ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«СИБИРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ТЕЛЕКОММУНИКАЦИЙ И ИНФОРМАТИКИ»

Лабораторная работа №5

«База данных»

Выполнили: студенты группы ИП-715

Кузнецов Е.В.

Винтер А.В.

Работу принял: ассистент кафедры ПМиК

Павлова У.В.

Новосибирск 2020

**Оглавление**

Цель работы…..................................................................................................3

Описание программы и некоторых функций…............................................4

Листинг программы….....................................................................................6

Приложение…..................................................................................................9

**Цель работы**

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

**Описание программы и некоторых функций.**

Запустив приложение, пользователь видит базу данных, в которой содержатся следующие пункты: 1) идентификатор 2) Имя 3) Вес 4) Рост 5) Возраст.

Вывод элементов отсортирован по возрасту студента в таблицу (TableLayout).

public void onCreate(SQLiteDatabase db) - создадим новую таблицу

public void onUpgrade(SQLiteDatabase database, int oldVersion, int newVersion) – перезапишем ее в старую.

db = dbHelper.getWritableDatabase(); - открываем базу данных

if(c.moveToFirst()) - переходим на первый элемент если он есть

int index = c.getColumnIndex("\_id"); - получаем номер колонки с имени id

Id\_label.setText(c.getString(index)); - выводим строковое значение по номеру колонки

while(c.moveToNext()); - переходим к следующему элементу.

Сортировка происходит через запрос:

SELECT \* FROM \_\_\_\_

ORDER BY \_\_\_\_ DESC

В нашем случае:

Cursor c; (это указатель на контекстную область памяти, с помощью которого программа на языке SQL может управлять контекстной областью и ее состоянием во время обработки оператора)

c = db.rawQuery("SELECT \* FROM StudTable ORDER BY age DESC", null); - выполняется запрос из БД

Заполнение начальных данных таблицы:

В методе **onCreate()** необходимо реализовать логику создания таблиц и при необходимости заполнить их начальными данными при помощи SQL-команды.

Начальные данные:

Имена: String[] itemname ={  
 **"Антон"**,  
 **"Егор"**,  
 **"Паша"**,  
 **"Тимур"**,  
 **"Дмитрий"**,  
 **"Федор"**,  
 **"Марк"**,  
 **"Евгений"**,  
 **"Максим"**};

**for**(**int** i = 0; i < 20; ++i) - 20 студентов в базе данных

Используем рандомные значения для возраста, веса и роста учитывая реальные значения: ( например, меньше 17 лет для студента мы не получим)

Random rand = **new** Random();  
  
Integer randInt = rand.nextInt(10) + 17 ;  
String st\_age = randInt.toString();  
  
randInt = rand.nextInt(40) + 70;  
String st\_weight = randInt.toString();  
  
randInt = rand.nextInt(100) + 110;  
String st\_height = randInt.toString();

**Листинг программы**

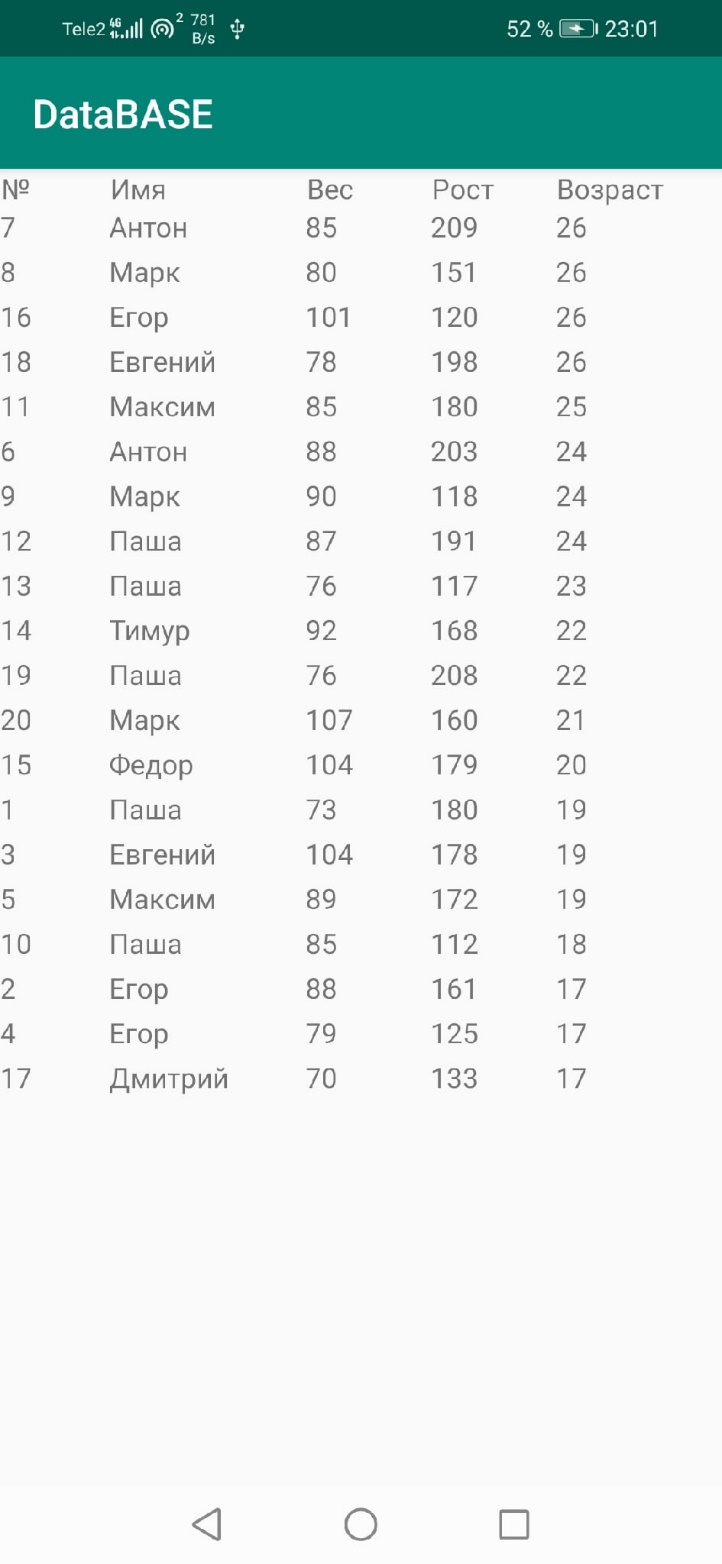
MainActivity

**package** ru.example.database;  
  
**import** android.database.Cursor;  
**import** android.database.sqlite.SQLiteDatabase;  
**import** android.support.v7.app.AppCompatActivity;  
**import** android.os.Bundle;  
**import** android.widget.TableLayout;  
**import** android.widget.TableRow;  
**import** android.widget.TextView;  
  
**public class** MainActivity **extends** AppCompatActivity {  
  
 **public static final int *colNum*** = 5;  
  
 **private** SQLiteDatabase **db**;  
 **private** StudentsDB **dbHelper**;  
  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_main***);  
 **dbHelper** = **new** StudentsDB(MainActivity.**this**);  
 **db** = **dbHelper**.getWritableDatabase();  
  
 TableLayout table = findViewById(R.id.***bd\_table***);  
  
 TableRow tablehead = **new** TableRow(**this**);  
 tablehead.setPadding(2, 0, 10, 0);  
  
 TextView htv1 = **new** TextView(**this**);  
 htv1.setText(**"№"**);  
 tablehead.addView(htv1);  
  
 TextView htv2 = **new** TextView(**this**);  
 htv2.setText(**"Имя"**);  
 tablehead.addView(htv2);  
  
 TextView htv3 = **new** TextView(**this**);  
 htv3.setText(**"Вес"**);  
 tablehead.addView(htv3);  
  
 TextView htv4 = **new** TextView(**this**);  
 htv4.setText(**"Рост"**);  
 tablehead.addView(htv4);  
  
 TextView htv5 = **new** TextView(**this**);  
 htv5.setText(**"Возраст"**);  
 tablehead.addView(htv5);  
 table.addView(tablehead);  
  
 Cursor c;  
  
 c = **db**.rawQuery(**"SELECT \* FROM StudTable ORDER BY age DESC"**, **null**);  
  
 **if**(c.moveToFirst()) {  
 **do** {  
 TableRow tr = **new** TableRow(**this**);  
 tr.setPadding(2, 0, 10, 0);  
  
  
 **int** index = c.getColumnIndex(**"\_id"**);  
 TextView Id\_label = **new** TextView(**this**);  
 Id\_label.setPadding(0, 0 ,30 ,0);  
 Id\_label.setText(c.getString(index));  
 tr.addView(Id\_label);  
  
 index = c.getColumnIndex(**"name"**);  
 TextView Name\_label = **new** TextView(**this**);  
 Name\_label.setPadding(0, 0 ,30 ,0);  
 Name\_label.setText(c.getString(index));  
 tr.addView(Name\_label);  
  
 index = c.getColumnIndex(**"weight"**);  
 TextView Weight\_label = **new** TextView(**this**);  
 Weight\_label.setPadding(0, 0 ,30 ,0);  
 Weight\_label.setText(c.getString(index));  
 tr.addView(Weight\_label);  
  
 index = c.getColumnIndex(**"height"**);  
 TextView Height\_label = **new** TextView(**this**);  
 Height\_label.setPadding(0, 0 ,30 ,0);  
 Height\_label.setText(c.getString(index));  
 tr.addView(Height\_label);  
  
 index = c.getColumnIndex(**"age"**);  
 TextView Age\_label = **new** TextView(**this**);  
 Age\_label.setText(c.getString(index));  
 tr.addView(Age\_label);  
 tr.setPadding(0, 0, 0, 10);  
  
 table.addView(tr);  
  
 } **while**(c.moveToNext());  
 }  
 }  
}

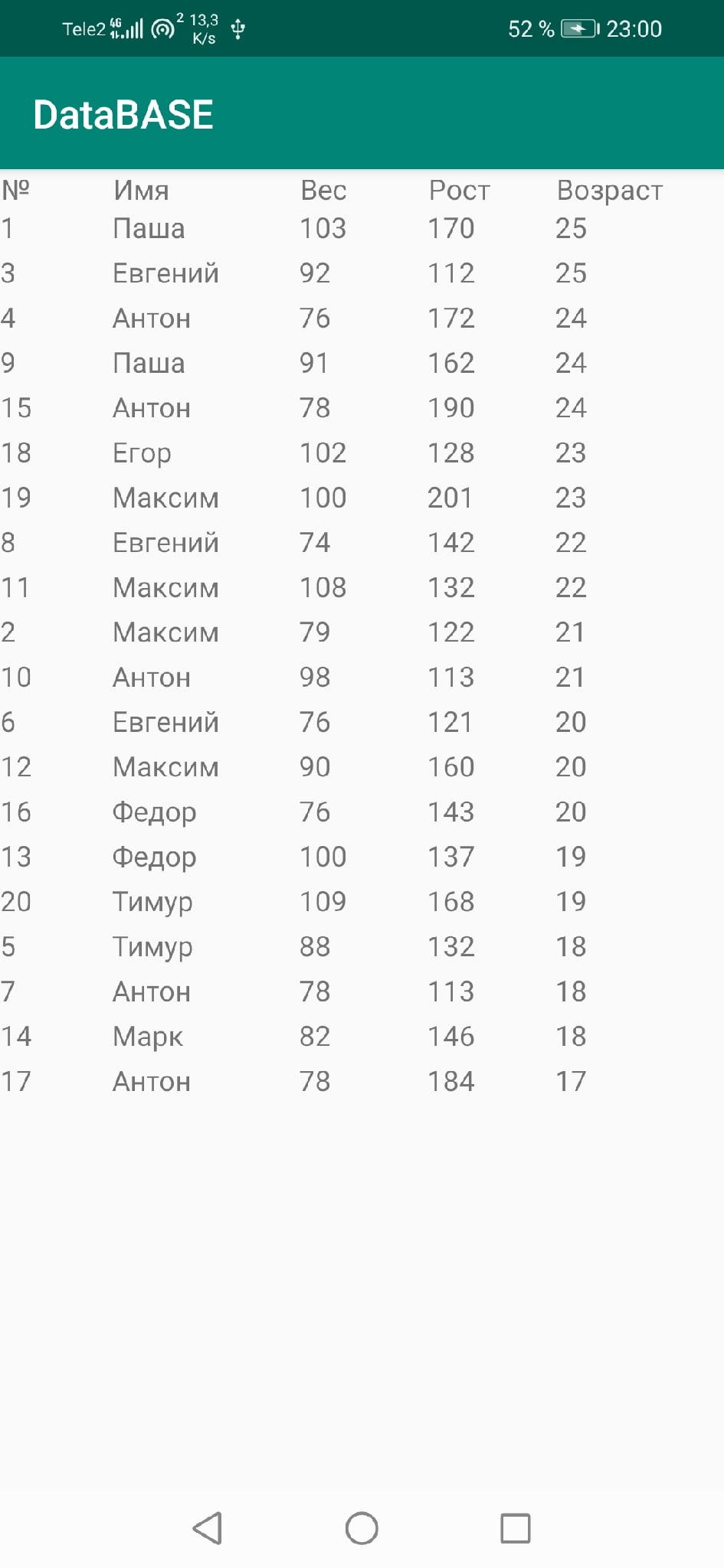
**StudentsDB**

**package** ru.example.database;  
  
**import** android.content.Context;  
**import** android.database.sqlite.SQLiteOpenHelper;  
**import** android.database.sqlite.SQLiteDatabase;  
**import** java.util.Random;  
  
**public class** StudentsDB **extends** SQLiteOpenHelper {  
  
 **public static final** String ***TABLE\_NAME*** = **"StudTable"**;  
  
 **public** StudentsDB(Context context){  
 **super**(context, ***TABLE\_NAME***,**null**,1);  
 }  
  
 **public void** onCreate(SQLiteDatabase db) {  
 String[] itemname ={  
 **"Антон"**,  
 **"Егор"**,  
 **"Паша"**,  
 **"Тимур"**,  
 **"Дмитрий"**,  
 **"Федор"**,  
 **"Марк"**,  
 **"Евгений"**,  
 **"Максим"** };  
  
  
  
 db.execSQL(**"CREATE TABLE "** + ***TABLE\_NAME*** + **" ("** +  
 **"\_id INTEGER PRIMARY KEY autoincrement, "** +  
 **"name TEXT, "** +  
 **"weight INTEGER, "** +  
 **"height INTEGER, "** +  
 **"age INTEGER );"**);  
  
 **for**(**int** i = 0; i < 20; ++i) {  
  
 String namepath = (itemname[**new** Random().nextInt(itemname.**length**)]);  
 String st\_name = namepath;  
  
  
 Random rand = **new** Random();  
  
 Integer randInt = rand.nextInt(10) + 17 ;  
 String st\_age = randInt.toString();  
  
 randInt = rand.nextInt(40) + 70;  
 String st\_weight = randInt.toString();  
  
 randInt = rand.nextInt(100) + 110;  
 String st\_height = randInt.toString();  
  
  
  
 db.execSQL(**"INSERT INTO StudTable ('name', 'weight', 'height', 'age') "** +  
 **"VALUES ('"**+ st\_name + **"', '"** + st\_weight + **"', '"** +  
 st\_height + **"', '"** + st\_age + **"')"**);  
  
 }  
 }  
  
 **public void** onUpgrade(SQLiteDatabase database, **int** oldVersion, **int** newVersion) {  
 database.execSQL(**"DROP TABLE IF EXISTS "** + ***TABLE\_NAME***);  
 database.execSQL(**"CREATE TABLE StudTable ("** +  
 **"\_id INTEGER PRIMARY KEY autoincrement, "** +  
 **"name TEXT, "** +  
 **"weight INTEGER, "** +  
 **"height INTEGER, "** +  
 **"age INTEGER );"**);  
 }  
}

**Приложение**



Запуск 1



Запуск 2