**Чарушин Ф.Д.,**

*студент 4 курса факультета естествознания, математики и информатики*

**Бужинская Н.В.,**

*доцент кафедры ИТ, к.п.н.*

*Нижнетагильский государственный социально-педагогический институт, (филиал) ФГБОУ ВО «Российский государственный профессионально-педагогический университет»*

*Россия, Нижний Тагил*

**Разработка мобильного приложения для хранения данных о результатах работы студентов**

**Аннотация**

 Мобильные приложения используются для решения самых разнообразных задач, в том числе и для автоматизации различных операций работника. В статье рассматривается процесс разработки мобильного приложения для хранения данных студентов, которые работают с курсовыми проектами. С помощью данного приложения можно быстро найти нужную информацию и вывести ее на экран. Особое внимание уделено разработке базы данных и ее подключению, поскольку именно она используется для хранения информации.

 **Ключевые слова:** мобильное приложение, база данных, таблица, Android, класс, Java, адаптер.

**Development of a mobile application for storing data on student performance results**

**Abstract**

Mobile applications are used to solve a wide variety of tasks, including automating various employee operations. The article discusses the process of developing a mobile application for storing data of students who work with course projects. Using this application, you can quickly find the information you need and display it on the screen. Particular attention is paid to the development of the database and its connection, since this is what is used to store information.

**Keywords:** mobile application, database, table, Android, class, Java, adapter.

 В настоящее время смарфон является важнейшей составляющей жизни человека. С помощью смарфтона люди узнают новости, общаются, организовывают свой досуг. Кроме того, смартфон может использоваться для организации работы в различных учреждениях. В данном исследовании представлен вариант использования мобильного приложения, разработанного для OS Android для хранения данных студентов в области их курсового проектирования.

 Постановка задачи: разработать мобильное приложение, в котором будет хранится информация о студентах, темах их проектов и ФИО руководителей.

 Требования к мобильному приложению:

- создать базу данных для хранения информации;

- вывести данные в виде таблицы;

- реализовать функцию добавления данных в таблицу.

Для начала создадим проект в Android Studio с пустой активностью и добавим кнопку, которая будет осуществлять переход на вторую активность. На другой активности будем выводить таблицу из базы данных. Напишем на кнопке текст «Продолжить». Также добавим в свойство у кнопки onClick = «onClickNextActivity» (рис. 1).



Рис. 1. Первая активность

Теперь создадим новую активность. Нажимаем «File» => «New» => «Empty Views Activity». Назовем новую активность «NextActivity».

Необходимо написать код для кнопки, с помощью которого нас будет отправлять на вторую активность. Для этого в файле «MainActivity.java» после функции «onCreate» добавим следующий код (листинг 1).

Листинг 1

Код для кнопки для перехода на следующую активность

public void onClickNextActivity(View view) {

Intent intent = new Intent(this, NextActivity.class);

startActivity(intent);}

Сделаем аналогичную кнопку на второй активности, которая будет возвращать нас обратно.

На следующем этапе необходимо создать новый Java класс, с помощью которого создадим базу данных. Для этого нажмем «File» => «New» => «Java Class» и назовем данный класс «MyDatabaseHelper». Добавим в него следующий код (листинг 2).

Листинг 2

Код для «MyDatabaseHelper»

package com.example.statya;

import android.content.Context;

import android.database.sqlite.SQLiteDatabase;

import android.database.sqlite.SQLiteOpenHelper;

import androidx.annotation.Nullable;

public class MyDatabaseHelper extends SQLiteOpenHelper {

 private Context context;

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

 private static final int DATABASE\_VERSION = 1;

 private static final String TABLE\_NAME = "kursachi";

 private static final String COLUMN\_ID = "\_id";

 private static final String COLUMN\_UCHENIK = "uchenik";

 private static final String COLUMN\_PREPOD = "prepod";

 private static final String COLUMN\_TEMA = "tema";

 private static final String COLUMN\_OCENKA = "ocenka";

 public MyDatabaseHelper(@Nullable Context context) {

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

 this.context = context;}

 @Override

 public void onCreate(SQLiteDatabase db) {

 String query =

 "CREATE TABLE " + TABLE\_NAME +

 " (" + COLUMN\_ID + " INTEGER PRIMARY KEY AUTOINCREMENT, " +

 COLUMN\_UCHENIK + " TEXT, " +

 COLUMN\_PREPOD + " TEXT, " +

 COLUMN\_TEMA + " TEXT, " +

 COLUMN\_OCENKA + " INTEGER);";

 db.execSQL(query);

 }

 @Override

 public void onUpgrade(SQLiteDatabase db, int i, int i1) {

 db.execSQL("DROP TABLE IF EXISTS " + TABLE\_NAME);

 onCreate(db);}}

База данных будет называться «Statya», а таблица внутри нее «Kursachi». Таблица будет иметь следующие поля:

* «\_id» ключевой, тип integer, автозаполяемое;
* «uchenik», тип string;
* «prepod» тип string;
* «tema», тип string;
* «ocenka», тип integer.

 Теперь добавим на нашей второй активности несколько объектов: TextView, FloatingActionButton и RecyclerView (рис. 2). Текст будет отображать смысл активности, кнопка будет необходима для добавления новых данных в таблицу, а объект FloatingActionButton будет отображать данные из таблицы.



Рис. 2. Вторая активность

После этого создадим третью активность, где мы будет заполнять данные для таблицы. Назовем новую активность «AddActivity». Также добавим код для кнопки со второй активности для перехода на третью активность.

На третьей активности добавим четыре объекта Plait Text, в которых будем вводить данные для добавления. Также добавим 2 кнопки: «Назад» и «Добавить» (рис. 3).



Рис. 3. Активность для добавления данных в таблицу

 Кнопка «Назад» будет возвращать на вторую активность. Кнопка «Добавить» будет добавлять данные в таблице. Также в файле «AddActivity.java» необходимо инициализировать строки, в которые будут записываться значения (листинг 3).

Листинг 3

Инициализация объектов

public class AddActivity extends AppCompatActivity {

 EditText uchenik\_input, prepod\_input, tema\_input, ocenka\_input;

 @Override

 protected void onCreate(Bundle savedInstanceState) {

 super.onCreate(savedInstanceState);

 setContentView(R.layout.activity\_add);

 uchenik\_input = findViewById(R.id.uchenik\_input);

 prepod\_input = findViewById(R.id.prepod\_input);

 tema\_input = findViewById(R.id.tema\_input);

 ocenka\_input = findViewById(R.id.ocenka\_input);

 }

…

}

Код для кнопки «Добавить» будет выглядеть следующим образом (листинг 4):

Листинг 4

Код для кнопки «Добавить»

public void onClickAdd(View view) {

 MyDatabaseHelper myDB = new MyDatabaseHelper(AddActivity.this);

 myDB.addKursach(uchenik\_input.getText().toString().trim(),

 prepod\_input.getText().toString().trim(),

 tema\_input.getText().toString().trim(),

 Integer.valueOf(ocenka\_input.getText().toString().trim()));

}

Кнопка будет создавать экземпляр созданного класса и использовать метод для добавления значений в таблицу. Также необходимо добавить новый код в файл «MyDatabaseHelper.java» (листинг 5).

Листинг 5

Метод «addKursach»

void addKursach(String uchenik, String prepod, String tema, int ocenka) {

 SQLiteDatabase db = this.getWritableDatabase();

 ContentValues cv = new ContentValues();

 cv.put(COLUMN\_UCHENIK, uchenik);

 cv.put(COLUMN\_PREPOD, prepod);

 cv.put(COLUMN\_TEMA, tema);

 cv.put(COLUMN\_OCENKA, ocenka);

 long result = db.insert(TABLE\_NAME, null, cv);

 if (result == -1){

 Toast.makeText(context, "Не удалось добавить данные", Toast.LENGTH\_SHORT).show();

 } else {

 Toast.makeText(context, "Успешно добавлено!", Toast.LENGTH\_SHORT).show();

 }}

Теперь данные записываются в таблицу, но на экране ничего не отображается. Для решения данной проблемы, необходимо добавить метод в класс, который будет считывать информацию из таблицы (листинг 6).

Листинг 6

Метод «readAllData»

Cursor readAllData(){

 String query = "SELECT \* FROM " + TABLE\_NAME;

 SQLiteDatabase db = this.getReadableDatabase();

 Cursor cursor = null;

 if (db != null){

 cursor = db.rawQuery(query, null);

 }

 return cursor; }

Теперь необходимо внести изменения в файл «NextActivity.java». Объявим переменные MyDatabaseHelper myDB; ArrayList<String> id, uchenik, prepod,tema, ocenka. Также инициализируем их в методе onCreate (листинг 7) и вызовем функцию «storeDataInArray», которая будет записывать данные из таблицы в массив.

Листинг 7

Инициализация новых переменных для вывода информации из таблицы

myDB = new MyDatabaseHelper(NextActivity.this);

id = new ArrayList<>();

uchenik = new ArrayList<>();

prepod = new ArrayList<>();

tema = new ArrayList<>();

ocenka = new ArrayList<>();

storeDataInArray();

Код для метода «storeDataInArray» будет выглядеть следующим образом (листинг 8).

Листинг 8

Метод «storeDataInArray»

void storeDataInArray(){

 Cursor cursor = myDB.readAllData();

 if (cursor.getCount() == 0){

 Toast.makeText(this, "Нет данных", Toast.LENGTH\_SHORT).show();

 } else {

 while (cursor.moveToNext()){

 id.add(cursor.getString(0));

 uchenik.add(cursor.getString(1));

 prepod.add(cursor.getString(2));

 tema.add(cursor.getString(3));

 ocenka.add(cursor.getString(4)); }}}

Теперь необходимо создать новый класс «CustomAdapter», чтобы отображать данные в RecyclerView. Также необходимо создать в папке layout новый Layout Resource File (рис. 4).



Рис. 4. Создание нового Layout Resource File

Добавляем на него объект «CardView», также в поиске объектов вводим «ConstraintLayout» и перетаскиваем его на «CardView». Расставляем объекты внутри окошка (рис. 5). Это будет шаблон, по которому будут вставляться данные из таблицы.



Рис. 5. Шаблон для вставки данных из таблицы

 Теперь напишем код для «CustomAdapter», который поможет соединить шаблон и данные воедино, чтобы вывести таблицу на экран (листинг 9)

Листинг 9

Код для класса «CustomAdapter»

…

public class CustomAdapter extends RecyclerView.Adapter<CustomAdapter.MyViewHolder> {

 private Context context;

 private ArrayList id, uchenik, prepod,tema, ocenka;

 CustomAdapter(Context context,

 ArrayList id,

 ArrayList uchenik,

 ArrayList prepod,

 ArrayList tema,

 ArrayList ocenka){

 this.context = context;

 this.id = id;

 this.uchenik = uchenik;

 this.prepod = prepod;

 this.tema = tema;

 this.ocenka = ocenka;

 }

 @NonNull

 @Override

 public MyViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {

 LayoutInflater inflater = LayoutInflater.from(context);

 View view = inflater.inflate(R.layout.my\_row, parent, false);

 return new MyViewHolder(view);

 }

 @Override

 public void onBindViewHolder(@NonNull MyViewHolder holder, int position) {

 holder.id\_txt.setText(String.valueOf(id.get(position)));

 holder.uchenik\_txt.setText(String.valueOf(uchenik.get(position)));

 holder.prepod\_txt.setText(String.valueOf(prepod.get(position)));

 holder.tema\_txt.setText(String.valueOf(tema.get(position)));

 holder.ocenka\_txt.setText(String.valueOf(ocenka.get(position)));

 }

 @Override

 public int getItemCount() {

 return id.size();

 }

 public class MyViewHolder extends RecyclerView.ViewHolder {

 TextView id\_txt, uchenik\_txt, prepod\_txt, tema\_txt, ocenka\_txt;

 public MyViewHolder(@NonNull View itemView) {

 super(itemView);

 id\_txt = itemView.findViewById(R.id.id\_txt);

 uchenik\_txt = itemView.findViewById(R.id.uchenik\_txt);

 prepod\_txt = itemView.findViewById(R.id.prepod\_txt);

 tema\_txt = itemView.findViewById(R.id.tema\_txt);

 ocenka\_txt = itemView.findViewById(R.id.ocenka\_txt);

 }

 }

}

После этого необходимо в файле «NextActivity» добавить и инициализировать адаптер (листинг 10).

Листинг 10

Добавление и инициализация объекта класса адаптера

…

 CustomAdapter customAdapter;

 @Override

 protected void onCreate(Bundle savedInstanceState) {

 …

 customAdapter = new CustomAdapter(NextActivity.this, id, uchenik, prepod, tema, ocenka);

 recyclerView.setAdapter(customAdapter);

 recyclerView.setLayoutManager(new LinearLayoutManager(NextActivity.this));

 }

Теперь таблица выводится на второй активности (рис. 5).



Рис. 5. Вывод данных из таблицы на экран

Также при добавлении новых данных они сразу же начнут выводится на второй активности. При перезагрузке приложения данные никуда не пропадают, т.к. они хранятся в базе данных на телефоне.

**Список литературы**

1. Парамонов, И. В. Разработка мобильных приложений для платформы Android: учебное пособие / И. В. Парамонов; Яросл. гос. ун-т им. П. Г. Демидова. — Ярославль : ЯрГУ, 2013. — 88 с.
2. Семенчук, В. Мобильное приложение как инструмент бизнеса / В. Семенчук. — Москва : Альпина Паблишер, 2017. — 240 с.