Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное

учреждение высшего образования

«Комсомольский-на-Амуре государственный университет»

Факультет компьютерных технологий

Кафедра МОП ЭВМ

РАСЧЕТНО-ГРАФИЧЕСКАЯ РАБОТА

по дисциплине «Программирование мобильных устройств»

Вариант 1

Студент группы 6ВТб-1 А.А. Бабынин

Преподаватель В.А. Тихомиров

2019

**Задания**

Из базы данных dbdemos.db3 из таблицы events вывести наименование события, его описание и изображение, а из связанной таблицы venues – время события и схему места проведения события. События перелистываются кнопками.

Содержание

[Задание 4](#_Toc27567029)

[Текст программы 4](#_Toc27567030)

[Тестирование программы 8](#_Toc27567031)

[Список использованных источников 13](#_Toc27567032)

## Задание

Из базы данных dbdemos.db3 из таблицы events вывести наименование события, его описание и изображение, а из связанной таблицы venues – время события и схему места проведения события. События перелистываются кнопками.

## Текст программы

Проект состоит из двух файлов исходного кода, которые приведен в листинге 1.1 и в листинге 1.2.

Листинг 1.1 – Текст файла MainActivity.java

|  |
| --- |
| **package** ru.diaran.lab8;  **import** android.app.Activity; **import** android.content.ContentValues; **import** android.content.Context; **import** android.database.Cursor; **import** android.database.SQLException; **import** android.database.sqlite.SQLiteDatabase; **import** android.database.sqlite.SQLiteOpenHelper; **import** android.graphics.Bitmap; **import** android.graphics.BitmapFactory; **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;  **import** java.io.File; **import** java.io.FileOutputStream; **import** java.io.IOException; **import** java.io.InputStream; **import** java.io.OutputStream; **import** java.util.Arrays;  **public class** MainActivity **extends** Activity **implements** OnClickListener {  **private static** String *DB\_NAME* = **"dbdemos.db3"**;  **private static** String *DB\_PATH* = **""**;  **private static final int *DB\_VERSION*** = 1;  Bitmap **bitmap**=**null**;  ImageView **iv001**, **iv002**;   **final** String **LOG\_TAG** = **"myLogs"**;   Button **btnAdd**, **btnRead**, **btnClear**;  EditText **etName**, **txtDesc**, **txtTime**;   DBHelper **dbHelper**;   Cursor **c**;    **int photoeColIndex**;  **int nameColIndex**;  **int descColIndex**;  **int dateColIndex**;  **int timeColIndex**;  **int mapColIndex**;   */\*\* Called when the activity is first created. \*/* @Override  **public void** onCreate(Bundle savedInstanceState) {  **super**.onCreate(savedInstanceState);  setContentView(R.layout.***activity\_main***);   **btnAdd** = (Button) findViewById(R.id.***btnAdd***);  **btnAdd**.setOnClickListener(**this**);   **btnRead** = (Button) findViewById(R.id.***btnRead***);  **btnRead**.setOnClickListener(**this**);   **etName** = (EditText) findViewById(R.id.***etName***);  **txtDesc** = (EditText) findViewById(R.id.***txtDesc***);  **txtTime** = (EditText) findViewById(R.id.***txtTime***);   **iv002** = (ImageView) findViewById(R.id.***iV002***);  **iv001** = (ImageView) findViewById(R.id.***iV001***);      *// создаем объект для создания и управления версиями БД* **dbHelper** = **new** DBHelper(**this**);   *// подключаемся к БД* SQLiteDatabase db = **dbHelper**.getWritableDatabase();   String selectQuery = **"SELECT \* FROM events INNER JOIN venues ON venues.VenueNo = events.VenueNo"**;  **c** = db.rawQuery(selectQuery, **null**);  **if** (!**c**.moveToFirst())  Log.*d*(**LOG\_TAG**, **"0 rows"**);  **else** {  **photoeColIndex** = **c**.getColumnIndex(**"Event\_Photo"**);  **nameColIndex** = **c**.getColumnIndex(**"Event\_Name"**);  **descColIndex** = **c**.getColumnIndex(**"Event\_Description"**);  **dateColIndex** = **c**.getColumnIndex(**"Event\_Date"**);  **timeColIndex** = **c**.getColumnIndex(**"Event\_Time"**);  **mapColIndex** = **c**.getColumnIndex(**"Venue\_Map"**);   **byte**[] blob = **c**.getBlob(**photoeColIndex**);  *// Convert the byte array to Bitmap* **int** bLen = blob.**length**;  **byte**[] newblob = Arrays.*copyOfRange*(blob,8, bLen-8);  **bitmap** = BitmapFactory.*decodeByteArray*(newblob, 0, newblob.**length**);  **iv001**.setImageBitmap(**bitmap**);   blob = **c**.getBlob(**mapColIndex**);  *// Convert the byte array to Bitmap* bLen = blob.**length**;  newblob = Arrays.*copyOfRange*(blob,8, bLen-8);  **bitmap** = BitmapFactory.*decodeByteArray*(newblob, 0, newblob.**length**);  **iv002**.setImageBitmap(**bitmap**);   **etName**.setText(**c**.getString(**nameColIndex**));  **txtTime**.setText(**c**.getString(**dateColIndex**) + **" "** + **c**.getString(**timeColIndex**));  **txtDesc**.setText(**c**.getString(**descColIndex**));  }  }   @Override  **protected void** onDestroy() {  **c**.close();   *// закрываем подключение к БД* **dbHelper**.close();  **super**.onDestroy();  }   @Override  **public void** onClick(View v) {  *// получаем данные из полей ввода* String name = **etName**.getText().toString();  String desc = **txtDesc**.getText().toString();  String time = **txtTime**.getText().toString();   **byte**[] blob;  *// Convert the byte array to Bitmap* **int** bLen;  **byte**[] newblob;   **switch** (v.getId()) {  **case** R.id.***btnAdd***:  **c**.moveToPrevious();   blob = **c**.getBlob(**photoeColIndex**);  *// Convert the byte array to Bitmap* bLen = blob.**length**;  newblob = Arrays.*copyOfRange*(blob,8, bLen-8);  **bitmap** = BitmapFactory.*decodeByteArray*(newblob, 0, newblob.**length**);  **iv001**.setImageBitmap(**bitmap**);   blob = **c**.getBlob(**mapColIndex**);  *// Convert the byte array to Bitmap* bLen = blob.**length**;  newblob = Arrays.*copyOfRange*(blob,8, bLen-8);  **bitmap** = BitmapFactory.*decodeByteArray*(newblob, 0, newblob.**length**);  **iv002**.setImageBitmap(**bitmap**);   **etName**.setText(**c**.getString(**nameColIndex**));  **txtTime**.setText(**c**.getString(**dateColIndex**) + **" "** + **c**.getString(**timeColIndex**));  **txtDesc**.setText(**c**.getString(**descColIndex**));  **break**;   **case** R.id.***btnRead***:   **c**.moveToNext();   blob = **c**.getBlob(**photoeColIndex**);  *// Convert the byte array to Bitmap* bLen = blob.**length**;  newblob = Arrays.*copyOfRange*(blob,8, bLen-8);  **bitmap** = BitmapFactory.*decodeByteArray*(newblob, 0, newblob.**length**);  **iv001**.setImageBitmap(**bitmap**);   blob = **c**.getBlob(**mapColIndex**);  *// Convert the byte array to Bitmap* bLen = blob.**length**;  newblob = Arrays.*copyOfRange*(blob,8, bLen-8);  **bitmap** = BitmapFactory.*decodeByteArray*(newblob, 0, newblob.**length**);  **iv002**.setImageBitmap(**bitmap**);   **etName**.setText(**c**.getString(**nameColIndex**));  **txtTime**.setText(**c**.getString(**dateColIndex**) + **" "** + **c**.getString(**timeColIndex**));  **txtDesc**.setText(**c**.getString(**descColIndex**));    **break**;  }  }   *//Класс для управления БД SQLite* **public class** DBHelper **extends** SQLiteOpenHelper {   **private** SQLiteDatabase **mDataBase**;  **private final** Context **mContext**;  **private boolean mNeedUpdate** = **false**;   *//Конструктор класса DBHelper* **public** DBHelper(Context context) {  **super**(context, *DB\_NAME*, **null**, ***DB\_VERSION***);  **if** (android.os.Build.VERSION.***SDK\_INT*** >= 17)  *DB\_PATH* = context.getApplicationInfo().**dataDir** + **"/databases/"**;  **else** *DB\_PATH* = **"/data/data/"** + context.getPackageName() + **"/databases/"**;  **this**.**mContext** = context;   copyDataBase();   **this**.getReadableDatabase();  }   *//Процедура обновления БД (срабатывает при ошибке ввада-вывода)* **public void** updateDataBase() **throws** IOException {  **if** (**mNeedUpdate**) {  File dbFile = **new** File(*DB\_PATH* + *DB\_NAME*);  **if** (dbFile.exists())  dbFile.delete();   copyDataBase();   **mNeedUpdate** = **false**;  }  }   *//Процедура проверки наличия файла БД в каталоге приложения* **private boolean** checkDataBase() {  File dbFile = **new** File(*DB\_PATH* + *DB\_NAME*);  **return** dbFile.exists();  }   *//Процедура создания файла БД в каталоге приложения* **private void** copyDataBase() {  *//Если файла БД НЕ существует, то скопировать его из ресурсов* **if** (!checkDataBase()) {  **this**.getReadableDatabase();  **this**.close();  **try** {  copyDBFile();  } **catch** (IOException mIOException) {  **throw new** Error(**"ErrorCopyingDataBase"**);  }  }  }   *//Процедура копирования файла из папки Assets в папку приложения* **private void** copyDBFile() **throws** IOException {  InputStream mInput = **mContext**.getAssets().open(*DB\_NAME*); *//InputStream mInput = mContext.getResources().openRawResource(R.raw.info);* OutputStream mOutput = **new** FileOutputStream(*DB\_PATH* + *DB\_NAME*);  **byte**[] mBuffer = **new byte**[1024];  **int** mLength;  **while** ((mLength = mInput.read(mBuffer)) > 0)  mOutput.write(mBuffer, 0, mLength);  mOutput.flush();  mOutput.close();  mInput.close();  }   *//Процедура открытия БД* **public boolean** openDataBase() **throws** SQLException {  **mDataBase** = SQLiteDatabase.*openDatabase*(*DB\_PATH* + *DB\_NAME*, **null**, SQLiteDatabase.***CREATE\_IF\_NECESSARY***);  **return mDataBase** != **null**;  }   @Override  **public synchronized void** close() {  **if** (**mDataBase** != **null**)  **mDataBase**.close();  **super**.close();  }   @Override  **public void** onCreate(SQLiteDatabase db) {   }   *//Процедура обновления БД при несовпадении версий* @Override  **public void** onUpgrade(SQLiteDatabase db, **int** oldVersion, **int** newVersion) {  **if** (newVersion > oldVersion)  **mNeedUpdate** = **true**;  }  } } |

Листинг 1.2 – Текст файла activity\_main.xml

|  |
| --- |
| *<?***xml version="1.0" encoding="utf-8"***?>* <**LinearLayout  xmlns:android="http://schemas.android.com/apk/res/android"  xmlns:app="http://schemas.android.com/apk/res-auto"  android:layout\_width="fill\_parent"  android:layout\_height="fill\_parent"  android:orientation="vertical"**>   <**LinearLayout  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:orientation="horizontal"**>   <**ImageView  android:id="@+id/iV001"  android:layout\_width="200dp"  android:layout\_height="200dp"  android:layout\_marginLeft="90dp"  android:src="@mipmap/ic\_launcher"**/>  </**LinearLayout**>   <**LinearLayout  android:id="@+id/linearLayout1"  android:layout\_width="match\_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="Название"**></**TextView**>  <**EditText  android:id="@+id/etName"  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="match\_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="Описание"**></**TextView**>  <**EditText  android:id="@+id/txtDesc"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_weight="1"**>  </**EditText**>  </**LinearLayout**>  <**LinearLayout  android:id="@+id/linearLayout5"  android:layout\_width="match\_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="Время"**></**TextView**>  <**EditText  android:id="@+id/txtTime"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_weight="1"**>  </**EditText**>  </**LinearLayout**>  <**LinearLayout  android:id="@+id/linearLayout6"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"**>   <**ImageView  android:id="@+id/iV002"  android:layout\_width="200dp"  android:layout\_height="200dp"  android:layout\_marginLeft="90dp"  android:src="@mipmap/ic\_launcher"**/>   </**LinearLayout**>   <**LinearLayout  android:id="@+id/linearLayout2"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"**>  <**Button  android:id="@+id/btnAdd"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:text="Прошлое"**>  </**Button**>  <**Button  android:id="@+id/btnRead"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:text="Следующее"**>  </**Button**>   </**LinearLayout**>  </**LinearLayout**> |

## Тестирование программы

На рисунках 1-2 показаны данные полученные из базы данных dbdemos.db.

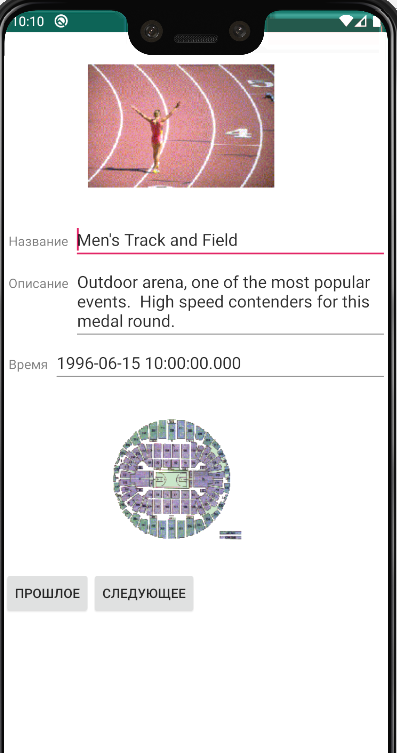


Рисунок 1 – Данные из базы данных dbdemos.db

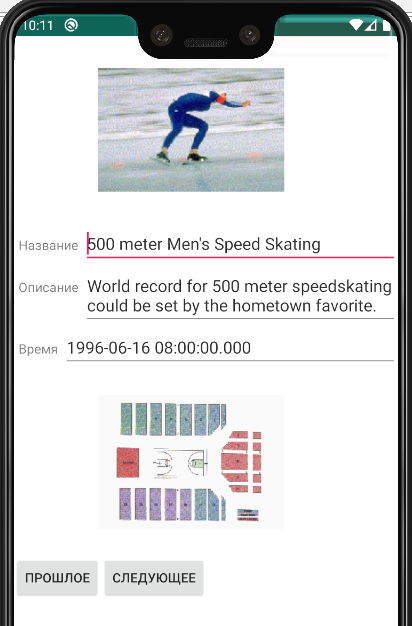


Рисунок 2 – Данные из базы данных dbdemos.db

Список использованных источников

1 Соколова, В. В. Разработка мобильных приложений [Электронный ресурс] : учеб. пособие / Соколова В.В. - Томск: Изд-во Томского политех. университета, 2014. - 176 с. // ZNANIUM.COM : электронно-библиотечная система. – Режим доступа: http://znanium.com/catalog.php#, ограниченный. – Загл. с экрана.

2 Тихомиров, В.А. Разработка простейших приложений для мобильных устройств: Учебное пособие / В. А. Тихомиров. - Комсомольск-наАмуре: Изд-во Комсомольского-на-Амуре гос.техн.ун-та, 2013. – 133 с.