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»

**ОТЧЕТ**

по лабораторной работе №3

«RecyclerView, взаимодействие между экранами. Memory Cache. Вариант 1»

по дисциплине:

**Программирование мобильных приложений**

|  |  |  |  |
| --- | --- | --- | --- |
| **Исполнитель:** |  | Е. В. Петрович | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| студент группы 0ВМ92 |  | Дата сдачи: | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
|  |  |  |  |
|  |  |  |  |
| **Руководитель:** |  | А. В. Погребной | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| доцент, |  | Дата проверки: | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| кандидат технических наук |  |  |  |
|  |  |  |  |

Томск - 2019

Оглавление

[**Цель работы** 2](#_Toc24451473)

[**Задание** 2](#_Toc24451474)

[Общее задание 2](#_Toc24451475)

[Задание варианта 1 2](#_Toc24451476)

[**Основная часть** 2](#_Toc24451477)

[**Заключение** 4](#_Toc24451478)

[**Приложение 1** 5](#_Toc24451479)

[Исходный код программы 5](#_Toc24451480)

[Модуль Lab3Activity 5](#_Toc24451481)

[Модуль StudentManager 7](#_Toc24451482)

[Модуль StudentsAdapter 8](#_Toc24451483)

[Модуль макета activity\_lab3 11](#_Toc24451484)

[**Приложение 2** 12](#_Toc24451485)

[Ссылка на проект 12](#_Toc24451486)

**Цель работы**

Работа с компонентом RecyclerView, организация кэширования данных в памяти и взаимодействия между экранами.

**Задание**

Общее задание

Ознакомиться с примером модуля lab3. Реализовать задание Варианта 1.

Задание варианта 1

На экран со студентами добавить SearchView. При поиске искать по ФИО студентов в списке на наличие введённой фразы в SearchView и выводить отфильтрованный список. Для каждого результата подсвечивать цветом кусок текста, который нашёлся (через SpannableString).

**Основная часть**

Для добавления компонента SearchView устанавливаем корневым макетом LinearLayout, сверху на него устанавливаем компонент SearchView, а оставшуюся часть экрана будет занимать FrameLayout с установленными на нем RecyclerView и кнопкой добавления студента. Слушателем изменения текста на компоненте SearchView устанавливаем Lab3Activity. Для фильтрации списка студентов, выводимого на компонент RecyclerView создаем специальный класс StudentManager. Компонент StudentsAdapter, который заполняет компоненты ViewHolder на RecyclerView будет получать не весь список из StudentsCache, а - отфильтрованный из StudentManager. Для подсветки строки-фильтра в списке, в StudentsAdapter в методе OnBindViewHolder вместо обычной строки вставляем в компоненты ViewHolder объекты класса Spannable, которые позволяют выделять текст. Для определения мест в тексте, который должны быть подсвечены, добавляем методы createStudentString() и findHighlightEntries(). 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.

|  |  |  |
| --- | --- | --- |
| D:\SharedFolder\screenshots3\Screenshot_20191110-110155_Android Labs.jpg | D:\SharedFolder\screenshots3\Screenshot_20191110-110212_Android Labs.jpg | D:\SharedFolder\screenshots3\Screenshot_20191110-110222_Android Labs.jpg |
| Рис. 1. Полный список | Рис. 2. Буква «и» присутствует во всех ФИО | Рис. 3. Фильтрация списка по «ив» |

|  |  |  |
| --- | --- | --- |
| D:\SharedFolder\screenshots3\Screenshot_20191110-110236_Android Labs.jpg | D:\SharedFolder\screenshots3\Screenshot_20191110-110255_Android Labs.jpg | D:\SharedFolder\screenshots3\Screenshot_20191110-110332_Android Labs.jpg |
| Рис. 4. Фильтрация списка по «из» | Рис. 5. Фильтрация списка по «ий» | Рис. 6. Фильтрация списка по «Ш» |

**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 Lab3Activity

package ru.tpu.courses.lab3;

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.lab3.adapter.StudentsAdapter;

public class Lab3Activity extends AppCompatActivity implements SearchView.OnQueryTextListener {

private static final int REQUEST\_STUDENT\_ADD = 1;

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

public static Intent newIntent(@NonNull Context context) {

return new Intent(context, Lab3Activity.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);

setTitle(getString(R.string.lab3\_title, getClass().getSimpleName()));

setContentView(R.layout.lab3\_activity);

list = findViewById(android.R.id.list);

fab = findViewById(R.id.fab);

studentManager = new StudentManager();

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.addStudent(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.lab3;

import androidx.annotation.NonNull;

import java.util.List;

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

class StudentManager {

private final StudentsCache studentsCache = StudentsCache.getInstance();

private String filter = new String();

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

public void setFilter(final String filter) {

// установка фильтра

this.filter = filter.toLowerCase();

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

public void addStudent(Student student) {

studentsCache.addStudent(student);

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

@NonNull

public List<Student> getStudents() {

// получаем полный список студентов из StudentCashe

List<Student> students = studentsCache.getStudents();

// если строка фильтра не пустая, удаляем из списка все элементы,

// которые не содержат строку фильтра

if (!filter.isEmpty()) {

for (int i = students.size() - 1; i >= 0; --i) {

Student student = students.get(i);

if (student.firstName.toLowerCase().contains(filter) |

student.secondName.toLowerCase().contains(filter) |

student.lastName.toLowerCase().contains(filter)) {

continue;

}

students.remove(i);

}

}

return students;

}

}

Модуль StudentsAdapter

package ru.tpu.courses.lab3.adapter;

import android.graphics.Color;

import android.graphics.Typeface;

import android.text.Spannable;

import android.text.SpannableString;

import android.text.Spanned;

import android.text.style.ForegroundColorSpan;

import android.text.style.StyleSpan;

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.lab3.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;

// Высчитыванием номер студента начиная с 1

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));

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 макета activity\_lab3

<?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/lab3\_ic\_add\_white\_24dp" />

</FrameLayout>

</LinearLayout>

**Приложение 2**

Ссылка на проект

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