**Лабораторная работа 7.2.**

**БД SQLite в Android-приложении**

Краткая теория

Работа с базами данных SQLite в командной строке

Win+R

cmd

cd C:\2kurs\android\_sdk\tools

emulator-arm.exe -avd avd210

Win+R

cmd

cd C:\2kurs\android\_sdk\platform-tools

adb devices

adb shell

cd /data

ls

sqlite3 /data/linux\_icons.db

.databases

.tables

create table myimages (id integer primary key, imgname text);

.tables

select \* from myimages;

insert into myimages (id, imgname) values (1, 'sky');

select \* from myimages;

.mode column

.headers on

insert into myimages (id, imgname) values (2, 'cat-eye');

select \* from myimages;

insert into myimages (id, imgname) values (3, 'leaf');

insert into myimages (id, imgname) values (4, 'astronaut');

insert into myimages (id, imgname) values (5, 'chess');

select \* from myimages;

.exit

ls -l linux\_icons.db

exit

adb pull /data/linux\_icons.db ./

Копирование файлов в файловую систему Android-устройства с помощью IDE Eclipse

В Eclipse откройте меню   
Window/Show View/Other/Android/File Explorer

Файл базы данных в Android-устройстве можно найти в File Explorer, например, по следующему пути:   
/data/data/org.ggpi.MyImgSQLite01/databases/linux\_icons.db

Об отметках

*Основные* упражнения не отмечены никакими специальными знаками, их выполнения достаточно для удовлетворительной оценки. Дополнительные упражнения отмечены "звездочками", их выполнение необходимо для получения повышенной оценки.

* "5" - в отчете выполнены все *основные* упражнения, дополнительно выполнены упражнения с одной и двумя "звездочками".
* "4" - в отчете выполнены все *основные* упражнения дополнительно выполнены упражнения с одной "звездочкой".
* "3" - в отчете выполнены все *основные* упражнения.

Упражнения

1. База данных **linux\_icons.db** содержит одну таблицу *myimg*, в которой созданы две колонки:   
   1) *id* - номер изображения,   
   2) *imgname* - название файла с изображением.   
   Файлы в формате JPEG хранятся на Android-устройстве в папке **/data**. Возможно добавление записей в таблицу.

Упражнение 1

База данных **linux\_icons.db** содержит одну таблицу *myimg*, в которой созданы две колонки:   
1) *id* - номер изображения,   
2) *imgname* - название файла с изображением.   
Файлы в формате JPEG хранятся на Android-устройстве в папке **/data**. Возможно добавление записей в таблицу.   
Задание для самостоятельной работы: добавьте в приложение возможность создавать третью колонку в таблице: imganim, Анимация.   
**Внимание**! Откройте файл   
gen/org/ggpi/MyImgSQLite01/R.java   
и проверяйте **после** выполнения каждого из ниже приведённых соответствующих примеров, что в нём зарегистрированы объекты:

Самостоятельно определите: какие объекты здесь должны быть?

Указания к выполнению

1. После компиляции проверьте, создана ли база данных на Android-устройстве с помощью adb в командной строке, см. [Краткую теорию](https://sites.google.com/site/ifizmat/java/lab-02).

*Пример 1.1*. Файл **res/layout/main.xml**

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

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

android:layout\_width="fill\_parent"

android:layout\_height="fill\_parent"

android:orientation="vertical" >

<LinearLayout

android:id="@+id/linearLayout1"

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content" >

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginLeft="5dp"

android:layout\_marginRight="5dp"

android:text="ID" >

</TextView>

<EditText

android:id="@+id/etID"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_weight="1" >

<requestFocus>

</requestFocus>

</EditText>

</LinearLayout>

<LinearLayout

android:id="@+id/linearLayout3"

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content" >

<TextView

android:id="@+id/textView2"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginLeft="5dp"

android:layout\_marginRight="5dp"

android:text="Name" >

</TextView>

<EditText

android:id="@+id/etName"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_weight="1" >

</EditText>

</LinearLayout>

<LinearLayout

android:id="@+id/linearLayout2"

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content" >

<Button

android:id="@+id/btnAdd"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Add" >

</Button>

<Button

android:id="@+id/btnRead"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Read" >

</Button>

<Button

android:id="@+id/btnClear"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Clear" >

</Button>

</LinearLayout>

</LinearLayout>

*Пример 1.2*. Файл **MainActivity.java**

package org.ggpi.MyImgSQLite01;

import android.app.Activity;

import android.content.ContentValues;

import android.content.Context;

import android.database.Cursor;

import android.database.sqlite.SQLiteDatabase;

import android.database.sqlite.SQLiteOpenHelper;

import android.os.Bundle;

import android.util.Log;

import android.view.View;

import android.view.View.OnClickListener;

import android.widget.Button;

import android.widget.EditText;

public class MainActivity extends Activity implements OnClickListener {

private static final String DATABASE\_NAME = "linux\_icons.db";

private static final String DATABASE\_TABLE = "myimages";

final String LOG\_TAG = "myLogs";

Button btnAdd, btnRead, btnClear;

EditText etID, etName;

DBHelper dbHelper;

/\*\* Called when the activity is first created. \*/

@Override

public void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.main);

btnAdd = (Button) findViewById(R.id.btnAdd);

btnAdd.setOnClickListener(this);

btnRead = (Button) findViewById(R.id.btnRead);

btnRead.setOnClickListener(this);

btnClear = (Button) findViewById(R.id.btnClear);

btnClear.setOnClickListener(this);

etID = (EditText) findViewById(R.id.etID);

etName = (EditText) findViewById(R.id.etName);

// создаем объект для создания и управления версиями БД

dbHelper = new DBHelper(this);

}

@Override

public void onClick(View v) {

// создаем объект для данных

ContentValues cv = new ContentValues();

// получаем данные из полей ввода

String id = etID.getText().toString();

String name = etName.getText().toString();

// подключаемся к БД

SQLiteDatabase db = dbHelper.getWritableDatabase();

switch (v.getId()) {

case R.id.btnAdd:

Log.d(LOG\_TAG, "--- Insert in mytable: ---");

// подготовим данные для вставки в виде пар: наименование столбца - значение

cv.put("id", id);

cv.put("imgname", name);

// вставляем запись и получаем ее ID

long rowID = db.insert(DATABASE\_TABLE, null, cv);

Log.d(LOG\_TAG, "row inserted, ID = " + rowID);

break;

case R.id.btnRead:

Log.d(LOG\_TAG, "--- Rows in mytable: ---");

// делаем запрос всех данных из таблицы mytable, получаем Cursor

Cursor c = db.query(DATABASE\_TABLE, null, null, null, null, null, null);

// ставим позицию курсора на первую строку выборки

// если в выборке нет строк, вернется false

if (c.moveToFirst()) {

// определяем номера столбцов по имени в выборке

int idColIndex = c.getColumnIndex("id");

int nameColIndex = c.getColumnIndex("imgname");

do {

// получаем значения по номерам столбцов и пишем все в лог

Log.d(LOG\_TAG,

"ID = " + c.getInt(idColIndex) +

", name = " + c.getString(nameColIndex));

// переход на следующую строку

// а если следующей нет (текущая - последняя), то false - выходим из цикла

} while (c.moveToNext());

} else

Log.d(LOG\_TAG, "0 rows");

c.close();

break;

case R.id.btnClear:

Log.d(LOG\_TAG, "--- Clear mytable: ---");

// удаляем все записи

int clearCount = db.delete(DATABASE\_TABLE, null, null);

Log.d(LOG\_TAG, "deleted rows count = " + clearCount);

break;

}

// закрываем подключение к БД

dbHelper.close();

}

class DBHelper extends SQLiteOpenHelper {

public DBHelper(Context context) {

// конструктор суперкласса

super(context, DATABASE\_NAME, null, 1);

}

@Override

public void onCreate(SQLiteDatabase db) {

Log.d(LOG\_TAG, "--- onCreate database ---");

// создаем таблицу с полями

// db.execSQL("create table "+ DATABASE\_TABLE + " ("

// + "id integer primary key autoincrement,"

// + "imgname text" + ");");

}

@Override

public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {

}

}

}

1. В приложение из предыдущего упражнения следует добавить кнопку *Next*, при нажатии на которую на экран выводится изображения из папки **/data** в порядке их следования в таблице БД **linux\_icons.db**.

Упражнение 2

В приложение из предыдущего упражнения следует добавить кнопку *Next*, при нажатии на которую на экран выводится изображения из папки **/data** в порядке их следования в таблице БД **linux\_icons.db**.  
**Внимание**! Откройте файл   
gen/org/ggpi/MyImgSQLite02/R.java   
и проверяйте **после** выполнения каждого из ниже приведённых соответствующих примеров, что в нём зарегистрированы объекты

Самостоятельно определите: какие объекты здесь должны быть?

Указания к выполнению

1. Скопируйте 10 изображений в папку /data виртуального Android-устройства, см. [Краткую теорию](https://sites.google.com/site/ifizmat/java/lab-02) ко 2-й лабораторной работе.
2. После компиляции проверьте, создана ли база данных на Android-устройстве с помощью adb в командной строке, см. [Краткую теорию](https://sites.google.com/site/ifizmat/java/lab-03).

*Пример 2.1*. Файл **res/layout/main.xml**

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

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

android:layout\_width="fill\_parent"

android:layout\_height="fill\_parent"

android:orientation="vertical" >

<LinearLayout

android:id="@+id/linearLayout1"

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content" >

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginLeft="5dp"

android:layout\_marginRight="5dp"

android:text="ID" >

</TextView>

<EditText

android:id="@+id/etID"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_weight="1" >

<requestFocus>

</requestFocus>

</EditText>

</LinearLayout>

<LinearLayout

android:id="@+id/linearLayout3"

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content" >

<TextView

android:id="@+id/textView2"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginLeft="5dp"

android:layout\_marginRight="5dp"

android:text="Name" >

</TextView>

<EditText

android:id="@+id/etName"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_weight="1" >

</EditText>

</LinearLayout>

<LinearLayout

android:id="@+id/linearLayout2"

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content" >

<Button

android:id="@+id/btnAdd"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Add" >

</Button>

<Button

android:id="@+id/btnRead"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Read" >

</Button>

<Button

android:id="@+id/btnClear"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Clear" >

</Button>

<Button

android:id="@+id/btnNext"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Next" >

</Button>

</LinearLayout>

<ImageView android:id="@+id/image1"

android:src="@drawable/ic\_launcher"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:background="#000000"

android:padding="1dp"

android:layout\_gravity="center\_horizontal" />

</LinearLayout>

*Пример 2.2*. Файл **MainActivity.java**

package org.ggpi.MyImgSQLite02;

import android.app.Activity;

import android.content.ContentValues;

import android.content.Context;

import android.database.Cursor;

import android.database.sqlite.SQLiteDatabase;

import android.database.sqlite.SQLiteOpenHelper;

import android.graphics.drawable.Drawable;

import android.os.Bundle;

import android.util.Log;

import android.view.View;

import android.view.View.OnClickListener;

import android.widget.Button;

import android.widget.EditText;

import android.widget.ImageView;

public class MainActivity extends Activity implements OnClickListener {

private static final String DATABASE\_NAME = "linux\_icons.db";

private static final String DATABASE\_TABLE = "myimages";

final String LOG\_TAG = "myLogs";

Button btnAdd, btnRead, btnClear, btnNext;

EditText etID, etName;

ImageView image1;

// Счётчик для нажатий btnNext.

int countNext = 0;

DBHelper dbHelper;

/\*\* Called when the activity is first created. \*/

@Override

public void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.main);

btnAdd = (Button) findViewById(R.id.btnAdd);

btnAdd.setOnClickListener(this);

btnRead = (Button) findViewById(R.id.btnRead);

btnRead.setOnClickListener(this);

btnClear = (Button) findViewById(R.id.btnClear);

btnClear.setOnClickListener(this);

btnNext = (Button) findViewById(R.id.btnNext);

btnNext.setOnClickListener(this);

etID = (EditText) findViewById(R.id.etID);

etName = (EditText) findViewById(R.id.etName);

image1 = (ImageView) findViewById(R.id.image1);

// создаем объект для создания и управления версиями БД

dbHelper = new DBHelper(this);

}

@Override

public void onClick(View v) {

// создаем объект для данных

ContentValues cv = new ContentValues();

// получаем данные из полей ввода

String id = etID.getText().toString();

String name = etName.getText().toString();

// подключаемся к БД

SQLiteDatabase db = dbHelper.getWritableDatabase();

// делаем запрос всех данных из таблицы DATABASE\_TABLE, получаем Cursor

Cursor c = db.query(DATABASE\_TABLE, null, null, null, null, null, null);

switch (v.getId()) {

case R.id.btnNext:

boolean nextExist = false;

countNext++;

Log.d(LOG\_TAG, "--- Rows in mytable: ---");

if (countNext == 1) {

if (c.moveToFirst()) {

int idColIndex = c.getColumnIndex("id");

int nameColIndex = c.getColumnIndex("imgname");

Log.d(LOG\_TAG,

"ID = " + c.getInt(idColIndex) +

", name = " + c.getString(nameColIndex));

}

}

if (countNext > 1) {

int idColIndex = c.getColumnIndex("id");

int nameColIndex = c.getColumnIndex("imgname");

// Локальный счётчик пройденных записей таблицы.

int iCount = 0;

while (iCount < countNext) {

iCount++;

nextExist = c.moveToNext();

Log.d(LOG\_TAG,

"iCount = " + iCount + ", nextExist = " + nextExist);

}

if (!nextExist) {

countNext = 0;

c.moveToFirst();

} else {

Log.d(LOG\_TAG,

"ID = " + c.getInt(idColIndex) +

", name = " + c.getString(nameColIndex));

etID.setText(Integer.toString(c.getInt(idColIndex)));

etName.setText(c.getString(nameColIndex));

image1.setImageDrawable(Drawable.createFromPath("/data/"+c.getString(nameColIndex)+".jpg") );

}

}

c.close();

break;

case R.id.btnAdd:

Log.d(LOG\_TAG, "--- Insert in mytable: ---");

// подготовим данные для вставки в виде пар: наименование столбца - значение

cv.put("id", id);

cv.put("imgname", name);

// вставляем запись и получаем ее ID

long rowID = db.insert(DATABASE\_TABLE, null, cv);

Log.d(LOG\_TAG, "row inserted, ID = " + rowID);

break;

case R.id.btnRead:

Log.d(LOG\_TAG, "--- Rows in mytable: ---");

// делаем запрос всех данных из таблицы mytable, получаем Cursor

//Cursor c = db.query(DATABASE\_TABLE, null, null, null, null, null, null);

// ставим позицию курсора на первую строку выборки

// если в выборке нет строк, вернется false

if (c.moveToFirst()) {

// определяем номера столбцов по имени в выборке

int idColIndex = c.getColumnIndex("id");

int nameColIndex = c.getColumnIndex("imgname");

do {

// получаем значения по номерам столбцов и пишем все в лог

Log.d(LOG\_TAG,

"ID = " + c.getInt(idColIndex) +

", name = " + c.getString(nameColIndex));

// переход на следующую строку

// а если следующей нет (текущая - последняя), то false - выходим из цикла

} while (c.moveToNext());

} else

Log.d(LOG\_TAG, "0 rows");

c.close();

break;

case R.id.btnClear:

Log.d(LOG\_TAG, "--- Clear mytable: ---");

// удаляем все записи

int clearCount = db.delete(DATABASE\_TABLE, null, null);

Log.d(LOG\_TAG, "deleted rows count = " + clearCount);

break;

}

// закрываем подключение к БД

dbHelper.close();

}

class DBHelper extends SQLiteOpenHelper {

public DBHelper(Context context) {

// конструктор суперкласса

super(context, DATABASE\_NAME, null, 1);

}

@Override

public void onCreate(SQLiteDatabase db) {

Log.d(LOG\_TAG, "--- onCreate database ---");

// создаем таблицу с полями

db.execSQL("create table "+ DATABASE\_TABLE + " ("

+ "id integer primary key autoincrement,"

+ "imgname text" + ");");

}

@Override

public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {

}

}

}

1. База данных **img\_anim.db** содержит одну таблицу *myanim*, в которой созданы три колонки:   
   1) *id* - номер изображения,   
   2) *imgname* - название файла с изображением,   
   3) *imganim* - вид анимации.   
   При нажатии на кнопку *Next* загружается изображение, к нему применяется соответствующая анимация: *alpha*, *combo*, *rotate*, *scale*, *trans*.

#### Упражнение 3

База данных **img\_anim.db** содержит одну таблицу *myanim*, в которой созданы три колонки:   
1) *id* - номер изображения,   
2) *imgname* - название файла с изображением,   
3) *imganim* - вид анимации.   
При нажатии на кнопку *Next* загружается изображение, к нему применяется соответствующая анимация: *alpha*, *combo*, *rotate*, *scale*, *trans*.   
**Внимание**! Откройте файл   
gen/org/ggpi/MyImgSQLite03/R.java   
и проверяйте **после** выполнения каждого из ниже приведённых соответствующих примеров, что в нём зарегистрированы объекты

Самостоятельно определите: какие объекты здесь должны быть?

Для приложения можно скопировать, распаковав из архива, готовую базу данных img\_anim.db в папку на Android-устройство:   
/data/data/org.ggpi.MyImgSQLite03/databases/   
Архив примера БД. Правый щелчок мыши на ссылке вызовет выпадающее меню, в котором нужно выбрать команду "Сохранить объект как..."   
[ [Скачать файл](https://sites.google.com/site/ifizmat/myfiles/img_anim_db.zip?attredirects=0) ]

Если БД не открывается в приложении, то, возможно, требуется изменить права доступа. Для этого нужно перейти с помощью   
adb   
в папку   
/data/data/org.ggpi.MyImgSQLite03/databases/   
Проверить права доступа, изменить режим, убедиться, что изменения произведены:

ls -l img\_anim.db

chmod 777 img\_anim.db

ls -l img\_anim.db

Создайте папку   
**res/anim/**   
Из примеров прошлых занятий заполните её описаниями анимации:

myalpha.xml

mycombo.xml

myrotate.xml

myscale.xml

mytrans.xml

*Пример 3.1*. Файл **res/values/strings.xml**

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

<resources>

<string name="app\_name">MyImgSQLite03</string>

<string name="action\_settings">Settings</string>

<string name="id">ID</string>

<string name="img\_name">Name</string>

<string name="img\_anim">Animation</string>

<string name="btn\_add">Add</string>

<string name="btn\_read">Read</string>

<string name="btn\_clear">Clear</string>

<string name="btn\_next">Next</string>

</resources>

*Пример 3.2*. Файл **res/layout/main.xml**

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

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

android:layout\_width="fill\_parent"

android:layout\_height="fill\_parent"

android:orientation="vertical" >

<LinearLayout

android:id="@+id/linearLayout1"

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content" >

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginLeft="5dp"

android:layout\_marginRight="5dp"

android:text="@string/id" >

</TextView>

<EditText

android:id="@+id/etID"

android:layout\_width="0dp"

android:layout\_height="wrap\_content"

android:layout\_weight="1" >

<requestFocus>

</requestFocus>

</EditText>

</LinearLayout>

<LinearLayout

android:id="@+id/linearLayout3"

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content" >

<TextView

android:id="@+id/textView2"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginLeft="5dp"

android:layout\_marginRight="5dp"

android:text="@string/img\_name" >

</TextView>

<EditText

android:id="@+id/etName"

android:layout\_width="0dp"

android:layout\_height="wrap\_content"

android:layout\_weight="1" >

</EditText>

</LinearLayout>

<LinearLayout

android:id="@+id/linearLayout4"

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content" >

<TextView

android:id="@+id/textView3"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginLeft="5dp"

android:layout\_marginRight="5dp"

android:text="@string/img\_anim" >

</TextView>

<EditText

android:id="@+id/etAnim"

android:layout\_width="0dp"

android:layout\_height="wrap\_content"

android:layout\_weight="1" >

</EditText>

</LinearLayout>

<LinearLayout

android:id="@+id/linearLayout2"

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content" >

<Button

android:id="@+id/btnAdd"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="@string/btn\_add" >

</Button>

<Button

android:id="@+id/btnRead"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="@string/btn\_read" >

</Button>

<Button

android:id="@+id/btnClear"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="@string/btn\_clear" >

</Button>

<Button

android:id="@+id/btnNext"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="@string/btn\_next" >

</Button>

</LinearLayout>

<ImageView android:id="@+id/image1"

android:src="@drawable/ic\_launcher"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:background="#000000"

android:padding="1dp"

android:layout\_gravity="center\_horizontal" />

</LinearLayout>

*Пример 3.3*. Файл **MainActivity.java**

package org.ggpi.MyImgSQLite03;

import android.app.Activity;

import android.content.ContentValues;

import android.content.Context;

import android.database.Cursor;

import android.database.sqlite.SQLiteDatabase;

import android.database.sqlite.SQLiteOpenHelper;

import android.graphics.drawable.Drawable;

import android.os.Bundle;

import android.util.Log;

import android.view.View;

import android.view.View.OnClickListener;

import android.view.animation.Animation;

import android.view.animation.AnimationUtils;

import android.widget.Button;

import android.widget.EditText;

import android.widget.ImageView;

public class MainActivity extends Activity implements OnClickListener {

private static final String DATABASE\_NAME = "img\_anim.db";

private static final String DATABASE\_TABLE = "myanim";

final String LOG\_TAG = "myLogs";

Button btnAdd, btnRead, btnClear, btnNext;

EditText etID, etName, etAnim;

ImageView image1;

// Счётчик для нажатий btnNext.

int countNext = 0;

DBHelper dbHelper;

/\*\* Called when the activity is first created. \*/

@Override

public void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.main);

btnAdd = (Button) findViewById(R.id.btnAdd);

btnAdd.setOnClickListener(this);

btnRead = (Button) findViewById(R.id.btnRead);

btnRead.setOnClickListener(this);

btnClear = (Button) findViewById(R.id.btnClear);

btnClear.setOnClickListener(this);

btnNext = (Button) findViewById(R.id.btnNext);

btnNext.setOnClickListener(this);

etID = (EditText) findViewById(R.id.etID);

etName = (EditText) findViewById(R.id.etName);

etAnim = (EditText) findViewById(R.id.etAnim);

image1 = (ImageView) findViewById(R.id.image1);

// создаем объект для создания и управления версиями БД

dbHelper = new DBHelper(this);

}

@Override

public void onClick(View v) {

// создаем объект для данных

ContentValues cv = new ContentValues();

// получаем данные из полей ввода

String id = etID.getText().toString();

String name = etName.getText().toString();

String anim = etAnim.getText().toString();

Animation animImg = null;

// подключаемся к БД

SQLiteDatabase db = dbHelper.getWritableDatabase();

// делаем запрос всех данных из таблицы DATABASE\_TABLE, получаем Cursor

Cursor c = db.query(DATABASE\_TABLE, null, null, null, null, null, null);

int idColIndex = c.getColumnIndex("id");

int nameColIndex = c.getColumnIndex("imgname");

int animColIndex = c.getColumnIndex("imganim");

switch (v.getId()) {

case R.id.btnNext:

boolean nextExist = false;

countNext++;

Log.d(LOG\_TAG, "--- Rows in mytable: ---");

if (countNext == 1) {

if (c.moveToFirst()) {

Log.d(LOG\_TAG,

"ID = " + c.getInt(idColIndex) +

", name = " + c.getString(nameColIndex) +

", anim = " + c.getString(animColIndex));

}

}

if (countNext > 1) {

// Локальный счётчик пройденных записей таблицы.

int iCount = 0;

while (iCount < countNext) {

iCount++;

nextExist = c.moveToNext();

Log.d(LOG\_TAG,

"iCount = " + iCount + ", nextExist = " + nextExist);

}

if (!nextExist) {

countNext = 0;

c.moveToFirst();

} else {

int idAnim = 0;

String strAnim = c.getString(animColIndex).toString();

if (strAnim.equals("alpha")) idAnim = R.anim.myalpha;

if (strAnim.equals("combo")) idAnim = R.anim.mycombo;

if (strAnim.equals("rotate")) idAnim = R.anim.myrotate;

if (strAnim.equals("scale")) idAnim = R.anim.myscale;

if (strAnim.equals("trans")) idAnim = R.anim.mytrans;

Log.d(LOG\_TAG,

"ID = " + c.getInt(idColIndex) +

", name = " + c.getString(nameColIndex) +

", anim = " + c.getString(animColIndex) +

", idAnim = " + idAnim);

etID.setText(Integer.toString(c.getInt(idColIndex)));

etName.setText(c.getString(nameColIndex));

etAnim.setText(c.getString(animColIndex));

image1.setImageDrawable(Drawable.createFromPath("/data/"+c.getString(nameColIndex)+".jpg") );

animImg = AnimationUtils.loadAnimation(this, idAnim);

image1.startAnimation(animImg);

}

}

c.close();

break;

case R.id.btnAdd:

Log.d(LOG\_TAG, "--- Insert in mytable: ---");

// подготовим данные для вставки в виде пар: наименование столбца - значение

cv.put("id", id);

cv.put("imgname", name);

cv.put("imganim", anim);

// вставляем запись и получаем ее ID

long rowID = db.insert(DATABASE\_TABLE, null, cv);

Log.d(LOG\_TAG, "row inserted, ID = " + rowID);

break;

case R.id.btnRead:

Log.d(LOG\_TAG, "--- Rows in mytable: ---");

// делаем запрос всех данных из таблицы mytable, получаем Cursor

//Cursor c = db.query(DATABASE\_TABLE, null, null, null, null, null, null);

// ставим позицию курсора на первую строку выборки

// если в выборке нет строк, вернется false

if (c.moveToFirst()) {

// определяем номера столбцов по имени в выборке

do {

// получаем значения по номерам столбцов и пишем все в лог

Log.d(LOG\_TAG,

"ID = " + c.getInt(idColIndex) +

", name = " + c.getString(nameColIndex) +

", anim = " + c.getString(animColIndex));

// переход на следующую строку

// а если следующей нет (текущая - последняя), то false - выходим из цикла

} while (c.moveToNext());

} else

Log.d(LOG\_TAG, "0 rows");

c.close();

break;

case R.id.btnClear:

Log.d(LOG\_TAG, "--- Clear mytable: ---");

// удаляем все записи

int clearCount = db.delete(DATABASE\_TABLE, null, null);

Log.d(LOG\_TAG, "deleted rows count = " + clearCount);

break;

}

// закрываем подключение к БД

dbHelper.close();

}

class DBHelper extends SQLiteOpenHelper {

public DBHelper(Context context) {

// конструктор суперкласса

super(context, DATABASE\_NAME, null, 1);

}

@Override

public void onCreate(SQLiteDatabase db) {

Log.d(LOG\_TAG, "--- onCreate database ---");

// создаем таблицу с полями

db.execSQL("create table "+ DATABASE\_TABLE + " ("

+ "id integer primary key autoincrement,"

+ "imgname text,"

+ "imganim text" + ");");

}

@Override

public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {

}

}

}

1. \*Добавить кнопку *Prev*, Предыдущая запись таблицы.
2. \*\*БД должна содержать, кроме указанных в упражнениях выше полей, третью колонку: *time*, Время выполнения анимации.