Media.***ALBUM\_ID***));  
 Uri sArtworkUri = Uri.*parse*(**"content://media/external/audio/albumart"**);  
 Uri albumArtUri = ContentUris.*withAppendedId*(sArtworkUri, albumId);  
 DataClass audioListModel = **new** DataClass();  
 audioListModel.**coverPath** = albumArtUri;  
 audioListModel.**title** = track;  
 audioListModel.**filePath** = data;  
 *audiosList*.add(audioListModel);  
 publishProgress(**"Audios loaded "** + z++ + **"/"** + count);  
 Log.*d*(**"LoadActivity"**, **"music added"**);  
 }  
 **aud** = **false**;  
 }  
  
 *//делаем фото отмеченными, если такие есть* **public void** PhotoMakeCheck(DataClass imageItem, **int** i) {  
 **for** (**int** q = 0; q < *checkboxes*.size(); q++)  
 **if** (*checkboxes*.get(q).**num** == i)  
 imageItem.setSelected(**true**);  
 }  
  
 *//обновляем процесс загрузки* @Override  
 **protected void** onProgressUpdate(String... values) {  
 **super**.onProgressUpdate(values);  
 **try** {  
 **info**.setText(**""** + values[0]);  
 } **catch** (Exception e) {  
 e.printStackTrace();  
 }  
 **if** (!**im**) {  
 **horizontalprogress**.setProgress(33);  
 **im** = **true**;  
 } **else if** (!**vid**) {  
 **horizontalprogress**.setProgress(66);  
 **vid** = **true**;  
 } **else if** (!**aud**) {  
 **horizontalprogress**.setProgress(100);  
 **aud** = **true**;  
 }  
 }  
  
 *//заверщаем загрузку* @Override  
 **protected void** onPostExecute(Void result) {  
 **super**.onPostExecute(result);  
 **info**.setText(**"Finished"**);  
 **horizontalprogress**.setProgress(0);  
 Intent intent = **new** Intent(LoadActivity.**this**, MainActivity.**class**);  
 startActivity(intent);  
 }  
  
 *//что-то пошло не так* @Override  
 **protected void** onCancelled() {  
 **super**.onCancelled();  
 **info**.setText(**"Error!!!"**);  
 **horizontalprogress**.setProgress(0);  
 }  
 }  
  
 *//анализатор нажатия на аппаратную кнопку назад* @Override  
 **public boolean** onKeyDown(**int** keyCode, KeyEvent event) {  
 **if** (keyCode == KeyEvent.***KEYCODE\_BACK***) {  
 finishAffinity();  
 **return true**;  
 }  
 **return false**;  
 }  
}

Класс DataClass  
  
**package** com.example.anast.app;  
  
**import** android.net.Uri;  
  
*//класс, хранящий в себе информацию о всех мультимедиа объектах***class** DataClass {  
 **public** String **filePath**;*//путь к файлу* **public** Uri **coverPath**;*//путь к обложке* **public int id**;*//ай ди файла* **public** String **mimeType**;*//mime type* **public** String **thumbPath**;*//путь к миниатюре* **public** String **title**;*//название* **private boolean selected**;*//выбран ли файл* **public** DataClass() {  
 **super**();  
 **this**.**selected** = **false**;  
 }  
  
 **public boolean** isSelected() {  
 **return selected**;  
 }  
  
 **public void** setSelected(**boolean** selected) {  
 **this**.**selected** = selected;  
 }  
}

# Класс HAA

**package** com.example.anast.app;  
  
**import** android.content.Intent;  
**import** android.content.pm.ActivityInfo;  
**import** android.net.Uri;  
**import** android.os.Bundle;  
**import** android.support.v7.app.AppCompatActivity;  
**import** android.text.Html;  
**import** android.util.DisplayMetrics;  
**import** android.view.KeyEvent;  
**import** android.view.Menu;  
**import** android.view.MenuItem;  
**import** android.view.View;  
**import** android.widget.Button;  
**import** android.widget.TextView;  
**import** android.widget.Toast;  
  
**public class** HAA **extends** AppCompatActivity {  
  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 setRequestedOrientation(ActivityInfo.***SCREEN\_ORIENTATION\_LANDSCAPE***);  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_ha***);  
 TextView text = (TextView) findViewById(R.id.***connection***);  
 TextView text1 = (TextView) findViewById(R.id.***newob***);  
 TextView text2 = (TextView) findViewById(R.id.***delob***);  
 TextView text3 = (TextView) findViewById(R.id.***droptext***);  
 TextView text4 = (TextView) findViewById(R.id.***sharetext***);  
 TextView text5 = (TextView) findViewById(R.id.***tick***);  
 TextView text6 = (TextView) findViewById(R.id.***obinfo***);  
 TextView text7 = (TextView) findViewById(R.id.***feedbacktext***);  
 TextView text8 = (TextView) findViewById(R.id.***googleplay***);  
 String str, str1, str2, str3, str4, str5, str6, str7, str8;  
 *// Узнаем размеры экрана из ресурсов* DisplayMetrics displaymetrics = getResources().getDisplayMetrics();  
 **if** (displaymetrics.**widthPixels** >= 2000)*//планшеты* {  
 str = **"<font color='#1b9900'>Please don't turn off the Internet connection the app won't work without the Internet</font>"**;  
 str1 = **"<font color='#1b9900'>To create a new object/excursion click on plus button</font>"**;  
 str2 = **"<font color='#1b9900'>To delete object/excursion click on delete button</font>"**;  
 str3 = **"<font color='#1b9900'>To login/logout to the Dropbox click on the Dropbox button</font>"**;  
 str4 = **"<font color='#1b9900'>To upload the data to the Dropbox click on share button</font>"**;  
 str5 = **"<font color='#1b9900'>Press the tick if you want to download the excursion to the Dropbox</font>"**;  
 str6 = **"<font color='#1b9900'>To find out all the object's info click on the object twice</font>"**;  
 str7 = **"<font color='#1b9900'>To send the letter to the developer press on the envelope on the left</font>"**;  
 str8 = **"<font color='#1b9900'>To rate the app in PlayMarket press on its bage on the left</font>"**;  
 } **else if** (displaymetrics.**widthPixels** >= 720)*//Galazy Nexus и больше* {  
 str = **"<font color='#1b9900'>Please don't turn off the Internet connection<br/>the app won't work without the Internet</font>"**;  
 str1 = **"<font color='#1b9900'>To create a new object/excursion click on plus button</font>"**;  
 str2 = **"<font color='#1b9900'>To delete object/excursion click on delete button</font>"**;  
 str3 = **"<font color='#1b9900'>To login/logout to the Dropbox<br/> click on the Dropbox button</font>"**;  
 str4 = **"<font color='#1b9900'>To upload the data to the Dropbox <br/>click on share button</font>"**;  
 str5 = **"<font color='#1b9900'>Press the tick if you want to download<br/> the excursion to the Dropbox</font>"**;  
 str6 = **"<font color='#1b9900'>To find out all the object's info<br/> click on the object twice</font>"**;  
 str7 = **"<font color='#1b9900'>To send the letter to the developer press on<br/> the envelope on the left</font>"**;  
 str8 = **"<font color='#1b9900'>To rate the app in PlayMarket press <br/>on its bage on the left</font>"**;  
 } **else** {  
 str = **"<font color='#1b9900'>Please don't turn off the <br/>Internet connection the app <br/>won't work without the Internet</font>"**;  
 str1 = **"<font color='#1b9900'>To create a new object/excursion click <br/>on plus button</font>"**;  
 str2 = **"<font color='#1b9900'>To delete object/excursion <br/>click on delete button</font>"**;  
 str3 = **"<font color='#1b9900'>To login/logout to the Dropbox<br/> click on the Dropbox button</font>"**;  
 str4 = **"<font color='#1b9900'>To upload the data to the Dropbox <br/>click on share button</font>"**;  
 str5 = **"<font color='#1b9900'>Press the tick if you<br/> want to download<br/> the excursion to the Dropbox</font>"**;  
 str6 = **"<font color='#1b9900'>To find out all the object's <br/>info click on the object twice</font>"**;  
 str7 = **"<font color='#1b9900'>To send the letter to the<br/> developer press on<br/> the envelope on the left</font>"**;  
 str8 = **"<font color='#1b9900'>To rate the app<br/> in PlayMarket press <br/>on its bage on the left</font>"**;  
 }  
 text.setText(Html.*fromHtml*(str), TextView.BufferType.***SPANNABLE***);  
 text1.setText(Html.*fromHtml*(str1), TextView.BufferType.***SPANNABLE***);  
 text2.setText(Html.*fromHtml*(str2), TextView.BufferType.***SPANNABLE***);  
 text3.setText(Html.*fromHtml*(str3), TextView.BufferType.***SPANNABLE***);  
 text4.setText(Html.*fromHtml*(str4), TextView.BufferType.***SPANNABLE***);  
 text5.setText(Html.*fromHtml*(str5), TextView.BufferType.***SPANNABLE***);  
 text6.setText(Html.*fromHtml*(str6), TextView.BufferType.***SPANNABLE***);  
 text7.setText(Html.*fromHtml*(str7), TextView.BufferType.***SPANNABLE***);  
 text8.setText(Html.*fromHtml*(str8), TextView.BufferType.***SPANNABLE***);  
 Button feedback = (Button) findViewById(R.id.***feedback***);  
 feedback.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View view) {  
 Intent intent = **new** Intent(Intent.***ACTION\_SENDTO***);  
 intent.setType(**"text/plain"**);  
 intent.putExtra(Intent.***EXTRA\_SUBJECT***, **"ExcursionMaker"**);  
 intent.putExtra(Intent.***EXTRA\_TEXT***, **""**);  
 intent.setData(Uri.*parse*(**"mailto:prostorepa@yandex.ru"**));  
 intent.addFlags(Intent.***FLAG\_ACTIVITY\_NEW\_TASK***);  
 startActivity(intent);  
 }  
 });  
 Button rate = (Button) findViewById(R.id.***ratebut***);  
 rate.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View view) {  
 Toast.*makeText*(getApplicationContext(), **"The link to the app's page in Google Play will be added soon"**, Toast.***LENGTH\_SHORT***).show();  
 *//Try Google play  
 /\* Intent intent = new Intent(Intent.ACTION\_VIEW);  
 intent.setData(Uri.parse("market://details?id=com.cubeactive.qnotelistfree"));  
 if (!MyStartActivity(intent)) {  
 //Market (Google play) app seems not installed, let's try to open a webbrowser  
 intent.setData(Uri.parse("https://play.google.com/store/apps/details?id=com.cubeactive.qnotelistfree"));  
 if (!MyStartActivity(intent)) {  
 //Well if this also fails, we have run out of options, inform the user.  
 Toast.makeText(getApplicationContext(), "Could not open Android market, please install the market app.", Toast.LENGTH\_SHORT).show();  
 }  
 }\*/  
 //****TODO: add link to google play*** }  
 });  
 Button back = (Button) findViewById(R.id.***backhaa***);  
 back.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View view) {  
 finish();  
 finishActivity(10);  
 Intent intent = **new** Intent(HAA.**this**, MainActivity.**class**);  
 startActivity(intent);  
 *// set the animation to move once the button is clicked* overridePendingTransition(R.anim.***slide\_in***, R.anim.***slide\_out***);  
 }  
 });  
 }  
  
  
 @Override  
 **public boolean** onCreateOptionsMenu(Menu menu) {  
 getMenuInflater().inflate(R.menu.***menu\_ha***, menu);  
 **return true**;  
 }  
  
 @Override  
 **public boolean** onOptionsItemSelected(MenuItem item) {  
 **int** id = item.getItemId();  
 **if** (id == R.id.***action\_settings***) {  
 **return true**;  
 }  
 **return super**.onOptionsItemSelected(item);  
 }  
  
 *//обработка нажатия на аппаратную кнопку назад* @Override  
 **public boolean** onKeyDown(**int** keyCode, KeyEvent event) {  
 **if** (keyCode == KeyEvent.***KEYCODE\_BACK***) {  
 finish();  
 finishActivity(10);  
 Intent intent = **new** Intent(**this**, MainActivity.**class**);  
 startActivity(intent);  
 *// set the animation to move once the button is clicked* overridePendingTransition(R.anim.***slide\_in***, R.anim.***slide\_out***);  
 **return true**;  
 }  
 **return false**;  
 }  
}

# Класс Constants

**package** com.example.anast.app;  
  
**public class** Constants {  
 **public static int** *UPLOAD\_CODE* = 3000;*//код для загрузки в DropBox* **public static boolean** *FIRST\_TIME\_FLAG* = **false**;*//проверка захода в приложение первый раз, чтобы подгружать список имеющихся экскурсий* **public static int** *MILLISEC* = 1000;*//переменная для рассчета нормального времени на таймере* **public static int** *MINUTE* = 60;*//переменная для рассчета нормального времени на таймере* **public static int** *PROC* = 100;*//переменная для рассчета нормального времени на таймере*}

# Класс Excursion

**package** com.example.anast.app;  
  
**import** java.util.ArrayList;  
  
*//Класс - экскурсия***class** Excursion {  
 **public** Excursion() {  
 **objects** = **new** ArrayList<>();  
 **exname** = **""**;  
 **selected** = **false**;  
 }  
  
 **public boolean selected**;*//выбран ли пользователем объект экскурсии для дальнейшего сохранения* **public** String **exname**;*//запланированное имя экскурсии* **public** ArrayList<ExcursionObject> **objects**;*//объекты конкретной экскурсии*}

# Класс ExcursionObject

**package** com.example.anast.app;  
  
**import** java.util.ArrayList;  
  
*//Класс - объект экскурсии***class** ExcursionObject {  
 **public** ArrayList<DataClass> **obselectedphotos**;*//конкретные фото отдельного объекта* **public** ArrayList<DataClass> **obselectedvideos**;*//конкретные видео отдельного объекта* **public** ArrayList<DataClass> **obselectedaudios**;*//конкретные аудио отдельного объекта* **public** String **obobjectText**;*//конкретный текст отдельного объекта* **public** Double **oblatitude**;*//широта объекта* **public** Double **oblongitude**;*//долгота объекта* **public** String **obname**;*//запланированное имя объекта* **public** ExcursionObject() {  
 **obselectedphotos** = **new** ArrayList<>();  
 **obselectedvideos** = **new** ArrayList<>();  
 **obselectedaudios** = **new** ArrayList<>();  
 **obname** = **""**;  
 **obobjectText** = **""**;  
 **oblatitude** = 0.0;  
 **oblongitude** = 0.0;  
 }  
}

# Интерфейс DropboxData

**package** com.example.anast.app;  
  
**import** com.dropbox.client2.session.Session;  
  
*//данные дропбокса***public interface** DropboxData {  
 String ***DROPBOX\_FILE\_DIR*** = **""**;*// /ExcursionMaker/* String ***DROPBOX\_NAME*** = **"dropbox\_prefs"**;  
 String ***ACCESS\_KEY*** = **"ejnv6emaxutkxkb"**;  
 String ***ACCESS\_SECRET*** = **"2tpqqwbnrptt8qn"**;  
 Session.AccessType ***ACCESS\_TYPE*** = Session.AccessType.***DROPBOX***;  
}

# Класс AudioActivity

**package** com.example.anast.app;  
  
**import** android.content.Context;  
**import** android.content.Intent;  
**import** android.content.pm.ActivityInfo;  
**import** android.graphics.Bitmap;  
**import** android.graphics.BitmapFactory;  
**import** android.media.MediaPlayer;  
**import** android.os.Bundle;  
**import** android.os.Handler;  
**import** android.provider.MediaStore;  
**import** android.support.v7.app.AppCompatActivity;  
**import** android.view.KeyEvent;  
**import** android.view.LayoutInflater;  
**import** android.view.Menu;  
**import** android.view.MenuItem;  
**import** android.view.View;  
**import** android.view.ViewGroup;  
**import** android.widget.BaseAdapter;  
**import** android.widget.Button;  
**import** android.widget.CheckBox;  
**import** android.widget.ImageButton;  
**import** android.widget.ImageView;  
**import** android.widget.ListView;  
**import** android.widget.SeekBar;  
**import** android.widget.TextView;  
**import** android.widget.Toast;  
  
**import** java.io.FileNotFoundException;  
**import** java.io.IOException;  
**import** java.util.ArrayList;  
**import** java.util.List;  
  
**public class** AudioActivity **extends** AppCompatActivity **implements** MediaPlayer.OnCompletionListener, SeekBar.OnSeekBarChangeListener {  
  
 **private** ImageButton **btnPlay**;*//кнопка плей* **private** TextView **SongsName**;*//имя трека* **private** ImageView **Cover**;*//art album* **private** MediaPlayer **mp**;*//аудиоплеер* **private** Handler **mHandler** = **new** Handler();*//работа с аудио на плеере* **private** Utilities **utils**;*//утилиты* **private int currentSongIndex**;*//номер текщей песни* **public static** List<AudioNum> *checkboxes* = **new** ArrayList<>();*//номера выбранных чекбоксов у видео* **public static** List<ViewHolder> *viewholders* = **new** ArrayList<>();*//список всех видео-хранилищ* **private int obpos**, **pos**;*//коды экскурсии и объекта* @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 setRequestedOrientation(ActivityInfo.***SCREEN\_ORIENTATION\_LANDSCAPE***);  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_audio***);  
 Toast.*makeText*(getApplicationContext(), **"Choose audio files now"**, Toast.***LENGTH\_LONG***).show();  
 **try** {  
 **pos** = Integer.*parseInt*(getIntent().getExtras().getString(**"position"**));  
 } **catch** (Exception e) {  
 **pos** = 0;  
 }  
 **try** {  
 **obpos** = Integer.*parseInt*(getIntent().getExtras().getString(**"obposition"**));  
 } **catch** (Exception e) {  
 **obpos** = 0;  
 }  
 **currentSongIndex** = 0;  
 **btnPlay** = (ImageButton) findViewById(R.id.***btnPlay***);  
 **SongsName** = (TextView) findViewById(R.id.***SongsName***);  
 **SongsName**.setText(**"Title"**);  
 **Cover** = (ImageView) findViewById(R.id.***trackcover***);  
 Bitmap bitmap = BitmapFactory.*decodeResource*(getResources(), R.drawable.***adele***);  
 **Cover**.setImageBitmap(bitmap);  
 **mp** = **new** MediaPlayer();  
 **utils** = **new** Utilities();  
 **btnPlay**.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View arg0) {  
 **if** (MainActivity.*exlist*.get(**pos**).**objects**.get(**obpos**).**obselectedaudios**.size() == 0)  
 **return**;  
 **if** (**mp**.isPlaying()) {  
 **if** (**mp** != **null**) {  
 **mp**.pause();  
 **btnPlay**.setImageResource(R.drawable.***play***);  
 }  
 } **else** {  
 **if** (**mp** != **null**) {  
 playSong(**currentSongIndex**);  
 **btnPlay**.setImageResource(R.drawable.stop);  
 }  
 }  
 }  
 });  
 ListView listView = (ListView) **this**.findViewById(R.id.***list***);  
 listView.setAdapter(**new** AudioGalleryAdapter());  
 Button back = (Button) findViewById(R.id.***backbutstep3***);  
 back.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View view) {  
 **mp**.pause();  
 **btnPlay**.setImageResource(R.drawable.***play***);  
 MainActivity.*exlist*.get(**pos**).**objects**.get(**obpos**).**obselectedaudios**.clear();  
 *checkboxes*.clear();  
 *viewholders*.clear();  
 finish();  
 finishActivity(3);  
 Intent intent = **new** Intent(AudioActivity.**this**, VideoActivity.**class**);  
 intent.putExtra(**"position"**, Integer.*toString*(**pos**));  
 intent.putExtra(**"obposition"**, Integer.*toString*(**obpos**));  
 startActivity(intent);  
 *// set the animation to move once the button is clicked* overridePendingTransition(R.anim.***slide\_in***, R.anim.***slide\_out***);  
 }  
 });  
 Button selectBtn = (Button) findViewById(R.id.***selectBtnstep3***);  
 selectBtn.setOnClickListener(**new** View.OnClickListener() {  
 **public void** onClick(View v) {  
 **if** (**mp**.isPlaying()) {  
 **mp**.pause();  
 **btnPlay**.setImageResource(R.drawable.stop);  
 }  
 Intent in = **new** Intent(AudioActivity.**this**, TextActivity.**class**);  
 in.putExtra(**"position"**, Integer.*toString*(**pos**));  
 in.putExtra(**"obposition"**, Integer.*toString*(**obpos**));  
 startActivityForResult(in, 4);  
 *// set the animation to move once the button is clicked* overridePendingTransition(R.anim.***slide\_forw***, R.anim.***slide\_back***);  
 }  
 });  
 }  
  
 *//класс нужный для добавления номера конкретного аудио в список выбранных пользователем* **private class** AudioNum {  
 **int num**;  
 }  
  
 *//класс, отвечающий за отображения списка аудио на gridview, нажатие на элементы, возврат исходных значений при возврате из последующей активности* **class** AudioGalleryAdapter **extends** BaseAdapter {  
 LayoutInflater **inflater**;  
  
 **public** AudioGalleryAdapter() {  
 **inflater** = (LayoutInflater) getSystemService(Context.***LAYOUT\_INFLATER\_SERVICE***);  
 }  
  
 **public int** getCount() {  
 **return** LoadActivity.*audiosList*.size();  
 }  
  
 **public** Object getItem(**int** position) {  
 **return** LoadActivity.*audiosList*.get(position);  
 }  
  
 **public long** getItemId(**int** position) {  
 **return** position;  
 }  
  
 *//инициализация, загрузка аудио с карты памяти* **public** View getView(**int** position, View convertView, ViewGroup parent) {  
 **final** ViewHolder holder;  
 **if** (convertView == **null**) {  
 LayoutInflater vi = (LayoutInflater) getSystemService(Context.***LAYOUT\_INFLATER\_SERVICE***);  
 convertView = vi.inflate(R.layout.***video\_info***, **null**);  
 holder = **new** ViewHolder();  
 holder.**code** = (TextView) convertView.findViewById(R.id.***code***);  
 holder.**name** = (CheckBox) convertView.findViewById(R.id.***checkBox1***);  
 convertView.setTag(holder);  
 } **else** holder = (ViewHolder) convertView.getTag();  
 DataClass vid = LoadActivity.*audiosList*.get(position);  
 **boolean** flag = **false**;  
 **for** (**int** q = 0; q < *checkboxes*.size(); q++)  
 **if** (*checkboxes*.get(q).**num** == position) {  
 vid.setSelected(**true**);  
 flag = **true**;  
 }  
 **if** (!flag)  
 vid.setSelected(**false**);  
 holder.**name**.setOnClickListener(**new** View.OnClickListener() {  
 **public void** onClick(View v) {  
 ImageButton pl = (ImageButton) findViewById(R.id.***btnPlay***);  
 pl.setEnabled(**true**);  
 CheckBox cb = (CheckBox) v;  
 DataClass w = (DataClass) cb.getTag();  
 **int** id = 0;  
 **for** (**int** i = 0; i < LoadActivity.*audiosList*.size(); i++) {  
 **if** (LoadActivity.*audiosList*.get(i).**filePath**.equals(w.**filePath**)) {  
 id = i;  
 **break**;  
 }  
 }  
 **currentSongIndex** = LoadActivity.*audiosList*.indexOf(w);  
 **if** (LoadActivity.*audiosList*.get(id).isSelected()) {  
 **if** (**mp**.isPlaying()) {  
 **if** (**mp** != **null**) {  
 **mp**.pause();  
 **btnPlay**.setImageResource(R.drawable.***play***);  
 }  
 }  
 **for** (**int** k = 0; k < MainActivity.*exlist*.get(**pos**).**objects**.get(**obpos**).**obselectedaudios**.size(); k++)  
 **if** (MainActivity.*exlist*.get(**pos**).**objects**.get(**obpos**).**obselectedaudios**.get(k).**id** == LoadActivity.*audiosList*.get(id).**id**) {  
 MainActivity.*exlist*.get(**pos**).**objects**.get(**obpos**).**obselectedaudios**.remove(k);  
 **break**;  
 }  
 cb.setChecked(**false**);  
 LoadActivity.*audiosList*.get(id).setSelected(**false**);  
 **for** (**int** q = 0; q < *checkboxes*.size(); q++)  
 **if** (*checkboxes*.get(q).**num** == id) {  
 *checkboxes*.remove(q);  
 q--;  
 }  
 *//playSong(currentSongIndex);* } **else** {  
 cb.setChecked(**true**);  
 LoadActivity.*audiosList*.get(id).setSelected(**true**);  
 MainActivity.*exlist*.get(**pos**).**objects**.get(**obpos**).**obselectedaudios**.add(LoadActivity.*audiosList*.get(id));  
 AudioNum e = **new** AudioNum();  
 e.**num** = id;  
 *checkboxes*.add(e);  
 playSong(**currentSongIndex**);  
 }  
 }  
 });  
 holder.**code**.setText(vid.**title**);  
 holder.**name**.setChecked(vid.isSelected());  
 holder.**name**.setTag(vid);  
 *viewholders*.add(holder);  
 **return** convertView;  
 }  
 }  
  
 *//Получение индекса песни из списка воспроизведения и ее воспроизведение* @Override  
 **protected void** onActivityResult(**int** requestCode, **int** resultCode, Intent data) {  
 **super**.onActivityResult(requestCode, resultCode, data);  
 **if** (resultCode == 100) {  
 **currentSongIndex** = data.getExtras().getInt(**"songIndex"**);  
 *//играем выбранную песню* playSong(**currentSongIndex**);  
 }  
 }  
  
 *//воcпроизводим аудио* **public void** playSong(**int** songIndex) {  
 **try** {  
 **mp**.reset();  
 **mp**.setDataSource(LoadActivity.*audiosList*.get(songIndex).**filePath**);  
 **mp**.prepare();  
 **mp**.start();  
 **SongsName**.setText(LoadActivity.*audiosList*.get(songIndex).**title**);  
 Bitmap bitmap = **null**;  
 **try** {  
 bitmap = MediaStore.Images.Media.*getBitmap*(getContentResolver(), LoadActivity.*audiosList*.get(songIndex).**coverPath**);  
 bitmap = Bitmap.*createScaledBitmap*(bitmap, 600, 600, **true**);  
 } **catch** (FileNotFoundException exception) {  
 bitmap = BitmapFactory.*decodeResource*(getResources(), R.drawable.***adele***);  
 } **catch** (IOException e) {  
 e.printStackTrace();  
 }  
 **Cover**.setImageBitmap(bitmap);  
 **btnPlay**.setImageResource(R.drawable.stop);  
 } **catch** (IllegalArgumentException | IllegalStateException | IOException e) {  
 e.printStackTrace();  
 }  
 }  
  
 **public void** updateProgressBar() {  
 **mHandler**.postDelayed(**mUpdateTimeTask**, 100);  
 }  
  
 **private** Runnable **mUpdateTimeTask** = **new** Runnable() {  
 **public void** run() {  
 **mHandler**.postDelayed(**this**, 100);  
 }  
 };  
  
 @Override  
 **public void** onProgressChanged(SeekBar seekBar, **int** progress, **boolean** fromTouch) {  
 }  
  
 @Override  
 **public void** onStartTrackingTouch(SeekBar seekBar) {  
 **mHandler**.removeCallbacks(**mUpdateTimeTask**);  
 }  
  
 @Override  
 **public void** onStopTrackingTouch(SeekBar seekBar) {  
 **mHandler**.removeCallbacks(**mUpdateTimeTask**);  
 **int** totalDuration = **mp**.getDuration();  
 **int** currentPosition = **utils**.progressToTimer(seekBar.getProgress(), totalDuration);  
 **mp**.seekTo(currentPosition);  
 updateProgressBar();  
 }  
  
 @Override  
 **public void** onCompletion(MediaPlayer arg0) {  
 playSong(**currentSongIndex**);  
 }  
  
 @Override  
 **public void** onDestroy() {  
 **super**.onDestroy();  
 **mp**.release();  
 }  
  
 @Override  
 **public boolean** onCreateOptionsMenu(Menu menu) {  
 getMenuInflater().inflate(R.menu.***menu\_audio***, menu);  
 **return true**;  
 }  
  
 @Override  
 **public boolean** onOptionsItemSelected(MenuItem item) {  
 **int** id = item.getItemId();  
 **if** (id == R.id.***action\_settings***) {  
 **return true**;  
 }  
 **return super**.onOptionsItemSelected(item);  
 }  
  
 *//обработка нажатия на аппаратную кнопку назад* @Override  
 **public boolean** onKeyDown(**int** keyCode, KeyEvent event) {  
 **if** (keyCode == KeyEvent.***KEYCODE\_BACK***) {  
 **mp**.pause();  
 **btnPlay**.setImageResource(R.drawable.***play***);  
 MainActivity.*exlist*.get(**pos**).**objects**.get(**obpos**).**obselectedaudios**.clear();  
 *checkboxes*.clear();  
 *viewholders*.clear();  
 finish();  
 finishActivity(3);  
 Intent intent = **new** Intent(**this**, VideoActivity.**class**);  
 intent.putExtra(**"position"**, Integer.*toString*(**pos**));  
 intent.putExtra(**"obposition"**, Integer.*toString*(**obpos**));  
 startActivity(intent);  
 *// set the animation to move once the button is clicked* overridePendingTransition(R.anim.***slide\_in***, R.anim.***slide\_out***);  
 **return true**;  
 }  
 **return false**;  
 }  
}

# Интерфейс API\_Listener

**package** com.example.anast.app;  
  
**public interface** API\_Listener   
{  
 **public void** onSuccess(**int** requestnumber, Object obj);*//загрузка прошла успешно* **public void** onFail(String errormessage);*//возникли проблемы при загрузке*}

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ | | | | | | | | | | |
| Изм. |  | Номера листов (страниц) | | | | Всего  листов  (страниц) в  документе | №  документа | Входящий №  сопроводительн  ого документа и  дата | Подпись | Дата |
| измененн  ых | | замененны  х | новых | аннулиро  ванных |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |

# ПРИЛОЖЕНИЕ 1

# СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ

1. ГОСТ 19.101-77 Виды программ и программных документов. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
2. ГОСТ 19.102-77 Стадии разработки. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
3. ГОСТ 19.103-77 Обозначения программ и программных документов. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
4. ГОСТ 19.104-78 Основные надписи. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
5. ГОСТ 19.105-78 Общие требования к программным документам. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
6. ГОСТ 19.106-78 Требования к программным документам, выполненным печатным способом. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
7. ГОСТ 19.401-78 Текст программы. Требования к содержанию и оформлению. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
8. ГОСТ 19.603-78 Общие правила внесения изменений. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
9. ГОСТ 19.604-78 Правила внесения изменений в программные документы, выполненные печатным способом. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.