Государственное образовательное учреждение

высшего профессионального образования

«Донецкий национальный технический университет»

Кафедра АСУ

ОТЧЕТ

по лабораторной работе № 1

по дисциплине

«Программирование мобильных устройств»

по теме

«РАБОТА С БАЗАМИ ДАННЫХ»

Выполнил:

студент группы ИС-17 (а)

Канатуш Сергей

Проверили:

Хмелевой С. В.

Донецк – 2020

Порядок выполнения работы

При домашней подготовке самостоятельно на основе материалов лекций, справочной системы соответствующих средств разработки и отладки, а также рекомендуемой литературы изучить:

1. Средства работы с базами данных класса SQLiteDatabase;
2. Интерфейсы работы с базами данных класса SQLiteOpenHelper;
3. Средства извлечения строк из баз данных Cursor;
4. Запросы Query, методы Insert, Update, Delete для работы с базами данных.

Задание

Разработать Android-приложение, которое предназначено для изучения иностранных слов и выполняет функции:

- запись новых слов в базу с экрана;

- удаление слов в архив по правильному их угадыванию;

- вывод на экран случайного слова из базы данных и несколько случайных значений кроме правильного.

Задания повышенной сложности

- запись новых слов через текстовый файл, загружаемый с компьютера;

- запись в базу данных звучания слова, их воспроизводство;

- проверка всех направлений перевода, удаление после нескольких безошибочных угадываний слова.

ЛИСТИНГ ПРОГРАММЫ:

**MainActivity.java**

package com.android.lab\_1;  
  
import androidx.appcompat.app.AppCompatActivity;  
import androidx.core.app.ActivityCompat;  
import androidx.core.content.ContextCompat;  
  
import android.Manifest;  
import android.content.Context;  
import android.content.Intent;  
import android.content.pm.PackageManager;  
import android.os.Bundle;  
import android.util.Log;  
import android.view.Gravity;  
import android.view.View;  
import android.widget.Button;  
import android.widget.Toast;  
  
import java.io.BufferedReader;  
import java.io.FileReader;  
import java.io.IOException;  
import java.util.LinkedList;  
  
  
public class MainActivity extends AppCompatActivity {  
  
 static final String *PATH* = "/sdcard/words.txt";  
 Button add;  
 Button show;  
 Button game;  
 Button del;  
 Button file;  
 Button archieve;  
 Model model;  
  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
 Init();  
 }  
  
  
 void Init()  
 {  
  
 model = new Model(this);  
 add = (Button) findViewById(R.id.*add\_btn*);  
 game = (Button) findViewById(R.id.*game\_btn*);  
 show = (Button) findViewById(R.id.*show\_all\_btn*);  
 archieve = (Button) findViewById(R.id.*show\_archieve\_btn*);  
 del = (Button) findViewById(R.id.*del\_btn*);  
 file = (Button) findViewById(R.id.*read\_file\_btn*);  
  
 add.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 Intent intent = new Intent(getApplicationContext(), AddActivity.class);  
 if (intent.resolveActivity(getPackageManager()) != null) {  
 startActivity(intent);  
 }  
 }  
 });  
  
 show.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 Intent intent = new Intent(getApplicationContext(), ShowActivity.class);  
 if (intent.resolveActivity(getPackageManager()) != null) {  
 startActivity(intent);  
 }  
 }  
 });  
  
 archieve.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 Intent intent = new Intent(getApplicationContext(), ShowActivity.class);  
 intent.putExtra("selection", DBHandler.*KEY\_ARCHIEVE* + "= ?");  
 intent.putExtra("selection\_args", "1");  
 if (intent.resolveActivity(getPackageManager()) != null) {  
 startActivity(intent);  
 }  
 }  
 });  
  
 del.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 model.DeleteAllWords();  
 *MyToast*(getApplicationContext(), "Все слова из базы данных удалены...");  
 }  
 });  
  
 game.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 Intent intent = new Intent(getApplicationContext(), GameActivity.class);  
 if (intent.resolveActivity(getPackageManager()) != null) {  
 startActivity(intent);  
 }  
 }  
 });  
  
  
 file.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 LinkedList<WordsBean> beans = new LinkedList<>();  
  
 /\* Запрос на разрешения доступа к хранилищу \*/  
 if (ContextCompat.*checkSelfPermission*(getApplicationContext(), Manifest.permission.*WRITE\_EXTERNAL\_STORAGE*)!= PackageManager.*PERMISSION\_GRANTED*) {  
 if (ActivityCompat.*shouldShowRequestPermissionRationale*(MainActivity.this,  
 Manifest.permission.*WRITE\_EXTERNAL\_STORAGE*)) {  
  
 } else {  
 ActivityCompat.*requestPermissions*(MainActivity.this,  
 new String[]{Manifest.permission.*WRITE\_EXTERNAL\_STORAGE*}, 1);  
 }  
 }  
  
 try {  
 FileReader fr = new FileReader(*PATH*);  
 BufferedReader br = new BufferedReader(fr);  
  
 if (fr == null || br == null)  
 {  
 throw new IOException("The path is uncorrect...");  
 }  
 else {  
 Log.*d*("Lab1", "File is open...");  
 }  
  
 String str;  
 while((str = br.readLine()) != null) {  
 Log.*d*("Lab1", str);  
 beans.add(Parsing(str));  
 model.AddWords(beans.getLast().rus, beans.getLast().eng);  
 }  
  
 br.close();  
 fr.close();  
 *MyToast*(getApplicationContext(), "Файлы успешно считаны...");  
 } catch (IOException e) {  
 *MyToast*(getApplicationContext(), e.getMessage());  
 return;  
 }  
 }  
 });  
  
 }  
  
 static public void MyToast(Context context, String string)  
 {  
 Toast toast = Toast.*makeText*(context, string, Toast.*LENGTH\_SHORT*);  
 toast.setGravity(Gravity.*BOTTOM*, 0, -300);  
 toast.show();  
 }  
  
 private WordsBean Parsing(String str)  
 {  
 String eng\_rus[] = str.split(";");  
 return new WordsBean(eng\_rus[0], eng\_rus[1]);  
 }  
}

**GameActivity.java**

package com.android.lab\_1;  
  
import androidx.appcompat.app.AppCompatActivity;  
import android.content.Intent;  
import android.os.Bundle;  
import android.util.Log;  
import android.view.View;  
import android.widget.Button;  
import android.widget.TextView;  
import java.util.LinkedList;  
import java.util.Random;  
  
public class GameActivity extends AppCompatActivity implements View.OnClickListener {  
  
 public final static int *SIZE\_BTNS* = 4;  
 public final static int *NUM\_GUESSING* = 3;  
 private String true\_word;  
 private int true\_id;  
 private LinkedList<Integer> cur\_ids = new LinkedList<>();  
 private Button[] btns;  
 private Model model;  
 private TextView giving\_word;  
 private int score;  
  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_game*);  
 model = new Model(this);  
  
 if (model.getNumRows(DBHandler.*KEY\_ARCHIEVE* + " = ?", new String[] { "0" }) >= *SIZE\_BTNS*)  
 {  
 giving\_word = findViewById(R.id.*giving\_word*);  
 model.OutputLog();  
 InitBtns();  
 } else {  
 Intent intent = new Intent(this, MainActivity.class);  
 MainActivity.*MyToast*(getApplicationContext(), "В базе данных недостаточно слов...");  
 startActivity(intent);  
 }  
 }  
  
  
 private void InitBtns()  
 {  
 btns = new Button[*SIZE\_BTNS*];  
 btns[0] = (Button) findViewById(R.id.*button1*);  
 btns[1] = (Button) findViewById(R.id.*button2*);  
 btns[2] = (Button) findViewById(R.id.*button3*);  
 btns[3] = (Button) findViewById(R.id.*button4*);  
 StartGame();  
 }  
  
  
 private void StartGame()  
 {  
 try {  
 cur\_ids.clear();  
 if (model.getNumRows(DBHandler.*KEY\_ARCHIEVE* + " = ?", new String[] { "0" }) < *SIZE\_BTNS*)  
 {  
 throw new Exception("В базе данных недостатчоно слов...");  
 }  
  
 Random random = new Random();  
 int chose = random.nextInt(*SIZE\_BTNS*);  
 Boolean is\_eng\_word = random.nextBoolean();  
 WordsBean bean = null;  
  
 for (int i = 0; i < *SIZE\_BTNS*; i++) {  
 bean = model.getRandomWord(cur\_ids);  
 btns[i].setOnClickListener(this);  
 if (is\_eng\_word) {  
 btns[i].setText(bean.eng);  
 if (i == chose) {  
 giving\_word.setText(bean.rus);  
 true\_word = bean.eng;  
 true\_id = bean.\_id;  
 }  
 }  
 else  
 {  
 btns[i].setText(bean.rus);  
 if (i == chose) {  
 giving\_word.setText(bean.eng);  
 true\_word = bean.rus;  
 true\_id = bean.\_id;  
 }  
 }  
 }  
 }  
 catch (Exception ex)  
 {  
 Intent intent = new Intent(this, MainActivity.class);  
 MainActivity.*MyToast*(getApplicationContext(), ex.getMessage());  
 startActivity(intent);  
 }  
 }  
  
 @Override  
 public void onClick(View v) {  
 String click\_word = ((Button) v).getText().toString();  
 if (click\_word.equals(true\_word)) {  
 score++;  
 MainActivity.*MyToast*(getApplicationContext(), "Количество очков: " + Integer.*toString*(score));  
 int nums = model.SetNums(true\_id);  
 Log.*d*("Lab1", "Слово: " + true\_word + " | кол-во: " + Integer.*toString*(nums));  
 if (nums >= *NUM\_GUESSING*) {  
 model.GoToArchieve(true\_id);  
 MainActivity.*MyToast*(getApplicationContext(), "Слово '" + true\_word + "' добавлено в архив");  
 }  
 StartGame();  
 } else {  
 score = 0;  
 MainActivity.*MyToast*(getApplicationContext(), "Неверно!");  
 }  
 }  
}

**DBHandler.java**

package com.android.lab\_1;  
  
import android.content.Context;  
import android.database.sqlite.SQLiteDatabase;  
import android.database.sqlite.SQLiteOpenHelper;  
import android.util.Log;  
  
  
public class DBHandler extends SQLiteOpenHelper {  
  
 public final static String *DB\_NAME* = "words";  
 public final static int *DB\_VERSION* = 2;  
 public final static String *DB\_TABLE\_TRANSLATE\_WORDS* = "TRANSLATE\_WORDS";  
  
 public final static String *KEY\_ID* = "\_id";  
 public final static String *KEY\_RUS\_WORD* = "rus";  
 public final static String *KEY\_ENG\_WORD* = "eng";  
 public final static String *KEY\_ARCHIEVE* = "archieve";  
 public final static String *KEY\_NUMS* = "nums";  
  
 final String LOG\_DBHANDLER = "ACTION BY DBHandler";  
  
  
 public DBHandler(Context context) {  
 super(context, *DB\_NAME*, null, *DB\_VERSION*);  
 Log.*d*(LOG\_DBHANDLER, "--- Constructor created ---");  
 }  
  
 @Override  
 public void onCreate(SQLiteDatabase db) {  
 Log.*d*(LOG\_DBHANDLER, "--- onCreate database ---");  
 String sql = "CREATE TABLE " + *DB\_TABLE\_TRANSLATE\_WORDS* + "(" +  
 *KEY\_ID* + " integer primary key autoincrement," +  
 *KEY\_RUS\_WORD* + " text," +  
 *KEY\_ENG\_WORD* + " text," +  
 *KEY\_ARCHIEVE* + " integer," +  
 *KEY\_NUMS* + " integer" +  
 ");";  
 db.execSQL(sql);  
 }  
  
 @Override  
 public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {  
 db.execSQL("drop table if exists " + *DB\_TABLE\_TRANSLATE\_WORDS* + "");  
 onCreate(db);  
 }  
}

**Model.java**

package com.android.lab\_1;  
  
import android.content.ContentValues;  
import android.content.Context;  
import android.database.Cursor;  
import android.database.sqlite.SQLiteDatabase;  
import android.util.Log;  
  
import java.util.LinkedList;  
import java.util.Random;  
  
  
public class Model {  
  
 SQLiteDatabase db;  
 DBHandler dbHandler;  
 final String LOG\_MODEL = "ACTION BY Model";  
  
  
 Model(Context context) {  
 dbHandler = new DBHandler(context);  
 db = dbHandler.getWritableDatabase();  
 }  
  
 public long AddWords(String rus, String eng) {  
 // SQLiteDatabase - для управления бд  
 // ContentValues - для добавления новых строк в таблицу  
 ContentValues cv = new ContentValues();  
 cv.put(DBHandler.*KEY\_RUS\_WORD*, rus);  
 cv.put(DBHandler.*KEY\_ENG\_WORD*, eng);  
 cv.put(DBHandler.*KEY\_ARCHIEVE*, 0);  
 cv.put(DBHandler.*KEY\_NUMS*, 0);  
  
 long row\_id = db.insert(DBHandler.*DB\_TABLE\_TRANSLATE\_WORDS*, null, cv);  
 Log.*d*(LOG\_MODEL, "row inserted, ID = " + row\_id);  
 return row\_id;  
 }  
  
  
 public Cursor ShowWords(String selection, String[] selection\_args) {  
 Cursor c = db.query(DBHandler.*DB\_TABLE\_TRANSLATE\_WORDS*, null, selection, selection\_args, null, null, null);  
 return c;  
 }  
  
  
 public void DeleteAllWords() {  
 db.execSQL("DELETE FROM " + DBHandler.*DB\_TABLE\_TRANSLATE\_WORDS*);  
 }  
  
  
 public int getNumRows(String selection, String[] selection\_args) {  
 Cursor c = db.query(DBHandler.*DB\_TABLE\_TRANSLATE\_WORDS*, null, selection, selection\_args, null, null, null);  
 int num = c.getCount();  
 c.close();  
 return num;  
 }  
  
  
 public void GoToArchieve(int \_id)  
 {  
 ContentValues cv = new ContentValues();  
 cv.put(DBHandler.*KEY\_ARCHIEVE*, 1);  
 db.update(DBHandler.*DB\_TABLE\_TRANSLATE\_WORDS*, cv, "\_id = ?", new String[] { Integer.*toString*(\_id) });  
 }  
  
  
 public int SetNums(int \_id)  
 {  
 Cursor c = db.query(DBHandler.*DB\_TABLE\_TRANSLATE\_WORDS*,  
 new String[] { DBHandler.*KEY\_NUMS* },  
 DBHandler.*KEY\_ID* + " = ?",  
 new String[] {Integer.*toString*(\_id)}, null, null, null);  
 c.moveToFirst();  
 int nums = c.getInt(c.getColumnIndex(DBHandler.*KEY\_NUMS*));  
 nums++;  
 ContentValues cv = new ContentValues();  
 cv.put(DBHandler.*KEY\_NUMS*, nums);  
 db.update(DBHandler.*DB\_TABLE\_TRANSLATE\_WORDS*, cv, "\_id = ?", new String[] { Integer.*toString*(\_id) });  
 c.close();  
 return nums;  
 }  
  
  
 public void OutputLog() {  
 Cursor c = db.query(DBHandler.*DB\_TABLE\_TRANSLATE\_WORDS*, null, null, null, null, null, null);  
 if (c != null) {  
 c.moveToFirst();  
 do {  
 Log.*d*("Lab1", c.getString(c.getColumnIndex(DBHandler.*KEY\_ENG\_WORD*)) +  
 ", Архив: " + c.getInt(c.getColumnIndex(DBHandler.*KEY\_ARCHIEVE*)) +  
 ", Кол-во: " + c.getInt(c.getColumnIndex(DBHandler.*KEY\_NUMS*)));  
 } while (c.moveToNext());  
 }  
 c.close();  
 }  
  
  
 public WordsBean getRandomWord(LinkedList<Integer> cur\_ids) throws Exception {  
 Cursor c = db.query(DBHandler.*DB\_TABLE\_TRANSLATE\_WORDS*, null,  
 DBHandler.*KEY\_ARCHIEVE* + " = ?", new String[] { Integer.*toString*(0) }, null, null, null);  
 c.moveToFirst();  
 Random random = new Random();  
 int nums = getNumRows(DBHandler.*KEY\_ARCHIEVE* + " = ?", new String[] { "0" });  
 int rand = -1;  
 int i = 0;  
  
 for (; ;) {  
 rand = random.nextInt(nums);  
 Boolean currents = cur\_ids.contains(rand);  
  
 if (!currents) {  
 cur\_ids.add(rand);  
 break;  
 }  
 }  
  
 if (rand == -1) {  
 throw new Exception("Не хватает слов для начала игры!");  
 }  
  
 c.move(rand);  
  
 WordsBean bean = new WordsBean(  
 c.getInt(c.getColumnIndex(DBHandler.*KEY\_ID*)),  
 c.getString(c.getColumnIndex(DBHandler.*KEY\_RUS\_WORD*)),  
 c.getString(c.getColumnIndex(DBHandler.*KEY\_ENG\_WORD*)));  
 c.close();  
 return bean;  
 }  
}

**ShowActivity.java**

package com.android.lab\_1;  
  
import androidx.appcompat.app.AppCompatActivity;  
import androidx.cursoradapter.widget.SimpleCursorAdapter;  
  
import android.content.Intent;  
import android.database.Cursor;  
import android.os.Bundle;  
import android.speech.tts.TextToSpeech;  
import android.view.View;  
import android.widget.AdapterView;  
import android.widget.ListView;  
import android.widget.TextView;  
  
  
import java.util.Locale;  
  
  
public class ShowActivity extends AppCompatActivity implements TextToSpeech.OnInitListener {  
  
 ListView listView;  
 Model model;  
 TextToSpeech myTextToSpeech;  
 private boolean mIsInit;  
  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_show*);  
 Init();  
 }  
  
  
 void Init()  
 {  
 model = new Model(this);  
 myTextToSpeech = new TextToSpeech(this, this);  
 listView = findViewById(R.id.*listen*);  
 Intent intent = getIntent();  
  
 String selection = "";  
 if (intent.hasExtra("selection")) {  
 selection = intent.getStringExtra("selection");  
 } else {  
 selection = null;  
 }  
  
 String[] selection\_args = new String[1];  
 if (intent.hasExtra("selection\_args")) {  
 selection\_args[0] = intent.getStringExtra("selection\_args");  
 } else {  
 selection\_args = null;  
 }  
  
 Cursor c = model.ShowWords(selection, selection\_args);  
 // Формируем столбцы сопоставления  
 String[] from = new String[] { DBHandler.*KEY\_RUS\_WORD*, DBHandler.*KEY\_ENG\_WORD* };  
 int[] to = new int[] { R.id.*rus*, R.id.*eng* };  
  
 // создааем адаптер и настраиваем список  
 SimpleCursorAdapter scAdapter = new SimpleCursorAdapter(this, R.layout.*item*, c, from, to);  
 listView.setAdapter(scAdapter);  
  
 listView.setOnItemClickListener(new AdapterView.OnItemClickListener() {  
 @Override  
 public void onItemClick(AdapterView<?> parent, View view, int position, long id) {  
 TextView tv = view.findViewById(R.id.*eng*);  
 String word = tv.getText().toString();  
 myTextToSpeech.speak(word, TextToSpeech.*QUEUE\_FLUSH*, null, "id1");  
 MainActivity.*MyToast*(getApplicationContext(), word);  
 }  
 });  
 }  
  
 @Override  
 public void onInit(int status) {  
 if (status == TextToSpeech.*SUCCESS*) {  
 Locale locale = new Locale("en\_US");  
 int result = myTextToSpeech.setLanguage(locale);  
 if (result == TextToSpeech.*LANG\_MISSING\_DATA* || result == TextToSpeech.*LANG\_NOT\_SUPPORTED*) {  
 mIsInit = false;  
 } else {  
 mIsInit = true;  
 }  
 } else {  
 mIsInit = false;  
 }  
 }  
  
 @Override  
 protected void onDestroy() {  
 super.onDestroy();  
 myTextToSpeech.shutdown();  
 }  
}