

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное автономное образовательное  
учреждение высшего образования

«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»



Инженерная школа ядерных технологий  
Направление 01.04.02 «Прикладная математика и информатика»  
Специализация «Математическое моделирование и компьютерные вычисления»

**ОТЧЕТ**  
**ПО ЛАБОРАТОРНОЙ РАБОТЕ №4**  
**«Взаимодействие с файловой системой, SQLite»**  
**Вариант 1**

по дисциплине:  
**Программирование мобильных приложений**

**Исполнитель:**  
студент группы 0BM92

Е. В. Петрович \_\_\_\_\_  
Дата сдачи: \_\_\_\_\_

**Руководитель:**  
ассистент

А. В. Погребной \_\_\_\_\_  
Дата проверки: \_\_\_\_\_

Томск - 2019

## Оглавление

<b>Цель работы</b> .....	2
<b>Задание</b> .....	2
Общее задание .....	2
Задание варианта 1 .....	2
<b>Основная часть</b> .....	2
<b>Заключение</b> .....	4
<b>Приложение 1</b> .....	5
Исходный код программы .....	5
Модуль Lab4Activity .....	5
Модуль StudentManager .....	8
Модуль StudentsAdapter .....	9
Модуль Student .....	12
Модуль StudentDao .....	14
Модуль макета lab4_activity .....	15
<b>Приложение 2</b> .....	16
Ссылка на проект .....	16

## Цель работы

Работа с RecyclerView, и базой данных SQLite.

## Задание

### *Общее задание*

Ознакомиться с примером модуля lab4. Все задания подразумевают продолжение выполнения задания из Лабораторной работы №3. Реализовать задание Варианта 1.

### *Задание варианта 1*

Организовать поиск через БД. Индексировать ФИО студента через отдельную виртуальную таблицу.

## Основная часть

Для работы с базой данных SQLite в Android предоставляется библиотека «Room Persistence Library». Room предоставляет высокоуровневый API поверх API SQLite. Room состоит из трех основных компонентов:

1. Entity - представляющую сущность или таблицу в базе данных;
2. Dao (Database Access Object) – компонент, предоставляющий методы для работы с базой данных и содержащий соответствующие этим методам SQL запросы;
3. Database – класс, представляющий саму базу данных.

В отличие от лабораторной работы №3, список студентов хранится в базе данных, а не в кэше в памяти. Для работы с базой данных нужно только переопределить соответствующие методы в классе StudentManager и добавить в компонент StudentDao метод, позволяющий получить отфильтрованный список студентов и соответствующий этому методу SQL запрос. Для хранения списка студентов в базе данных создана виртуальная таблица Fts4.

Приложение было запущено на реальном устройстве Samsung Galaxy Note 9. Скриншоты работы представлены на рисунках. Исходный код модулей представлен в Приложении 1. Ссылка на репозиторий проекта на Github представлена в Приложении 2.

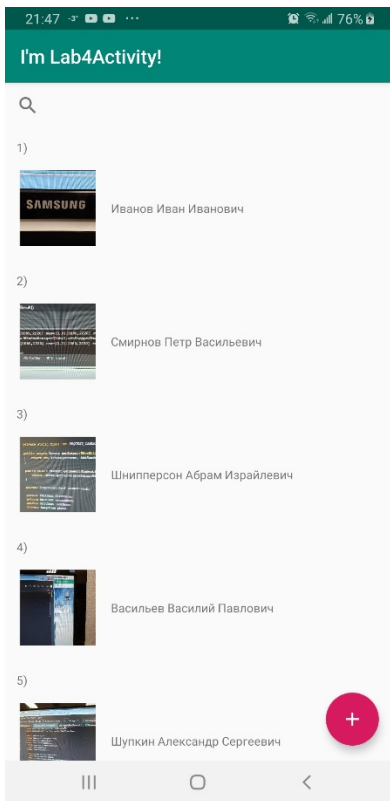


Рис. 1. Полный список

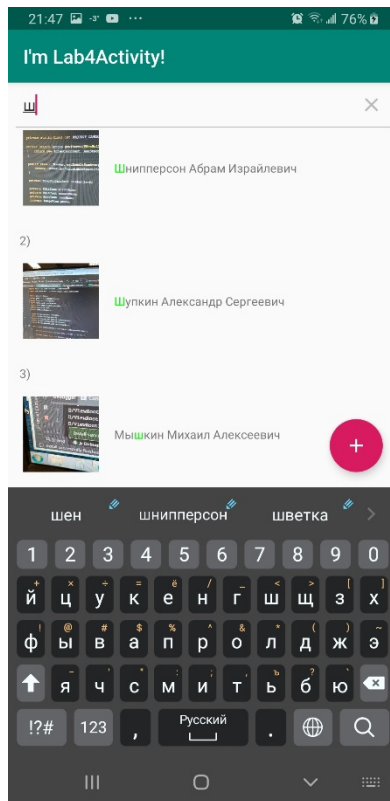


Рис. 2. Фильтрация списка по «ш»

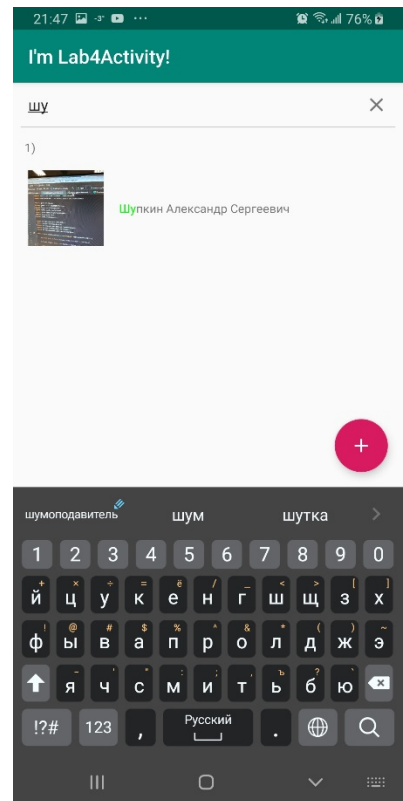


Рис. 3. Фильтрация списка по «шу»

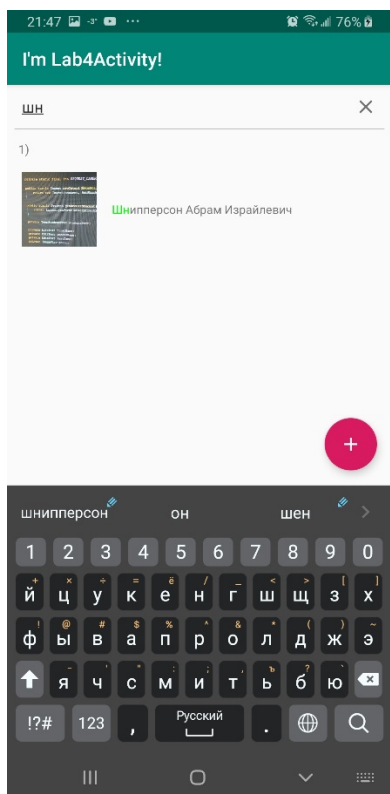


Рис. 4. Фильтрация списка по «шн»

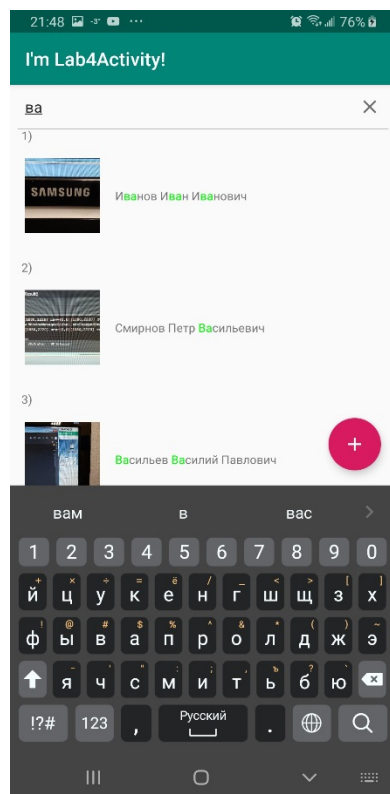


Рис. 5. Фильтрация списка по «ва»

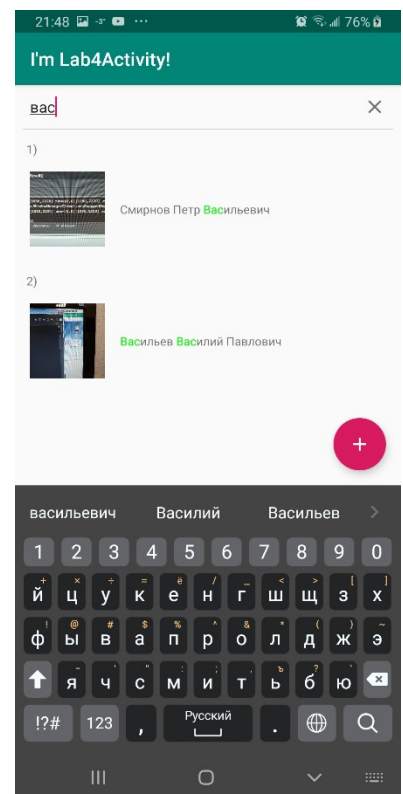


Рис. 6. Фильтрация списка по «вас»

## **Заключение**

В работе реализована фильтрация списка студентов в соответствии с требованиями, изложенными в задании. Подстрока фильтра подсвечивается на отображаемых элементах списка. Проект протестирован на реальном устройстве и загружен на сайт «Github».

## Приложение 1

### *Исходный код программы*

#### *Модуль Lab4Activity*

```
package ru.tpu.courses.lab4;

import android.content.Context;
import android.content.Intent;
import android.os.Bundle;
import android.widget.SearchView;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;

import com.google.android.material.floatingactionbutton.FloatingActionButton;

import ru.tpu.courses.lab4.adapter.StudentsAdapter;
import ru.tpu.courses.lab4.add.AddStudentActivity;
import ru.tpu.courses.lab4.db.Lab4Database;
import ru.tpu.courses.lab4.db.Student;
import ru.tpu.courses.lab4.db.StudentDao;
import ru.tpu.courses.lab4.db.StudentManager;

public class Lab4Activity extends AppCompatActivity implements
SearchView.OnQueryTextListener {

    private static final int REQUEST_STUDENT_ADD = 1;

    /**
     *
     */
    public static Intent newIntent(@NonNull Context context) {
        return new Intent(context, Lab4Activity.class);
    }
}

/**
 *
 */
private RecyclerView list;
private FloatingActionButton fab;
private StudentsAdapter studentsAdapter;
private SearchView filter;
private StudentManager studentManager;
/**
 *
 */
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);

    //studentDao = Lab4Database.getInstance(this).studentDao();

    setTitle(getString(R.string.lab4_title, getClass().getSimpleName()));
```

```

setContentView(R.layout.lab4_activity);
list = findViewById(android.R.id.list);
fab = findViewById(R.id.fab);
studentManager = new StudentManager(this);

LinearLayoutManager layoutManager = new LinearLayoutManager(this);
list.setLayoutManager(layoutManager);

list.setAdapter(studentsAdapter = new StudentsAdapter());
studentsAdapter.setStudents(studentManager.getStudents());

fab.setOnClickListener(
    v -> startActivityForResult(
        AddStudentActivity.newIntent(this),
        REQUEST_STUDENT_ADD
    )
);

// Установка "слушателя" SearchView
filter = findViewById(R.id.filterView);
filter.setOnQueryTextListener(this);
}

//*****
@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    super.onActivityResult(requestCode, resultCode, data);
    if (requestCode == REQUEST_STUDENT_ADD && resultCode == RESULT_OK) {
        Student student = AddStudentActivity.getResultStudent(data);
        studentManager.insert(student);
        updateAdapterData();
    }
}

//*****
@Override
public boolean onQueryTextSubmit(String query) {
    return false;
}

//*****
@Override
public boolean onQueryTextChange(String newText) {
    studentManager.setFilter(newText);
    studentsAdapter.setHighlight(newText);
    updateAdapterData();
    return false;
}

//*****

```

```
private void updateAdapterData() {  
    studentsAdapter.setStudents(studentManager.getStudents());  
    studentsAdapter.notifyDataSetChanged();  
    list.scrollToPosition(studentsAdapter.getItemCount() - 1); }  
}
```



### *Модуль StudentManager*

```
package ru.tpu.courses.lab4.db;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;

import java.util.List;

/**
 */
public class StudentManager {
    private final StudentDao studentDao;
    private String filter = new String();

    public StudentManager(AppCompatActivity activity) {
        studentDao = Lab4Database.getInstance(activity).studentDao();
    }

    /**
     */
    public void setFilter(final String filter) {
        // установка фильтра
        this.filter = filter.toLowerCase();
    }

    /**
     */
    public void insert(Student student) {
        studentDao.insert(student);
    }

    /**
     */
    public List<Student> getStudents() {
        if(filter.isEmpty())
        {
            return studentDao.getAll();
        }
        else
        {
            return studentDao.getFiltered(filter);
        }
    }
}
```

### *Модуль StudentsAdapter*

```
package ru.tpu.courses.lab4.adapter;

import android.graphics.Color;
import android.net.Uri;
import android.text.Spannable;
import android.text.SpannableString;
import android.text.Spanned;
import android.text.TextUtils;
import android.text.style.ForegroundColorSpan;
import android.view.View;
import android.view.ViewGroup;

import androidx.annotation.NonNull;
import androidx.recyclerview.widget.RecyclerView;

import java.util.ArrayList;
import java.util.List;
import java.util.regex.Matcher;
import java.util.regex.Pattern;

import ru.tpu.courses.lab4.db.Student;

/*****
public class StudentsAdapter extends RecyclerView.Adapter<RecyclerView.ViewHolder> {

    public static final int TYPE_NUMBER = 0;
    public static final int TYPE_STUDENT = 1;

    private List<Student> students = new ArrayList<>();
    // строка для подсветки
    public String highlight = new String();

    /*****
    @Override
    @NonNull
    public RecyclerView.ViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int
viewType) {
        switch (viewType) {
            case TYPE_NUMBER:
                return new NumberHolder(parent);
            case TYPE_STUDENT:
                return new StudentHolder(parent);
        }
        throw new IllegalArgumentException("unknown viewType = " + viewType);
    }

    /*****
    @Override
```

```

public void onBindViewHolder(@NonNull RecyclerView.ViewHolder holder, int position) {
    switch (getItemViewType(position)) {
        case TYPE_NUMBER:
            NumberHolder numberHolder = (NumberHolder) holder;
            numberHolder.bind((position + 1) / 2);
            break;
        case TYPE_STUDENT:
            StudentHolder studentHolder = (StudentHolder) holder;
            Student student = students.get(position / 2);

            // метод createStudentString() формирует строку для отображения
            studentHolder.student.setText(createStudentString(student));

            if (!TextUtils.isEmpty(student.photoPath)) {
                studentHolder.photo.setVisibility(View.VISIBLE);
                studentHolder.photo.setImageURI(Uri.parse(student.photoPath));
            } else {
                studentHolder.photo.setVisibility(View.GONE);
                studentHolder.photo.setImageURI(null);
            }
            break;
    }
}

//*****
@Override
public int getItemCount() {
    return students.size() * 2;
}

//*****
@Override
public int getItemViewType(int position) {
    return position % 2 == 0 ? TYPE_NUMBER : TYPE_STUDENT;
}

//*****
public void setStudents(List<Student> students) {
    this.students = students;
}

//*****
// Установка строки для подсвечивания
public void setHighlight(String highlight) {
    this.highlight = highlight.toLowerCase();
}

//*****
// Формирование строки для отображения
private Spannable createStudentString(final Student student) {

```

```

// Строка для отображения
String studentString = student.lastName + " " + student.firstName + " " + student.secondName;
// поиск позиций вхождения строки highlight
List<Integer> entries = findHighLightEntries(studentString);
// устанавливаем подсветку во все позиции строки
Spannable highlightedString = new SpannableString(studentString);
for (int i = 0; i < entries.size(); ++i) {
    highlightedString.setSpan(new ForegroundColorSpan(Color.GREEN),
        entries.get(i),
        entries.get(i) + highlight.length(),
        Spanned.SPAN_EXCLUSIVE_EXCLUSIVE);
}

return highlightedString;
}

//*****
private List<Integer> findHighLightEntries(final String string) {
    // приводим исходную строку к нижнему регистру
    String lowerString = string.toLowerCase();
    // список позиций вхождения
    List<Integer> entries = new ArrayList<>();

    // объект класса Matcher представляет исходную строку
    // и производит поиск в ней вхождения регулярного выражения
    Matcher m = Pattern.compile("(?=(" + highlight + "))").matcher(lowerString);
    while (m.find()) {
        // найденная позиция помещается в список
        entries.add(m.start());
    }

    return entries;
}
}

```

### *Модуль Student*

```
package ru.tpu.courses.lab4.db;

import android.os.Parcel;
import android.os.Parcelable;

import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.room.ColumnInfo;
import androidx.room.Entity;
import androidx.room.Fts4;
import androidx.room.Ignore;
import androidx.room.PrimaryKey;

@Fts4
@Entity
public class Student implements Parcelable {
    @PrimaryKey(autoGenerate = true)
    @ColumnInfo(name = "rowid")
    @Ignore
    public int id;
    @NonNull
    @ColumnInfo(name = "first_name")
    public String firstName;
    @NonNull
    @ColumnInfo(name = "second_name")
    public String secondName;
    @NonNull
    @ColumnInfo(name = "last_name")
    public String lastName;
    @Nullable
    @ColumnInfo(name = "photo_path")
    public String photoPath;

    public Student(
        @NonNull String firstName,
        @NonNull String secondName,
        @NonNull String lastName,
        @Nullable String photoPath
    ) {
        this.lastName = lastName;
        this.firstName = firstName;
        this.secondName = secondName;
        this.photoPath = photoPath;
    }

    protected Student(Parcel in) {
        id = in.readInt();
```

```

        firstName = in.readString();
        secondName = in.readString();
        lastName = in.readString();
        photoPath = in.readString();
    }

    public static final Creator<Student> CREATOR = new Creator<Student>() {
        @Override
        public Student createFromParcel(Parcel in) {
            return new Student(in);
        }

        @Override
        public Student[] newArray(int size) {
            return new Student[size];
        }
    };

    @Override
    public int describeContents() {
        return 0;
    }

    @Override
    public void writeToParcel(Parcel dest, int flags) {
        dest.writeInt(id);
        dest.writeString(firstName);
        dest.writeString(secondName);
        dest.writeString(lastName);
        dest.writeString(photoPath);
    }
}

```

### *Модуль StudentDao*

```
package ru.tpu.courses.lab4.db;

import androidx.annotation.NonNull;
import androidx.room.Dao;
import androidx.room.Insert;
import androidx.room.Query;

import java.util.List;

@Dao
public interface StudentDao {
    @Query("SELECT * FROM student")
    List<Student> getAll();

    @Query("SELECT * FROM student WHERE " +
            "first_name LIKE '%||:filter||%' OR " +
            "second_name LIKE '%||:filter||%' OR " +
            "last_name LIKE '%||:filter||%'")
    List<Student> getFiltered(@NonNull String filter);

    @Insert
    void insert(@NonNull Student student);

    @Query(
        "SELECT COUNT(*) FROM student WHERE " +
        "first_name = :firstName AND " +
        "second_name = :secondName AND " +
        "last_name = :lastName"
    )
    int count(@NonNull String firstName, @NonNull String secondName, @NonNull String
lastName);
}
```

### *Модуль макета lab4\_activity*

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">

    <SearchView
        android:id="@+id/filterView"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" />

    <FrameLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent">

        <androidx.recyclerview.widget.RecyclerView
            android:id="@android:id/list"
            android:layout_width="match_parent"
            android:layout_height="match_parent">

        </androidx.recyclerview.widget.RecyclerView>

        <com.google.android.material.floatingactionbutton.FloatingActionButton
            android:id="@+id/fab"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_gravity="end|bottom"
            android:layout_margin="16dp"
            android:src="@drawable/lab4_ic_add_white_24dp" />

    </FrameLayout>
</LinearLayout>
```



## Приложение 2

*Ссылка на проект*

<https://github.com/etrovicheugene/MobAppLabs.git>