**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ**

федеральное государственное автономное образовательное учреждение высшего образования

**«Национальный исследовательский Томский политехнический университет»**

Инженерная школа информационных технологий и робототехники

Направление «Информационные системы и технологии»

Отчёт по лабораторной работе №3

**«RecyclerView, взаимодействие между экранами. Memory Cache»**

по дисциплине **«Программирование мобильных устройств и встраиваемых систем»**

Выполнил:

Студент группы 8И6А Козловский В. Е. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Проверил:

Преподаватель ОИТ Погребной А.В. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# Томск 2020

# Задание

Вариант 4. Над кнопкой добавления студента добавить еще один FloatingActionButton, который открывает диалог (AlertDialog) с добавлением группы (название группы и кнопка «ок»). На экране добавления студента добавить Spinner, который позволяет выбрать, к какой группе принадлежит студент. При нажатии на студента в списке, открывать AddStudentActivity с заполненными параметрами студента. Выводить список студентов, сгруппированный по группам. Т.е. сначала выводится группа, потом список студентов в группе.

# Ход работы

Для описания логики отображения данных о студентах в группах был создан класс “GroupStudentAdapter”:

**package** ru.tpu.courses.lab3.adapter;  
  
**import** android.util.Log;  
**import** android.view.ViewGroup;  
  
**import** androidx.annotation.NonNull;  
**import** androidx.recyclerview.widget.RecyclerView;  
  
**import** java.util.ArrayList;  
**import** java.util.List;  
  
**import** ru.tpu.courses.lab3.Group;  
**import** ru.tpu.courses.lab3.Student;  
**import** ru.tpu.courses.lab3.StudentGroupListItem;  
  
**public class** GroupStudentAdapter **extends** RecyclerView.Adapter<RecyclerView.ViewHolder> {  
  
 **public static final int *TYPE\_GROUP*** = 0;  
 **public static final int *TYPE\_STUDENT*** = 1;  
  
 **private** List<StudentGroupListItem> **studentGroups** = **new** ArrayList<>();  
 **private** onStudentClickListener **mOnStudentClickListener**;  
  
 **public** GroupStudentAdapter(onStudentClickListener listener){  
 **mOnStudentClickListener** = listener;  
 }  
  
 @Override  
 @NonNull  
 **public** RecyclerView.ViewHolder onCreateViewHolder(@NonNull ViewGroup parent, **int** viewType) {  
 **switch** (viewType) {  
 **case *TYPE\_STUDENT***:  
 **return new** StudentHolder(parent, **mOnStudentClickListener**);  
 **case *TYPE\_GROUP***:  
 **return new** GroupHolder(parent);  
 }  
 **throw new** IllegalArgumentException(**"unknown viewType = "** + viewType);  
 }  
  
 @Override  
 **public void** onBindViewHolder(@NonNull RecyclerView.ViewHolder holder, **int** position) {  
 **switch** (getItemViewType(position)) {  
 **case *TYPE\_STUDENT***:  
 Student student = (Student) **studentGroups**.get(position);  
 StudentHolder studentHolder = (StudentHolder) holder;  
 studentHolder.setStudent(student);  
 **break**;  
 **case *TYPE\_GROUP***:  
 Group group = (Group) **studentGroups**.get(position);  
 GroupHolder groupHolder = (GroupHolder) holder;  
 groupHolder.**group**.setText(group.**groupName**);  
 **break**;  
 }  
 }  
  
 @Override  
 **public int** getItemCount() {  
 **return studentGroups**.size();  
 }  
  
 @Override  
 **public int** getItemViewType(**int** position) {  
 **return studentGroups**.get(position).getType();  
 }  
  
 **public interface** onStudentClickListener{  
 **void** onStudentClick(**int** id);  
 }  
  
 **public void** setData(List<Student> students, List<Group> groups){  
 **this**.**studentGroups**.clear();  
 **for** (Group group: groups  
 ) {  
 **this**.**studentGroups**.add(group);  
 **for** (Student student:students  
 ) {  
 **if**(student.**groupName**.equals(group.**groupName**)){  
  
 **this**.**studentGroups**.add(student);  
 }   
 }  
 }  
  
 **for** (Student student: students){  
 **if**(student.**groupName** == **""**)  
 **this**.**studentGroups**.add(student);  
 }  
 }  
}

AddStudentActivity для добавления нового студента было дополнено с учетом возможности редактирования данных студента (принадлежность к группе):

**package** ru.tpu.courses.lab3;  
  
**import** android.content.Context;  
**import** android.content.Intent;  
**import** android.os.Bundle;  
**import** android.text.TextUtils;  
**import** android.util.Log;  
**import** android.view.Menu;  
**import** android.view.MenuItem;  
**import** android.widget.ArrayAdapter;  
**import** android.widget.EditText;  
**import** android.widget.Spinner;  
**import** android.widget.SpinnerAdapter;  
**import** android.widget.Toast;  
  
**import** androidx.annotation.NonNull;  
**import** androidx.appcompat.app.AppCompatActivity;  
  
**import** ru.tpu.courses.lab3.adapter.StudentsAdapter;  
**import** ru.tpu.courses.lab3.adapter.GroupsAdapter;**public class** AddStudentActivity **extends** AppCompatActivity {  
  
 **private static final** String ***EXTRA\_STUDENT*** = **"student"**;  
 **private static** Student *oldStudent*;  
  
 **public static** Intent newIntent(@NonNull Context context) {  
 **return new** Intent(context, AddStudentActivity.**class**);  
 }  
  
 **public static** Intent newIntent(@NonNull Context context, Student student) {  
 Intent ourIntent = *newIntent*(context);  
  
 ourIntent.putExtra(***EXTRA\_STUDENT***, student);  
 *oldStudent* = student;  
 **return** ourIntent;  
 }  
  
 **public static** Student getOldStudent(){  
 **return** *oldStudent*;  
 }  
  
 **public static** Student getResultStudent(@NonNull Intent intent) {  
 **return** intent.getParcelableExtra(***EXTRA\_STUDENT***);  
 }  
  
 **private final** StudentsCache **studentsCache** = StudentsCache.*getInstance*();  
  
 **private** EditText **firstName**;  
 **private** EditText **secondName**;  
 **private** EditText **lastName**;  
 **private** Spinner **groupSpinner**;  
  
 **private** StudentsAdapter **studentsAdapter**;  
 **private static** GroupsCache *groupsCache* = GroupsCache.*getInstance*();  
  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.*lab3\_activity\_add\_student*);  
getSupportActionBar().setDisplayHomeAsUpEnabled(**true**);  
  
 **firstName** = findViewById(R.id.*first\_name*);  
 **secondName** = findViewById(R.id.*second\_name*);  
 **lastName** = findViewById(R.id.*last\_name*);  
 **groupSpinner** = findViewById(R.id.*lab3\_spinner*);  
  
 ArrayAdapter<Group> adapter = **new** ArrayAdapter<Group>(getApplicationContext(), R.layout.*lab3\_item\_group*, R.id.*group*);  
 adapter.addAll(*groupsCache*.getGroups());  
 **groupSpinner**.setAdapter(adapter);  
  
 Bundle arguments = getIntent().getExtras();  
  
 **if**(arguments!=**null**){  
 Student student = arguments.getParcelable(***EXTRA\_STUDENT***);  
 **firstName**.setText(student.**firstName**);  
 **lastName**.setText(student.**lastName**);  
 **secondName**.setText(student.**secondName**);  
 **if**(student.**groupName**.length() > 0)  
 *selectSpinnerItemByValue*(**groupSpinner**, student.**groupName**);  
 }  
 }  
  
 **private static void** selectSpinnerItemByValue(Spinner spnr, String value) {  
 SpinnerAdapter adapter = (SpinnerAdapter) spnr.getAdapter();  
 **for** (**int** position = 0; position < adapter.getCount(); position++) {  
 **if**(adapter.getItem(position).toString() == value) {  
 spnr.setSelection(position);  
 **return**;  
 }  
 }  
 }  
  
 @Override  
 **public boolean** onCreateOptionsMenu(Menu menu) {  
 getMenuInflater().inflate(R.menu.*lab3\_add\_student*, menu);  
 **return super**.onCreateOptionsMenu(menu);  
 }  
  
 @Override  
 **public boolean** onOptionsItemSelected(MenuItem item) {  
 **if** (item.getItemId() == android.R.id.***home***) {  
 finish();  
 **return true**;  
 }  
  
 **if** (item.getItemId() == R.id.*action\_save*) {  
 Student student = **new** Student(  
 **firstName**.getText().toString(),  
 **secondName**.getText().toString(),  
 **lastName**.getText().toString()  
 );  
  
 Object selectedGroupName = **groupSpinner**.getSelectedItem();  
 **if**(selectedGroupName != **null**) {  
 student.**groupName** = selectedGroupName.toString();  
 }  
  
 **if** (TextUtils.*isEmpty*(student.**firstName**) ||  
 TextUtils.*isEmpty*(student.**secondName**) ||  
 TextUtils.*isEmpty*(student.**lastName**)) {  
 Toast.*makeText*(**this**, R.string.*lab3\_error\_empty\_fields*, Toast.***LENGTH\_LONG***).show();  
 **return true**;  
 }  
  
 **if** (**studentsCache**.contains(student)) {  
 Toast.*makeText*(**this**, R.string.*lab3\_error\_already\_exists*, Toast.***LENGTH\_LONG***).show();  
 **return true**;  
 }  
  
 Intent data = **new** Intent();  
 data.putExtra(***EXTRA\_STUDENT***, student);  
 setResult(***RESULT\_OK***, data);  
  
 finish();  
 **return true**;  
 }  
 **return super**.onOptionsItemSelected(item);  
 }  
}

В основной Activity модуля были внесены изменения для обеспечения нового функционала. Для обработки нажатия на запись о студенте в Activity был имплементирован созданный ранее интерфейс “onStudentClick” класса “GroupStudentAdapter”.

**import** ru.tpu.courses.lab3.adapter.GroupStudentAdapter;  
  
**public class** Lab3Activity **extends** AppCompatActivity **implements** GroupStudentAdapter.onStudentClickListener{

Также был определен метод обработчика “onStudentClick”, вызывающий форму редактирования студента с предзаполненными данными.

@Override  
 **public void** onStudentClick(**int** hashCode) {  
 Student student = **studentsCache**.getByHashCode(hashCode);  
 startActivityForResult(  
 AddStudentActivity.*newIntent*(**this**, student),  
 ***REQUEST\_STUDENT\_EDIT*** );  
 }  
}

# Результат

При переходе к модулю 3 лабораторной работы на экран устройства выводится список студентов без группы, изначально пустой (рис. 1).

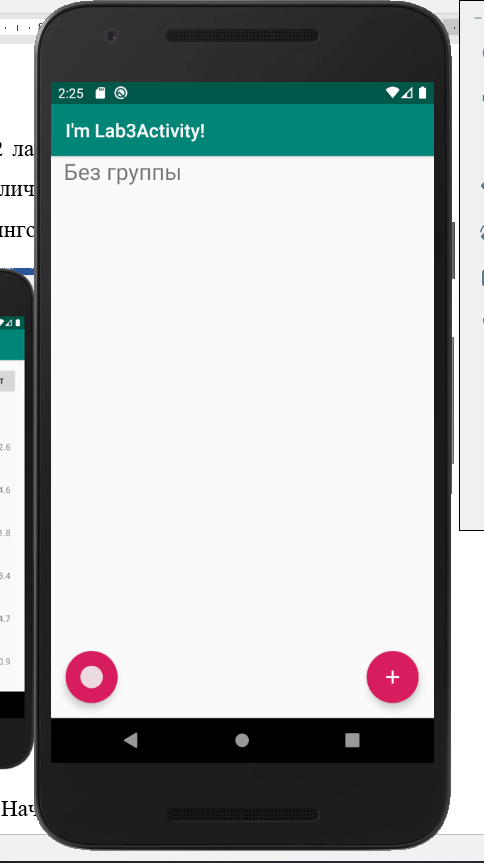


Рисунок 1 – Начальный вид модуля

При нажатии на кнопку слева, появляется экран добавления (рис. 2), где пользователь вводит название новой группы.

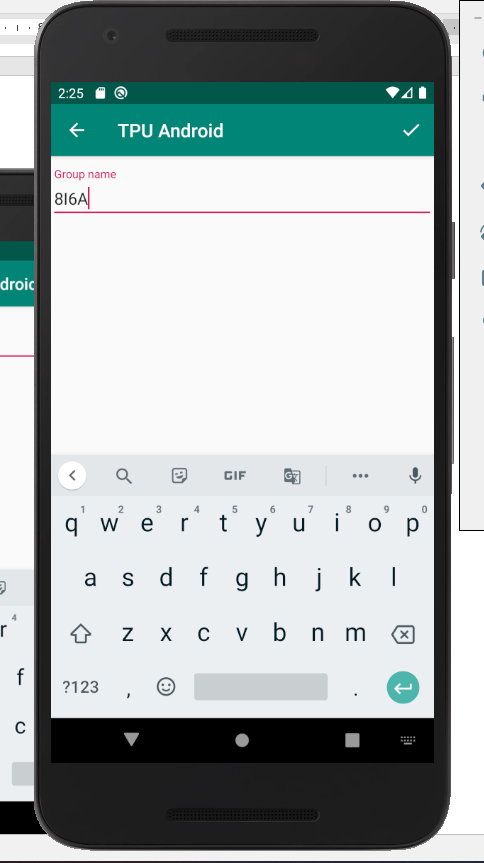
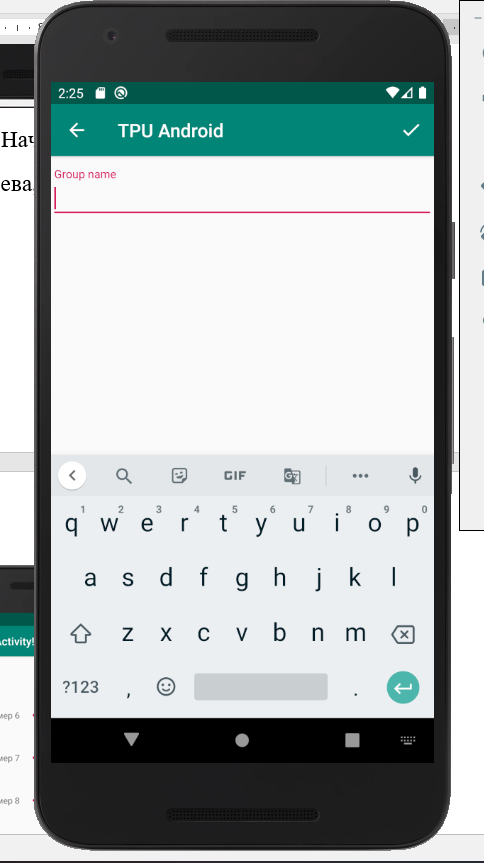


Рисунок 2 – Добавление новой группы

Теперь при добавлении нового студента можно выбрать новую группу (рис. 3)

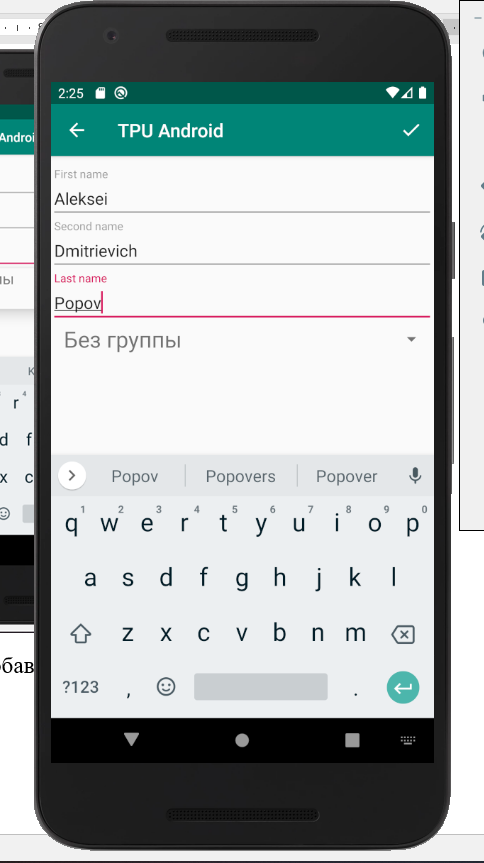
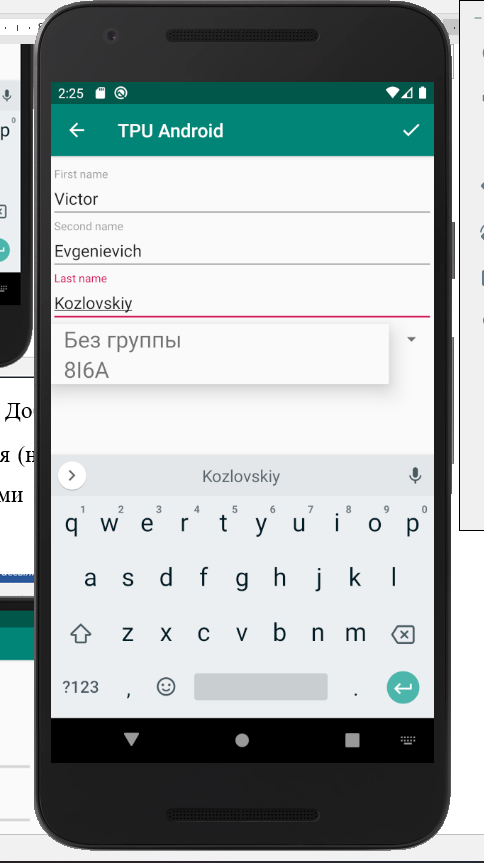


Рисунок 3 – Добавление нового студента

Добавленные студенты отображаются в разных группах (рис. 4).

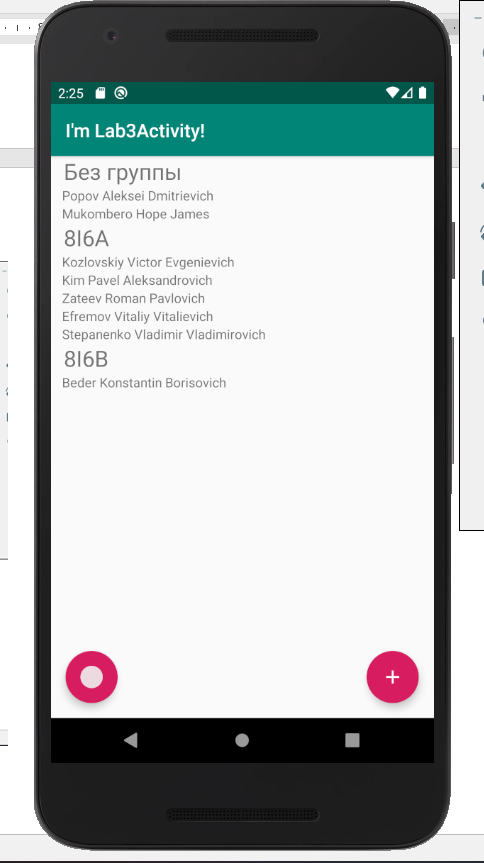
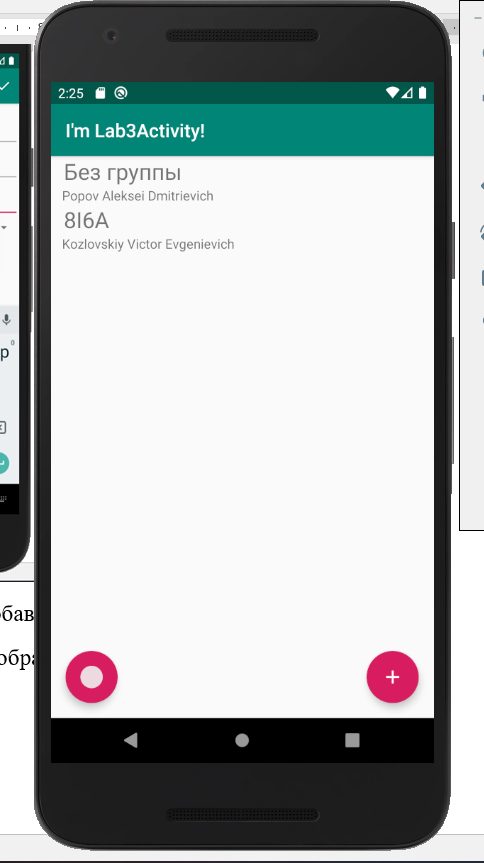


Рисунок 4 – Список студентов в группах

При нажатии на запись студента, открывается окно создания студента с заполненными данными (рис. 5).

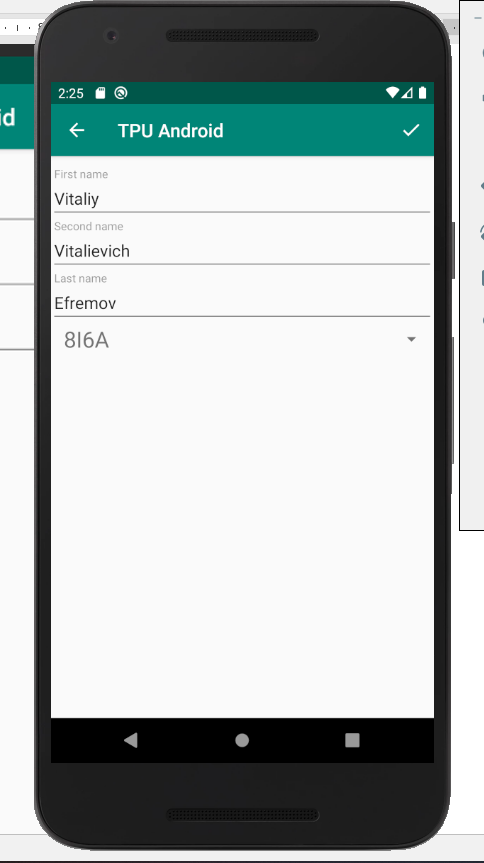
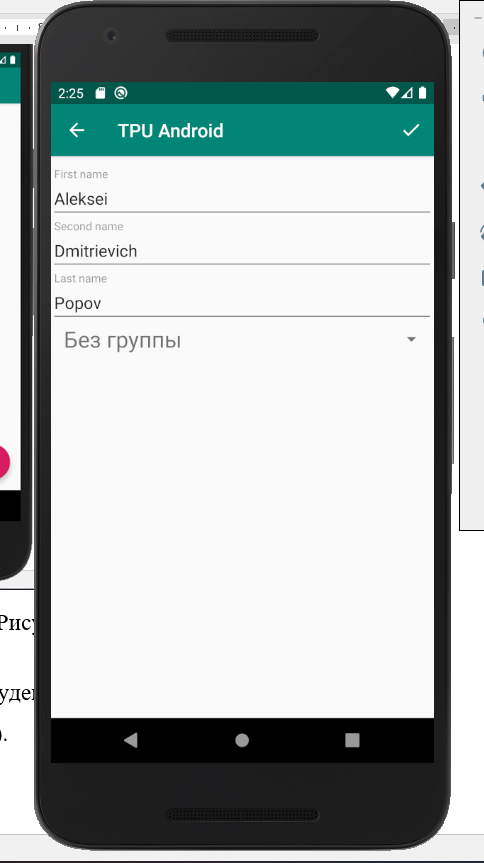


Рисунок 5 – Редактирование данных студента

# Вывод

В ходе выполнения лабораторной работы были закреплены на практике знания принципов работы компонентов Cache, Adapter, ViewHolder. Был реализован собственный интерфейс, содержащий метод обработки нажатия на элемент.