

Федеральное государственное бюджетное образовательное учреждение высшего образования
«Сибирский государственный университет телекоммуникаций и информатики»
(СибГУТИ)

Кафедра прикладной математики и кибернетики

Лабораторная работа «База данных»
по дисциплине «Программирование для мобильных устройств»

Выполнил:

студент группы ИП-811

Разумов Д.Б.

Работу проверил:

Павлова У.В.

Новосибирск 2021

Содержание

Содержание.....	2
Задание.....	3
Листинг программы.....	4
Файл MainActivity.java.....	4
Файл MyConstants.java.....	6
Файл MyDbHelper.java.....	7
Файл MyDbManager.java.....	7
Файл activity_main.xml.....	10
Скриншоты работы программы.....	12

Задание

Создать базу данных студентов (Имя, вес, рост, возраст - сгенерировать случайно). Вывести из базы данных все записи, отсортированные по возрасту, в таблицу (TableLayout).

Листинг программы

Файл MainActivity.java

```
package com.example.lab_sql;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;
import android.view.View;
import android.widget.TableLayout;
import android.widget.TableRow;
import android.widget.TextView;

import com.example.lab_sql.database.MyDbManager;

import java.util.ArrayList;
import java.util.Vector;

public class MainActivity extends AppCompatActivity {
    private MyDbManager myDbManager;
    private TableLayout tableForDb;
    ArrayList<TableRow> rows = new ArrayList<>(); //массив для строк в TableLayout

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        myDbManager = new MyDbManager(this);
        tableForDb = findViewById(R.id.tableForDb);
        addRowInTableForDb("Фамилия", "вес", "рост", "возраст", 0);
//верхняя строка в TableLayout
    }

    @Override
    protected void onResume() {
        super.onResume();
        myDbManager.openDb();
    }

    @Override
    protected void onDestroy() {
        myDbManager.closeDb();
        super.onDestroy();
    }

    public void onClickAdd(View view) {
        myDbManager.insertToDb();
    }

    public void onClickShowDb(View view) {
//функция для показа БД в таблице TableLayout
    }
```

```

        for (int i = rows.size() - 1; i >= 1; i--) { //удалить все
            строки кроме самой верхней
            TableRow row = rows.get(i);
            tableForDb.removeView(row);
            rows.remove(i);
        }

        Vector<String> whole_table = myDbManager.readSortedDb();
        int ind_row = 1; //индекс строки
        for (int i = 0; i < whole_table.size(); i += 4) {
            addRowInTableForDb(whole_table.get(i),
                               whole_table.get(i+1),
                               whole_table.get(i+2),
                               whole_table.get(i+3),
                               ind_row++);
        }
    }
    public void onClickClear(View view) {
        myDbManager.clearDb();
    }

    //вспомогательная функция для добавления строки в таблицу
    TableLayout
    private void addRowInTableForDb(String s1, String s2, String s3,
    String s4, int ind) {
        TableRow tableRow = new TableRow(this); //создать строку для
        TableLayout
        rows.add(tableRow); //записать адрес (?) этой строки в rows[]
        tableRow.setLayoutParams(new TableRow.LayoutParams(
            TableRow.LayoutParams.MATCH_PARENT,
            TableRow.LayoutParams.MATCH_PARENT));
        for (int j = 0; j < 4; j++) { //создать 4 TextView - это как
            бы ячейки в строке
            TextView textView = new TextView(this);
            switch (j) { //установить ширину ячеек и задать текст
                TextView
                case 0:
                    textView.setText(s1);
                    textView.setWidth(300);
                    break;
                case 1:
                    textView.setText(s2);
                    textView.setWidth(100);
                    break;
                case 2:
                    textView.setText(s3);
                    textView.setWidth(150);
                    break;
                case 3:
                    textView.setText(s4);
                    textView.setWidth(200);
            }
        }
    }

```

```

        break;
    }
    tableRow.addView(textView, j); //добавить textView[j] в
    строку
    }
    tableForDb.addView(tableRow, ind); //добавить строку в таблицу
    TableLayout
    }
}

```

Файл MyConstants.java

```
package com.example.lab_sql.database;
```

```

public class MyConstants {
    //это класс для строковых констант, которые задают те или иные
    названия в базе данных
    public static final String DATABASE_NAME = "my_database.db";
    //название базы данных
    public static final int DATABASE_VERSION = 2; //версия базы данных
    - при каждом изменении в БД
    // надо менять версию
    public static final String TABLE_NAME = "table_students";
    //название таблицы
    public static final String S_ID = "s_id"; //1-ая колонка в таблице
    = айди
    public static final String S_NAME = "s_name"; //2-ая колонка в
    таблице = имя студента
    public static final String S_WEIGHT = "s_weight"; //3 колонка в
    таблице = вес
    public static final String S_HEIGHT = "s_height"; //4 колонка в
    таблице = рост
    public static final String S_AGE = "s_age"; //5 колонка в таблице
    = возраст
    // ниже команда, которая создает таблицу в БД (если таблицы с
    таким именем еще нет)
    public static final String CREATE_TABLE = "CREATE TABLE IF NOT
    EXISTS " +
        TABLE_NAME + " (" +
        S_ID + " INTEGER PRIMARY KEY," +
        S_NAME + " TEXT," +
        S_WEIGHT + " INTEGER," +
        S_HEIGHT + " INTEGER," +
        S_AGE + " INTEGER)";
    // удаление (drop) таблицы:
    public static final String DELETE_TABLE = "DROP TABLE IF EXISTS "
    + TABLE_NAME;
}

```

Файл MyDbHelper.java

```
package com.example.lab_sql.database;

import android.content.Context;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;

import androidx.annotation.Nullable;

public class MyDbHelper extends SQLiteOpenHelper {
    //вспомогательный класс, который наследуется от SQLiteOpenHelper и
    //меняет некоторые методы
    public MyDbHelper(@Nullable Context context) { //конструктор
        //класса
        super(context, MyConstants.DATABASE_NAME, null,
            MyConstants.DATABASE_VERSION);
        // "super" для вызова конструктора родительского класса с
        //переданными в скобках параметрами
    }

    @Override
    public void onCreate(SQLiteDatabase db) {
        db.execSQL(MyConstants.CREATE_TABLE);
    }

    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int
        newVersion) {
        //если новая версия таблицы, то удалить базу данных и создать
        //заново
        db.execSQL(MyConstants.DELETE_TABLE);
        onCreate(db);
    }
}
```

Файл MyDbManager.java

```
package com.example.lab_sql.database;

import android.content.ContentValues;
import android.content.Context;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;

import java.util.Random;
import java.util.Vector;

//класс, для открытия и добавления записей в базу данных
public class MyDbManager {
```

//класс, методы которого вызываются в MainActivity, тут все нужные функции для работы с БД

private Context **context**;
private MyDbHelper **myDbHelper**; *//использовать класс, который мы унаследовали от SQLiteOpenHelper*

private SQLiteDatabase **sqlite_db**; *//база данных*

public MyDbManager(Context context) {
 this.context = context;
 myDbHelper = **new** MyDbHelper(context);
}

public void openDb() {
 //открыть базу данных для чтения и записи:
 sqlite_db = **myDbHelper**.getWritableDatabase();
}

public void insertToDb() {
 //для добавления записи в БД используются случайные значения
 Random rand = **new** Random();
 int n = rand.nextInt(5);
 String name; *//случайное имя студента*
 switch (n) {
 case 0:
 name = "Иванов";
 break;
 case 1:
 name = "Иванова";
 break;
 case 2:
 name = "Вассерман";
 break;
 case 3:
 name = "Канделаки";
 break;
 case 4:
 name = "Якубович";
 break;
 default:
 name = "ошибка в insertToDb()";
 break;
 }

int weight = rand.nextInt(100); *//случайный вес от 50 до 150*
 weight += 50;
 int height = rand.nextInt(50); *//случайный рост от 150 до 200*
 height += 150;
 int age = rand.nextInt(10); *//случайный возраст от 18 до 28*
 age += 18;
 //нужно создать объект класса ContentValues, записать в него значения

//и передать этот объект для добавления записи в БД
 ContentValues cv = **new** ContentValues();


```

        cv.put(MyConstants.S_NAME, name); //put(ключ, значение)
        cv.put(MyConstants.S_WEIGHT, weight);
        cv.put(MyConstants.S_HEIGHT, height);
        cv.put(MyConstants.S_AGE, age);
        //выбрать в базе данных таблицу и добавить туда созданный
ContentValues:
        sqlite_db.insert(MyConstants.TABLE_NAME, null, cv);
    }
    public Vector<String> readSortedDb() {
        //данный метод делает запрос к базе данных, в котором берутся
все значения таблицы
        //в отсортированном (по возрасту) порядке
        Vector<String> returnVector = new Vector<>(); //этот массив
будет возвращаться
        //Cursor - класс для считывания из БД. Делаем запрос:
        Cursor cursor = sqlite_db.rawQuery("SELECT * FROM " +
MyConstants.TABLE_NAME +
        " ORDER BY " + MyConstants.S_AGE, null);
        //в цикле ниже в массив добавляются имя, вес, рост, возраст
каждой записи
        while (cursor.moveToNext()) {
            String s =
cursor.getString(cursor.getColumnIndex(MyConstants.S_NAME));
            returnVector.add(s);
            s =
cursor.getString(cursor.getColumnIndex(MyConstants.S_WEIGHT));
            returnVector.add(s);
            s =
cursor.getString(cursor.getColumnIndex(MyConstants.S_HEIGHT));
            returnVector.add(s);
            s =
cursor.getString(cursor.getColumnIndex(MyConstants.S_AGE));
            returnVector.add(s);
        }
        cursor.close();
        return returnVector;
    }
    public void closeDb() {
        myDbHelper.close();
    }
    public void clearDb() {
        //очистить базу данных
        myDbHelper.onUpgrade(sqlite_db, -1, -1);
    }
}

```

Файл *activity_main.xml*

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <Button
        android:id="@+id/buttonAdd"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="16dp"
        android:onClick="onClickAdd"
        android:text="Добавить запись"
        app:layout_constraintTop_toTopOf="parent"
        tools:layout_editor_absoluteX="16dp" />

    <Button
        android:id="@+id/buttonShowDb"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:onClick="onClickShowDb"
        android:text="Вывести содержимое базы данных"
        app:layout_constraintTop_toBottomOf="@+id/buttonAdd"
        tools:layout_editor_absoluteX="16dp" />

    <TableLayout
        android:id="@+id/tableForDb"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:padding="10dp"
        app:layout_constraintTop_toBottomOf="@+id/buttonClear"
        android:textColor="#FF000000"
        tools:layout_editor_absoluteX="129dp">

        <!-- <TableRow-->
        <!--         android:layout_width="match_parent"-->
        <!--         android:layout_height="match_parent"-->
        <!--         android:textColor="@color/black" />-->

        <!-- <TableRow-->
        <!--         android:layout_width="match_parent"-->
        <!--         android:layout_height="match_parent"-->
        <!--         android:textColor="@color/black" />-->

        <!-- <TableRow-->
        <!--         android:layout_width="match_parent"-->
```

```

<!--          android:layout_height="match_parent"-->
<!--          android:textColor="@color/black" />-->

<!--      <TableRow-->
<!--          android:layout_width="match_parent"-->
<!--          android:layout_height="match_parent"-->
<!--          android:textColor="#000000" />-->

</TableLayout>

<Button
    android:id="@+id/buttonClear"
    android:layout_width="237dp"
    android:layout_height="44dp"
    android:onClick="onClickClear"
    android:text="Очистить базу данных"
    app:layout_constraintTop_toBottomOf="@+id/buttonShowDb"
    tools:layout_editor_absoluteX="19dp" />
</androidx.constraintlayout.widget.ConstraintLayout>

```

Скриншоты работы программы



Рисунок 1