Создаем базу данных SQLite Database

|  |
| --- |
| Сегодня мы сделаем приложение, которое с помощью использования базы данных SQLite Database будет способно сохранять введенные в него данные и выдавать их пользователю при запуске приложения. Мы сделаем более упрощенное приложение и здесь пока не будет возможности для пользователя вводить данные в базу, данные мы введем сами в коде класса MainActivity и они будут храниться и отображаться базой данных при запущенном приложении.  Не охота влезать в теорию, поэтому если вы хотите постигнуть всю теорию основ работы с базами данных в Android, то поищите ее на других ресурсах, либо ждите, пока она появится здесь :). Мы же просто сделаем несложный пример занесения данных в базу SQLite Database и их чтение и отображение пользователю.  [Создаем](http://learn-android.ru/news/rabochaja_sreda_android_studio_i_hellow_development_world/2015-01-03-5.html) новый проект, выбираем Blank Activity. Начнем с создание пользовательского интерфейса приложения. Он будет состоять из двух текстовых полей, одно с названием приложения, другое будет использоваться для вывода данных с базы данных. Открываем файл **activity\_main.xml** и добавляем туда следующий код:  <?xml version="1.0" encoding="utf-8"?>  <**LinearLayout** xmlns:android="http://schemas.android.com/apk/res/android"  android:orientation="vertical"  android:layout\_width="fill\_parent"  android:layout\_height="fill\_parent">  <**TextView**  android:layout\_width="fill\_parent"  android:layout\_height="wrap\_content"  android:text="Пример работы Android SQLiteDatabase"  android:textSize="12pt"/>  <**TextView**  android:id="@+id/content\_list"  android:layout\_width="fill\_parent"  android:layout\_height="wrap\_content"  android:textSize="8pt"/>    </**LinearLayout**>  Теперь перейдем к работе в классе **MainActivity.java**. Здесь мы объявляем объект класса SQLiteAdapter, который нужен для связи activity приложения с базой данных. Мы выполняем с ним следующие действия: открываем для записи, удаляем все, что там уже есть, с помощью команды insert выполняем запись данных и закрываем. После этого, открываем ту же базу уже для чтения, берем с нее все данные и отображаем в TextView:  import **android.app.Activity**;  import **android.os.Bundle**;  import **android.widget.TextView**;  **public** **class** MainActivity **extends** Activity {  **private** **SQLiteAdapter** mySQLiteAdapter;  **@Override**  **public** **void** onCreate(**Bundle** *savedInstanceState*) {  super.onCreate(savedInstanceState);  setContentView(**R**.layout.activity\_main);  **TextView** listContent = (**TextView**)findViewById(**R**.id.content\_list);  *// Создаем/открываем SQLite базу данных для записи, удаляем все ее содержимое,*  *// заполняем указанным содержимым с помощью команды insert, и закрываем:*  mySQLiteAdapter = new **SQLiteAdapter**(this);  mySQLiteAdapter.openToWrite();  mySQLiteAdapter.deleteAll();  mySQLiteAdapter.insert("Привет");  mySQLiteAdapter.insert("вот так");  mySQLiteAdapter.insert("мы ");  mySQLiteAdapter.insert("храним текст");  mySQLiteAdapter.insert("в базе данных");  mySQLiteAdapter.close();  *// Открываем ту же базу данных для чтения ее содержимого:*  mySQLiteAdapter = new **SQLiteAdapter**(this);  mySQLiteAdapter.openToRead();  *//Вытаскиваем всю информацию в качестве строковых ресурсов:*  **String** contentRead = mySQLiteAdapter.queueAll();  mySQLiteAdapter.close();  *// Отображаем прочитанные данные в TextView:*  listContent.setText(contentRead);  }  }  Работа в этом классе закончена. Но пока что у нас нет самого главного - адаптера SQLiteAdapter.  Создаем новый класс под названием **SQLiteAdapter**. Этот файл мы заполняем следующим образом:  import **android.content.ContentValues**;  import **android.content.Context**;  import **android.database.Cursor**;  import **android.database.sqlite.SQLiteDatabase**;  import **android.database.sqlite.SQLiteOpenHelper**;  import **android.database.sqlite.SQLiteDatabase.CursorFactory**;  **public** **class** SQLiteAdapter {  *// Вводим используемые значения и переменные:*  **public** **static** **final** **String** DATABASE\_NAME = "MY\_DATABASE";  **public** **static** **final** **String** DATABASE\_TABLE = "MY\_TABLE";  **public** **static** **final** **int** DATABASE\_VERSION = 1;  **public** **static** **final** **String** KEY\_CONTENT = "Content";  *// Создаем таблицу с данными MY\_DATABASE:*  **private** **static** **final** **String** SCRIPT\_CREATE\_DATABASE = DATABASE\_TABLE + KEY\_CONTENT;  **private** **SQLiteHelper** sqLiteHelper;  **private** **SQLiteDatabase** sqLiteDatabase;  **private** **Context** context;  **public** SQLiteAdapter(**Context** *c*){context = c;  }  *// Читаем данные с базы данных:*  **public** **SQLiteAdapter** openToRead() **throws** **android.database.SQLException** {  sqLiteHelper = new **SQLiteHelper**(context, DATABASE\_NAME, null, DATABASE\_VERSION);  sqLiteDatabase = sqLiteHelper.getReadableDatabase();  return this;  }  *// Записываем данные в базу данных:*  **public** **SQLiteAdapter** openToWrite() **throws** **android.database.SQLException** {  sqLiteHelper = new **SQLiteHelper**(context, DATABASE\_NAME, null, DATABASE\_VERSION);  sqLiteDatabase = sqLiteHelper.getWritableDatabase();  return this;  }  *// Закрываем sqLiteHelper:*  **public** **void** close(){sqLiteHelper.close();  }  *// Вставляем введенное содержимое в базу:*  **public** **long** insert(**String** *content*){  **ContentValues** contentValues = new **ContentValues**();  contentValues.put(KEY\_CONTENT, content);  return sqLiteDatabase.insert(DATABASE\_TABLE, null, contentValues);  }  *// Удаляем все содержимое базы данных:*  **public** **int** deleteAll(){return sqLiteDatabase.delete(DATABASE\_TABLE, null, null);  }  *// Делаем запрос на получение текстовых данных с базы, отображаемых в виде столбца:*  **public** **String** queueAll(){  **String**[] columns = new **String**[]{KEY\_CONTENT};  **Cursor** cursor = sqLiteDatabase.query(DATABASE\_TABLE, columns,null, null, null, null, null);  **String** result = "";  **int** index\_CONTENT = cursor.getColumnIndex(KEY\_CONTENT);  for(cursor.moveToFirst(); !(cursor.isAfterLast()); cursor.moveToNext()){  result = result + cursor.getString(index\_CONTENT) + "\n";  }  return result;  }  **public** **class** SQLiteHelper **extends** SQLiteOpenHelper {  **public** SQLiteHelper(**Context** *context*, **String** *name*,**CursorFactory** *factory*, **int** *version*) {  super(context, name, factory, version);  }  *// Метод, создающий таблицу для хранения данных:*  **@Override**  **public** **void** onCreate(**SQLiteDatabase** *db*) {db.execSQL(SCRIPT\_CREATE\_DATABASE);  }  *// Метод для обновления базы данных, оставим пустым:*  **@Override**  **public** **void** onUpgrade(**SQLiteDatabase** *db*, **int** *oldVersion*, **int** *newVersion*) {  }  }  }  Немного разберемся, что же тут понаписано. Все это можно разделить на такие основные блоки:  - ввод используемых переменных;  - **onCreate ()** - создание базы;  - **onUpgrade ()** - обновление базы в случае надобности. Мы задаем версию базы:  **public** **static** **final** **int** DATABASE\_VERSION = 1;  это значит, что в случае смены версии базы заработает метод обновления для обновления ее содержимого;  - метод **openToWrite ()** - открывает базу для записи данных;  - метод **openToRead ()** - выполняет чтение данных;  - метод **close**- закрываем базу;  - метод **insert**- вводим в базу данные в формате "ключ - значение"  - **deleteAll()**- удаляем все данные с базы;  Ну что же, запустим наше приложение и посмотрим на результат:  Работает база данных  Внесение и чтение данных в базу данных SQLite Database произведено успешно!  Также вспомните работу с хранением настроек приложения с помощью [SharedPreferences](http://learn-android.ru/news/sokhranjaem_nastrojki_prilozhenija_s_pomoshhju_shared_preferences/2015-04-28-101.html). |