Белорусский Государственный Технологический Университет

Факультет информационных технологий

Отчет к лабораторной работе №2

«Работа с файлами»

Выполнил: студент 3 курса 7 группы Горбачев Д.А.

Проверила: Герман Юлия Олеговна

Минск 2018

**Цель.** Изучить простой способ чтения и записи данных в фал.

Ход выполнения

1. После записи в файла отобразить на экране диалоговое окно, подтверждающее запись или диагностирующее ошибку.
2. Создать файл для хранения маленького словаря английских слов и их переводов в виде (пример)

хлеб: bread

молоко:milk

Набираешь в поле ввода русское слово и получаешь его перевод. Для выполнения перевода реализовать отдельную кнопку.

1.Главный файл (Прописаны обработчики)

**import** android.content.ContentValues;  
**import** android.database.Cursor;  
**import** android.database.sqlite.SQLiteDatabase;  
**import** android.os.Environment;  
**import** android.support.v7.app.AppCompatActivity;  
**import** android.os.Bundle;  
**import** android.view.View;  
**import** android.widget.Button;  
**import** android.widget.EditText;  
**import** android.widget.TextView;  
**import** android.widget.Toast;  
**import** java.io.File;  
**import** java.io.IOException;  
**import** java.util.ArrayList;  
**import** java.util.List;  
**import** java.util.Map;  
**import** java.util.TreeMap;  
  
**public class** MainActivity **extends** AppCompatActivity {  
  
 **private final static** String ***pathDirectory*** = **"Dictionary"**;  
 **private final static** String ***nameFileWithDictionary*** = **"Dictionary.txt"**;  
 **private static** File *sdPath* = **null**;  
 **private static** File *sdFile* = **null**;  
 **private** EditText **engWord**, **rusWord**;  
 **private** TextView **dictionary** = **null**, **error** = **null**;  
 **private** Button **save**;  
 **private** Map<String, String> **map** = **null**;  
 **private** DBHelper **dbHelper**;  
  
  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_main***);  
 *sdPath* = **new** File(Environment.*getExternalStorageDirectory*().getAbsolutePath() + **"/"** + ***pathDirectory***);  
 *sdFile* = **new** File(*sdPath*, ***nameFileWithDictionary***);  
 **if**(!*sdPath*.isDirectory()){  
 *sdPath*.mkdirs();  
 Toast.*makeText*(getApplicationContext(), **"Папка создана!!!"**, Toast.***LENGTH\_SHORT***).show();  
 **if**(!*sdFile*.exists()) {  
 **try** {  
 **boolean** b = *sdFile*.createNewFile();  
 **if**(b){  
 Toast.*makeText*(getApplicationContext(), **"Файл создан!!!"**, Toast.***LENGTH\_SHORT***).show();  
 }  
 } **catch** (IOException e) {  
 e.printStackTrace();  
 }  
 }  
 }**else** {  
 Toast.*makeText*(getApplicationContext(), **"Папка уже существует!!!"**, Toast.***LENGTH\_SHORT***).show();  
 **if** (!*sdFile*.exists()) {  
 **try** {  
 **boolean** b = *sdFile*.createNewFile();  
 **if** (b) {  
 Toast.*makeText*(getApplicationContext(), **"Файл создан!!!"**, Toast.***LENGTH\_SHORT***).show();  
 }  
 } **catch** (IOException e) {  
 e.printStackTrace();  
 }  
 }**else** {  
 Toast.*makeText*(getApplicationContext(), **"Файл уже создан!!!"**, Toast.***LENGTH\_SHORT***).show();  
 }  
 }  
  
 **dbHelper** = **new** DBHelper(**this**);  
  
 */\* writeToDB("молоко", "milk");  
 writeToDB("яблоко", "apple");  
 writeToDB("дерево", "tree");  
 writeToDB("дом", "home");  
 writeToDB("человек", "person");  
 writeToDB("собака", "dog");  
 writeToDB("кошка", "cat");  
 writeToDB("глаз", "eye");  
 writeToDB("хлеб", "bread");  
 writeToDB("телефон", "phone");\*/* **rusWord** = (EditText)findViewById(R.id.***rusWord***);  
 **engWord** = (EditText)findViewById(R.id.***engWord***);  
 **dictionary** = (TextView)findViewById(R.id.***dicrionary***);  
 **error** = (TextView)findViewById(R.id.***error***);  
 **save** = (Button)findViewById(R.id.***save***);  
 **save**.setEnabled(**false**);  
 }  
  
 **public void** onSave(View v){  
 String eng = **engWord**.getText().toString();  
 String rus = **rusWord**.getText().toString();  
 **if**(!eng.isEmpty() && !rus.isEmpty()){  
 WorkWithFille.*write*(*sdFile*, rus + **" : "** + eng);  
 }**else**{  
 Toast.*makeText*(getApplicationContext(), **"Не все поля заполнены!!!"**, Toast.***LENGTH\_SHORT***).show();  
 }  
 }  
  
 **public void** onRead(View v){  
 List<String> list = WorkWithFille.*reader*(*sdFile*);  
 **dictionary**.setText(**"Dictionary\n"**);  
 **for** (String s : list){  
 **dictionary**.append(s + **"\n"**);  
 }  
 }  
  
 **public void** onTranslate(View v){  
 String s = **rusWord**.getText().toString();  
 queryDB();  
 **if**(!s.isEmpty()){  
 String temp = equlsWord(s);  
 **if**(temp.equals(**"error"**)){  
 Toast.*makeText*(getApplicationContext(), **"Вы ввели слово которого нет!!!"**, Toast.***LENGTH\_SHORT***).show();  
 **return**;  
 } **else if**(!temp.equals(s)){  
 **rusWord**.setText(temp);  
 **error**.setText(**"Я нашел ошибку и исправил её!!!"**);  
 }  
 **engWord**.setText(**map**.get(temp));  
 **save**.setEnabled(**true**);  
 }**else**{  
 Toast.*makeText*(getApplicationContext(), **"Вы не ввели слово!!!"**, Toast.***LENGTH\_SHORT***).show();  
 }  
 }  
  
 **private** String equlsWord(String s) {  
 List<Integer> a = **new** ArrayList<>();  
 **for** (String te : **map**.keySet()) {  
 a.add(Levinshtein.*equls*(te, s));  
 }  
 List<String> str = **new** ArrayList<String>(**map**.keySet());  
 **if** (a.contains(0)) {  
 **return** str.get(a.indexOf(0));  
 }**else if**(a.contains(1)) {  
 **return** str.get(a.indexOf(1));  
 } **else if** (a.contains(2)) {  
 **return** str.get(a.indexOf(2));  
 }**else**{  
 **return "error"**;  
 }  
 }  
  
  
 **private void** writeToDB(String rus, String eng){  
 SQLiteDatabase database = **dbHelper**.getWritableDatabase();  
 ContentValues contentValues = **new** ContentValues();  
 contentValues.put(DBHelper.***KEY\_RUS***, rus);  
 contentValues.put(DBHelper.***KEY\_ENG***, eng);  
 database.insert(DBHelper.***TABLE\_CONTACTS***, **null**, contentValues);  
 **dbHelper**.close();  
 }  
  
 **private void** queryDB(){  
 **map** = **new** TreeMap<>();  
 SQLiteDatabase database = **dbHelper**.getWritableDatabase();  
 Cursor cursor = database.query(DBHelper.***TABLE\_CONTACTS***, **null**, **null**, **null**, **null**, **null**, **null**);  
 **if** (cursor.moveToFirst()) {  
 **int** eng = cursor.getColumnIndex(DBHelper.***KEY\_ENG***);  
 **int** rus = cursor.getColumnIndex(DBHelper.***KEY\_RUS***);  
 **do** {  
 **map**.put(cursor.getString(rus),cursor.getString(eng));  
 } **while** (cursor.moveToNext());  
 }  
 cursor.close();  
 **dbHelper**.close();  
 }  
}

2.Работа с файлами

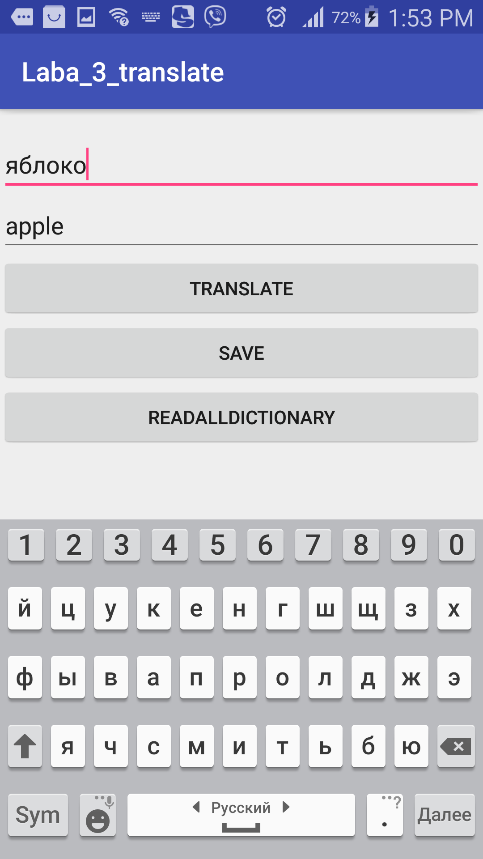
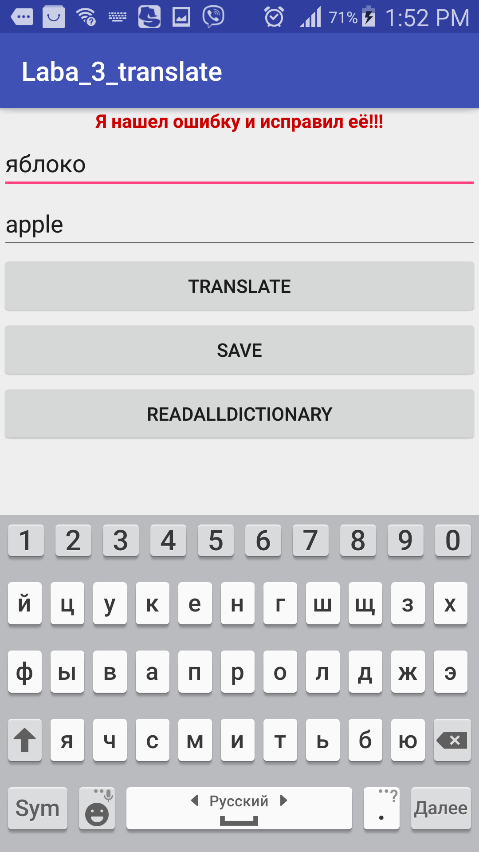
**import** java.io.BufferedReader;  
**import** java.io.BufferedWriter;  
**import** java.io.File;  
**import** java.io.FileReader;  
**import** java.io.FileWriter;  
**import** java.io.IOException;  
**import** java.util.ArrayList;  
**import** java.util.List;  
  
*/\*\*  
 \* Created by Dima on 22.02.2018.  
 \*/***public class** WorkWithFille {  
 **public static** List<String> reader(File path){  
 List<String> list = **new** ArrayList<>();  
 **try**(BufferedReader br = **new** BufferedReader(**new** FileReader(path)))  
 {  
 String s;  
 **while**((s=br.readLine())!=**null**){  
 list.add(s);  
 }  
 }  
 **catch**(IOException ex) {  
  
 System.***out***.println(ex.getMessage());  
 }  
 **return** list;  
 }  
  
 **public static void** write(File path, String text){  
 **try**(BufferedWriter bw = **new** BufferedWriter(**new** FileWriter(path, **true**)))  
 {  
 bw.write(text + **"\n"**);  
 }  
 **catch**(IOException ex){  
  
 System.***out***.println(ex.getMessage());  
 }  
 }  
}

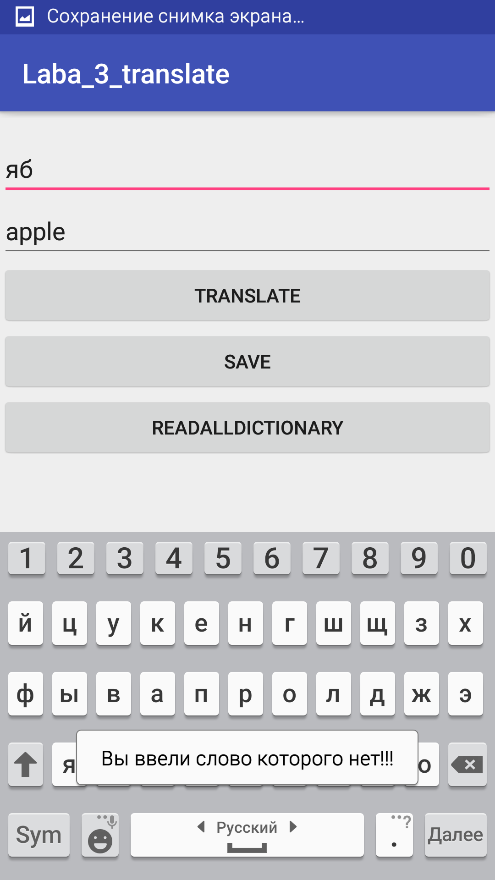
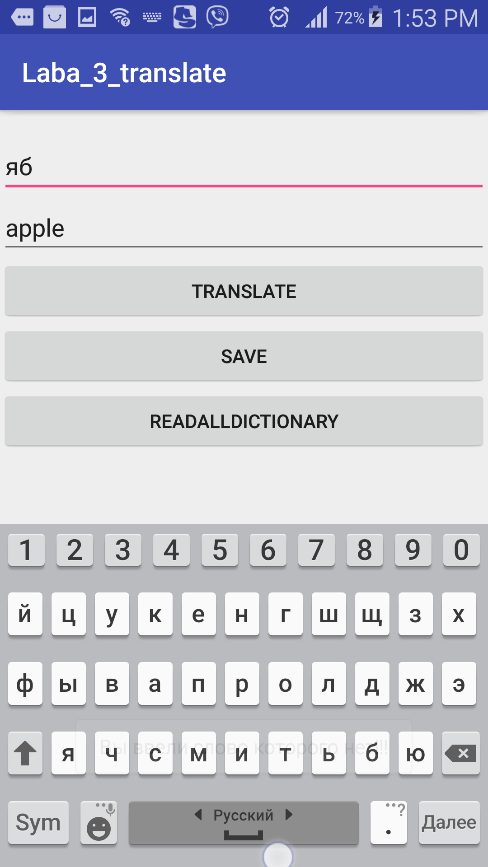
3.Сравнивание слов с помощью расстояния Левенштейна

**public class** Levinshtein {  
 **public static int** equls(String s1 ,String s2){  
 **int** m = s1.length(), n = s2.length();  
 **int**[] D1;  
 **int**[] D2 = **new int**[n + 1];  
  
 **for**(**int** i = 0; i <= n; i ++)  
 D2[i] = i;  
  
 **for**(**int** i = 1; i <= m; i ++) {  
 D1 = D2;  
 D2 = **new int**[n + 1];  
 **for**(**int** j = 0; j <= n; j ++) {  
 **if**(j == 0) D2[j] = i;  
 **else** {  
 **int** cost = (s1.charAt(i - 1) != s2.charAt(j - 1)) ? 1 : 0;  
 **if**(D2[j - 1] < D1[j] && D2[j - 1] < D1[j - 1] + cost)  
 D2[j] = D2[j - 1] + 1;  
 **else if**(D1[j] < D1[j - 1] + cost)  
 D2[j] = D1[j] + 1;  
 **else** D2[j] = D1[j - 1] + cost;  
 }  
 }  
 }  
 **return** D2[n];  
 }  
}

4.Работа с базой данных

**public class** DBHelper **extends** SQLiteOpenHelper{  
  
 **public static final int *DATABASE\_VERSION*** = 1;  
 **public static final** String ***DATABASE\_NAME*** = **"contactDb"**;  
 **public static final** String ***TABLE\_CONTACTS*** = **"contacts"**;  
  
 **public static final** String ***KEY\_ID*** = **"\_id"**;  
 **public static final** String ***KEY\_ENG*** = **"eng"**;  
 **public static final** String ***KEY\_RUS*** = **"rus"**;  
  
 **public** DBHelper(Context context) {  
 **super**(context, ***DATABASE\_NAME***, **null**, ***DATABASE\_VERSION***);  
 }  
  
 @Override  
 **public void** onCreate(SQLiteDatabase db) {  
 db.execSQL(**"create table "** + ***TABLE\_CONTACTS*** + **"("** + ***KEY\_ID*** + **" integer primary key,"** + ***KEY\_RUS*** + **" text,"** + ***KEY\_ENG*** + **" text"** + **")"**);  
  
 }  
  
 @Override  
 **public void** onUpgrade(SQLiteDatabase db, **int** oldVersion, **int** newVersion) {  
 db.execSQL(**"drop table if exists "** + ***TABLE\_CONTACTS***);  
  
 onCreate(db);  
  
 }  
}





Для записи во внешнее хранилище нужно еще указать в манифест файле разрешение:

<**uses-permission android:name="android.permission.WRITE\_EXTERNAL\_STORAGE"**/>  
<**uses-permission android:name="android.permission.READ\_EXTERNAL\_STORAGE"**/>

Вывод: Научились простой работе с файлами.