|  |  |  |
| --- | --- | --- |
|  |  |  |
| МИНОБРНАУКИ РОССИИ | | |
| Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** | | |

Институт Информационных технологий

Кафедра Математического обеспечения и стандартизации информационных технологий

**Отчет по практической работе №5**

по дисциплине «Разработка мобильных приложений»

|  |  |
| --- | --- |
| **Выполнил:**  студент группыИКБО-28-22 | Некрасов Г.А. |
| **Проверил:**  к.э.н доцент | Петросян Л.Э. |

Москва 2024 г.

**СОДЕРЖАНИЕ**

[Задание 3](#_Toc167625822)

[Выполнение работы 3](#_Toc167625823)

[Тестирование работы 15](#_Toc167625824)

# Задание

* Создать несколько Activity. На первом экране выводится список категорий продуктов (или своя категория). В зависимости от выбранного элемента списка, происходит переход на следующую Activity. На второй Activity выводится подробный список элементов выбранной категории. Например, выбираете категорию «Яблоки» и на второй Activity у вас перечень сортов яблок. Также должны быть реализованы возможности добавления и удаления элементов из списка второго Activity (хотя бы для одной выбираемой категории, в остальных можно просто вывести список).
* Создать Activity с RecyclerView для отображения данных. Выводить несколько полей с разными элементами: ОБЯЗАТЕЛЬНО картинку и TextView.
* Создать Activity со ScrollView с элементами TextView (минимальное количество элементов должно быть такое, чтобы возможно было проверить функцию скроллинга) и EditText (внизу страницы, чтобы можно было ввести текст).
* Создать Activity с выпадающим списком (Spinner).

# Выполнение работы

Во-первых, я создал список категорий фруктов чтобы их можно было выбрать в MainActivity (см. Листинг 1).

Листинг 1. MainActivity.java

|  |
| --- |
| package com.example.nekrasovglebandreevich\_5practpart1;  import android.content.Intent; import android.os.Bundle; import android.view.View; import android.widget.AdapterView; import android.widget.ArrayAdapter; import android.widget.ListView;  import androidx.appcompat.app.AppCompatActivity;  public class MainActivity extends AppCompatActivity {  String[] categories = {"Яблоки", "Груши", "Бананы"}; // пример категорий   @Override  protected void onCreate(Bundle savedInstanceState) {  super.onCreate(savedInstanceState);  setContentView(R.layout.*activity\_main*);   ListView listView = findViewById(R.id.*listView*);  ArrayAdapter<String> adapter = new ArrayAdapter<>(this, android.R.layout.*simple\_list\_item\_1*, categories);  listView.setAdapter(adapter);   listView.setOnItemClickListener(new AdapterView.OnItemClickListener() {  @Override  public void onItemClick(AdapterView<?> parent, View view, int position, long id) {  String selectedCategory = categories[position];  Intent intent = new Intent(MainActivity.this, CategoryActivity.class);  intent.putExtra("category", selectedCategory);  startActivity(intent);  }  });  } } |

Во-вторых, после создал CategoryActivity, в котором указал начальные примеры сортов, создал кнопки (для добавления/удаления сортов), а также возможность перейти к описанию выбранных сортов с их картинками (переход на следующую активность) (см. Листинг 2).

Листинг 2. CategoryActivity.java

|  |
| --- |
| package com.example.nekrasovglebandreevich\_5practpart1;  import android.os.Bundle; import android.util.SparseBooleanArray; import android.view.View; import android.widget.ArrayAdapter; import android.widget.Button; import android.widget.EditText; import android.widget.ListView; import android.widget.TextView; import android.widget.Toast; import android.content.Intent;  import androidx.appcompat.app.AppCompatActivity; import java.util.ArrayList; import java.util.List;  public class CategoryActivity extends AppCompatActivity {   ArrayList<String> products = new ArrayList<>();  ArrayAdapter<String> adapter;  TextView selectedItemsTextView;   @Override  protected void onCreate(Bundle savedInstanceState) {  super.onCreate(savedInstanceState);  setContentView(R.layout.*activity\_category*);   ListView listView = findViewById(R.id.*categoryListView*);  EditText editText = findViewById(R.id.*editText*);  Button addButton = findViewById(R.id.*addButton*);  Button deleteButton = findViewById(R.id.*deleteButton*);  Button showDescriptionButton = findViewById(R.id.*showDescriptionButton*);  selectedItemsTextView = findViewById(R.id.*selectedItemsTextView*);   adapter = new ArrayAdapter<>(this, android.R.layout.*simple\_list\_item\_multiple\_choice*, products);  listView.setAdapter(adapter);  listView.setChoiceMode(ListView.*CHOICE\_MODE\_MULTIPLE*);   // Получаем категорию из Intent  String category = getIntent().getStringExtra("category");   // Пример заполнения списка продуктов  if (category.equals("Яблоки")) {  products.add("Сорт Гала");  products.add("Сорт Джонаголд");  products.add("Сорт Болдуин");  } else if (category.equals("Груши")) {  products.add("Сорт Анжуйский");  products.add("Сорт Азиатский");  } else if (category.equals("Бананы")) {  products.add("Сорт Кавендиш");  products.add("Сорт Бурро");  products.add("Сорт Нино");  }   adapter.notifyDataSetChanged();   addButton.setOnClickListener(new View.OnClickListener()  {  @Override  public void onClick(View v) {  String newProduct = editText.getText().toString().trim();  if (!newProduct.isEmpty()) {  products.add(newProduct);  adapter.notifyDataSetChanged();  editText.setText("");  Toast.*makeText*(CategoryActivity.this, "Добавлен сорт: " + newProduct, Toast.*LENGTH\_SHORT*).show();  } else {  Toast.*makeText*(CategoryActivity.this, "Введите название сорта", Toast.*LENGTH\_SHORT*).show();  }  }  });   // Установка слушателя нажатий для кнопки  showDescriptionButton.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View v) {  // Получаем список выбранных сортов  ArrayList<String> selectedSorts = new ArrayList<>();  SparseBooleanArray checkedItems = listView.getCheckedItemPositions();  for (int i = 0; i < checkedItems.size(); i++) {  int position = checkedItems.keyAt(i);  if (checkedItems.valueAt(i)) {  selectedSorts.add(products.get(position));  }  }   // Проверяем количество выбранных сортов  if (selectedSorts.size() > 3) {  Toast.*makeText*(CategoryActivity.this, "Можно выбрать только три сорта", Toast.*LENGTH\_SHORT*).show();  return;  }   // Переходим к SortDescriptionActivity и передаем выбранные сорта  Intent intent = new Intent(CategoryActivity.this, SortDescriptionActivity.class);  intent.putStringArrayListExtra("selectedSorts", selectedSorts);  startActivity(intent);  }  });   deleteButton.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View v) {  SparseBooleanArray checkedItemPositions = listView.getCheckedItemPositions();  List<String> itemsToRemove = new ArrayList<>();  // Определяем элементы для удаления  for (int i = checkedItemPositions.size() - 1; i >= 0; i--) {  int position = checkedItemPositions.keyAt(i);  if (checkedItemPositions.get(position)) {  itemsToRemove.add(products.get(position));  }  }  // Удаляем элементы из списка  for (String item : itemsToRemove) {  adapter.remove(item);  }  adapter.notifyDataSetChanged();  // Снимаем выбор со всех удаляемых элементов  listView.clearChoices();  // Обновляем отображение выбранных сортов после удаления  updateSelectedItems();  }  });    // Обновляем отображение выбранных сортов при выборе элементов  listView.setOnItemClickListener((parent, view, position, id) -> updateSelectedItems());  }   // Метод для обновления списка выбранных сортов  private void updateSelectedItems() {  StringBuilder selectedItems = new StringBuilder("");  SparseBooleanArray checkedItemPositions = ((ListView) findViewById(R.id.*categoryListView*)).getCheckedItemPositions();  for (int i = 0; i < checkedItemPositions.size(); i++) {  int position = checkedItemPositions.keyAt(i);  if (checkedItemPositions.get(position)) {  selectedItems.append("\n- ").append(products.get(position));  }  }  selectedItemsTextView.setText(selectedItems.toString());  } } |

В-третьих, перейдём к реализации третьей активности, которая будет выводить описание, название и картинку сорта (см. Листинги 3-5).

Листинг 3. SortItem.java

|  |
| --- |
| package com.example.nekrasovglebandreevich\_5practpart1;  public class SortItem {  private String name;  private String description;  private int imageResource;   public SortItem(String name, String description, int imageResource) {  this.name = name;  this.description = description;  this.imageResource = imageResource;  }   public String getName() {  return name;  }   public String getDescription() {  return description;  }   public int getImageResource() {  return imageResource;  } } |

Листинг 4. SortAdapter.java

|  |
| --- |
| package com.example.nekrasovglebandreevich\_5practpart1;  import android.content.Context; import android.view.LayoutInflater; import android.view.View; import android.view.ViewGroup; import android.widget.ImageView; import android.widget.TextView;  import androidx.annotation.NonNull; import androidx.recyclerview.widget.RecyclerView; import java.util.ArrayList;  public class SortAdapter extends RecyclerView.Adapter<SortAdapter.SortViewHolder> {  private Context mContext;  private ArrayList<SortItem> mSortList;   public static class SortViewHolder extends RecyclerView.ViewHolder {  public ImageView mImageView;  public TextView mNameTextView;  public TextView mDescriptionTextView;   public SortViewHolder(View itemView) {  super(itemView);  mImageView = itemView.findViewById(R.id.*imageView*);  mNameTextView = itemView.findViewById(R.id.*nameTextView*);  mDescriptionTextView = itemView.findViewById(R.id.*descriptionTextView*);  }  }   public SortAdapter(Context context, ArrayList<SortItem> sortList) {  mContext = context;  mSortList = sortList;  }   @NonNull  @Override  public SortViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {  View v = LayoutInflater.*from*(mContext).inflate(R.layout.*sort\_item*, parent, false);  return new SortViewHolder(v);  }   @Override  public void onBindViewHolder(@NonNull SortViewHolder holder, int position) {  SortItem currentItem = mSortList.get(position);   holder.mImageView.setImageResource(currentItem.getImageResource());  holder.mNameTextView.setText(currentItem.getName());  holder.mDescriptionTextView.setText(currentItem.getDescription());  }  @Override  public int getItemCount() {  return mSortList.size();  } } |

Листинг 5. SortDescriptionActivity.java

|  |
| --- |
| package com.example.nekrasovglebandreevich\_5practpart1;  import android.content.Intent; import android.os.Bundle; import android.view.View; import android.widget.Button;  import androidx.appcompat.app.AppCompatActivity; import androidx.recyclerview.widget.LinearLayoutManager; import androidx.recyclerview.widget.RecyclerView;  import java.util.ArrayList;  public class SortDescriptionActivity extends AppCompatActivity {  private RecyclerView mRecyclerView;  private SortAdapter mAdapter;  private ArrayList<SortItem> mSortList;   @Override  protected void onCreate(Bundle savedInstanceState) {  super.onCreate(savedInstanceState);  setContentView(R.layout.*activity\_sort\_description*);   mRecyclerView = findViewById(R.id.*recyclerView*);  mRecyclerView.setHasFixedSize(true);  mRecyclerView.setLayoutManager(new LinearLayoutManager(this));  Button nextButton = findViewById(R.id.*nextButton*); // Найдите кнопку по идентификатору   // Получаем выбранные сорта из Intent  ArrayList<String> selectedSorts = getIntent().getStringArrayListExtra("selectedSorts");   // Инициализируем список сортов  mSortList = new ArrayList<>();   // Добавляем информацию о выбранных сортах в список  for (String sortName : selectedSorts) {  // Получите описание и изображение для каждого сорта  String description = getDescriptionForSort(sortName);  int imageResource = getImageResourceForSort(sortName);  mSortList.add(new SortItem(sortName, description, imageResource));  }   // Создаем адаптер и устанавливаем его для RecyclerView  mAdapter = new SortAdapter(this, mSortList);  mRecyclerView.setAdapter(mAdapter);   nextButton.setOnClickListener(new View.OnClickListener() { // Назначьте слушатель нажатий на кнопку  @Override  public void onClick(View v) {  // Создайте намерение для перехода на четвёртую активность и начните его  Intent intent = new Intent(SortDescriptionActivity.this, FourthActivity.class);  startActivity(intent);  }  });  }   // Метод для получения описания сорта  private String getDescriptionForSort(String sortName) {  // Здесь вы можете реализовать логику для получения описания сорта  // Возвращайте описание в зависимости от названия сорта  if (sortName.equals("Сорт Гала")) {  return "Они высокие и тонкие, с кожицей чуть более желтого цвета, чем у других яблок. Иногда на них даже бывают желтые полосы.";  } else if (sortName.equals("Сорт Джонаголд")) {  return "Джонагольды - это гибридные яблоки. Нечто среднее между терпким яблоком Джонатан и сладким голден делишес, они сладкие с легкой кислинкой.";  } else if (sortName.equals("Сорт Болдуин")) {  return "Эти яблоки в основном продаются на северо-востоке на местных фермерских рынках. У них пряно-терпко-сладкий вкус.";  } else if (sortName.equals("Сорт Анжуйский")) {  return "Анжуйские груши пухлые, с гладкой кожицей. Мякоть анжуйских груш сочная и упругая. Они сладкие и вкусные, если употреблять их в сыром виде.";  } else if (sortName.equals("Сорт Азиатский")) {  return "Азиатские груши также больше всего похожи на яблоки, чем другие груши. Благодаря своей хрустящей текстуре и сладкому вкусу эти груши идеально подходят для перекусов или употребления в сыром виде в салатах.";  } else if (sortName.equals("Сорт Кавендиш")) {  return "Их кожица ярко-зеленая и становится ярко- золотисто-желтой по мере созревания. Они имеют классический банановый вкус.";  } else if (sortName.equals("Сорт Бурро")) {  return "Бананы Бурро напоминают сорт Кавендиш, но они короче. Их мякоть при созревании имеет лимонно-банановый вкус.";  } else if (sortName.equals("Сорт Нино")) {  return "Бананы Нино короткие. После созревания их вкус становится богатым и сладким с ванильными и карамельными оттенками.";  }  return "Описание для " + sortName;  }   // Метод для получения ресурса изображения для сорта  private int getImageResourceForSort(String sortName) {  // Здесь вы можете реализовать логику для получения ресурса изображения для сорта  // Возвращайте соответствующий ресурс в зависимости от названия сорта  if (sortName.equals("Сорт Гала")) {  return R.drawable.*sort\_gala*;  } else if (sortName.equals("Сорт Джонаголд")) {  return R.drawable.*sort\_djonagold*;  } else if (sortName.equals("Сорт Болдуин")) {  return R.drawable.*sort\_bolduin*;  } else if (sortName.equals("Сорт Анжуйский")) {  return R.drawable.*sort\_anjuskiy*;  } else if (sortName.equals("Сорт Азиатский")) {  return R.drawable.*sort\_azian*;  } else if (sortName.equals("Сорт Кавендиш")) {  return R.drawable.*sort\_kavendish*;  } else if (sortName.equals("Сорт Бурро")) {  return R.drawable.*sort\_burro*;  } else if (sortName.equals("Сорт Нино")) {  return R.drawable.*sort\_nino*;  }  return R.drawable.*ic\_launcher\_background*;  }  } |

Теперь реализуем 4-ую активность, в которой нам необходимо сделать скроллинг вводимого текста (см. Листинг 6).

Листинг 6. FourthActivity.java

|  |
| --- |
| package com.example.nekrasovglebandreevich\_5practpart1;  import android.content.Intent; import android.os.Bundle; import android.text.Editable; import android.text.TextWatcher; import android.view.View; import android.widget.Button; import android.widget.EditText; import android.widget.LinearLayout; import android.widget.TextView;  import androidx.appcompat.app.AppCompatActivity;  public class FourthActivity extends AppCompatActivity {   private LinearLayout sortLayout;  private EditText editText;   @Override  protected void onCreate(Bundle savedInstanceState) {  super.onCreate(savedInstanceState);  setContentView(R.layout.*activity\_fourth*);   sortLayout = findViewById(R.id.*sortLayout*);  editText = findViewById(R.id.*editText*);  Button goToFiveActivityButton = findViewById(R.id.*goToFiveActivityButton*);   // Устанавливаем слушатель изменений текста на EditText  editText.addTextChangedListener(new TextWatcher() {  @Override  public void beforeTextChanged(CharSequence s, int start, int count, int after) {}   @Override  public void onTextChanged(CharSequence s, int start, int before, int count) {}   @Override  public void afterTextChanged(Editable s) {  // Получаем текст из EditText  String newSort = s.toString().trim();  if (!newSort.isEmpty()) {  // Создаем новый элемент TextView  TextView textView = new TextView(FourthActivity.this);  textView.setText(newSort);  // Добавляем новый элемент в LinearLayout  sortLayout.addView(textView);  // Очищаем EditText после добавления  editText.setText("");  }  }  });   // Устанавливаем слушатель клика на кнопку для перехода на пятую активность  goToFiveActivityButton.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View v) {  Intent intent = new Intent(FourthActivity.this, FifthActivity.class);  startActivity(intent);  }  });  } } |

Ну и наконец, реализуем пятую активность в виде списка выбора (см. Листинг 7).

Листинг 7. FifthActivity.java

|  |
| --- |
| package com.example.nekrasovglebandreevich\_5practpart1;  import android.os.Bundle; import android.view.View; import android.widget.AdapterView; import android.widget.ArrayAdapter; import android.widget.Spinner; import android.widget.Toast;  import androidx.appcompat.app.AppCompatActivity;  public class FifthActivity extends AppCompatActivity {   @Override  protected void onCreate(Bundle savedInstanceState) {  super.onCreate(savedInstanceState);  setContentView(R.layout.*activity\_fifth*);   // Получаем ссылку на Spinner из макета  Spinner spinner = findViewById(R.id.*spinner*);   // Создаем массив сортов  String[] sorts = {"Сорт 1", "Сорт 2", "Сорт 3", "Сорт 4", "Сорт 5"};   // Создаем ArrayAdapter с использованием массива сортов и стандартного макета для элементов списка  ArrayAdapter<String> adapter = new ArrayAdapter<>(this, android.R.layout.*simple\_spinner\_item*, sorts);   // Устанавливаем макет для выпадающего списка  adapter.setDropDownViewResource(android.R.layout.*simple\_spinner\_dropdown\_item*);   // Применяем адаптер к Spinner  spinner.setAdapter(adapter);   // Устанавливаем слушатель выбора элемента в Spinner  spinner.setOnItemSelectedListener(new AdapterView.OnItemSelectedListener() {  @Override  public void onItemSelected(AdapterView<?> parent, View view, int position, long id) {  // Получаем выбранный элемент  String selectedSort = parent.getItemAtPosition(position).toString();  // Выводим выбранный сорт в виде всплывающего сообщения  Toast.*makeText*(FifthActivity.this, "Выбран сорт: " + selectedSort, Toast.*LENGTH\_SHORT*).show();  }   @Override  public void onNothingSelected(AdapterView<?> parent) {  // Код, который выполняется, если не выбран ни один элемент (в данном случае не требуется)  }  });  } } |

# Тестирование работы

Результаты тестирования работы представлены на рисунках (см. Рисунки 30-34).

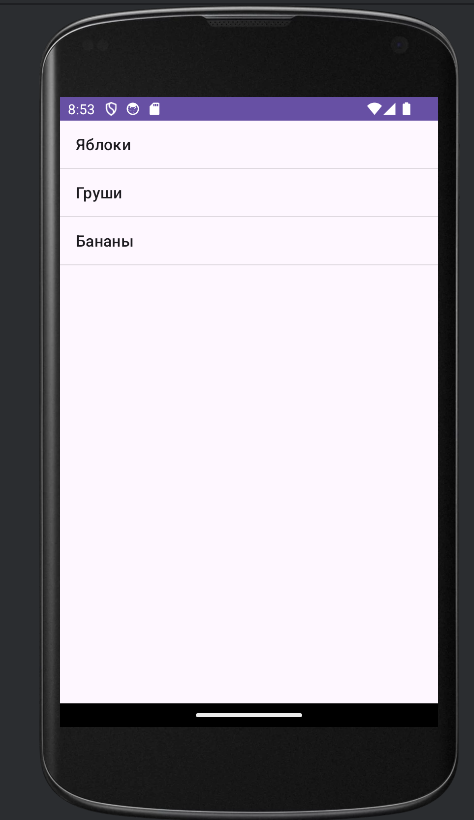


Рисунок 1. Выбор категории

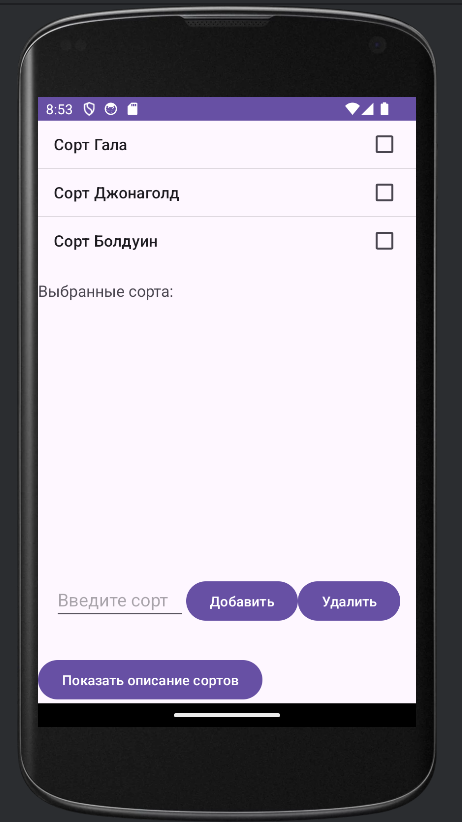


Рисунок 2. После выбора категории яблоки

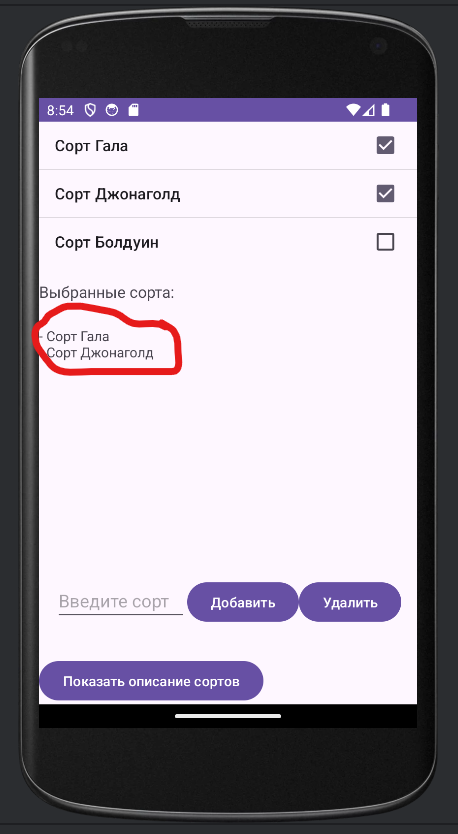


Рисунок 3. Просмотр выбранных сортов

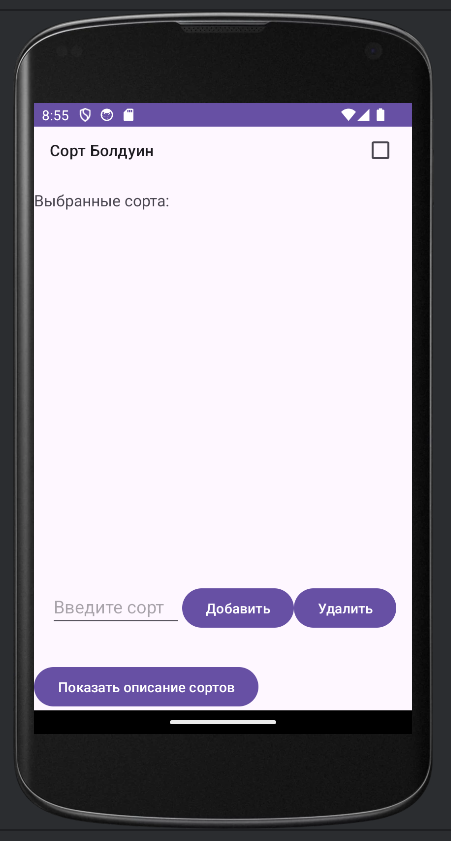


Рисунок 4. Удаление этих сортов кнопкой Удалить

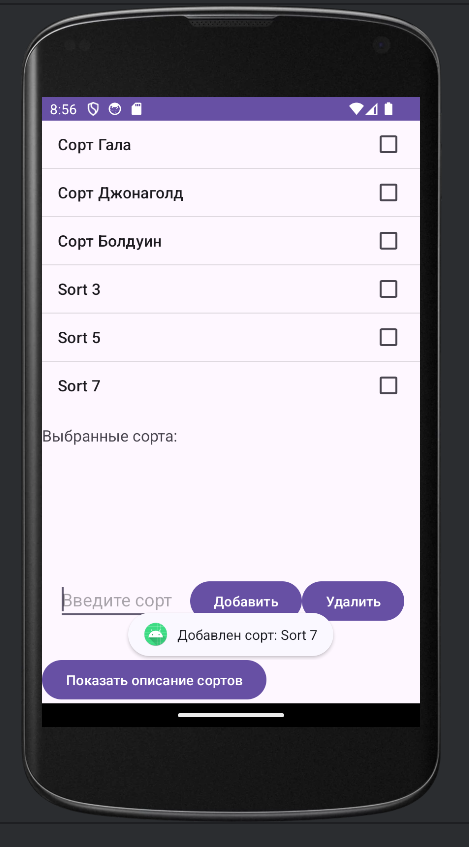


Рисунок 5. Проверка работы кнопки Добавить



Рисунок 6. Проверка кнопки Показать описание сортов (т.е. переход на 3 активность)



Рисунок 7. Проверка скроллинга на 4 активности

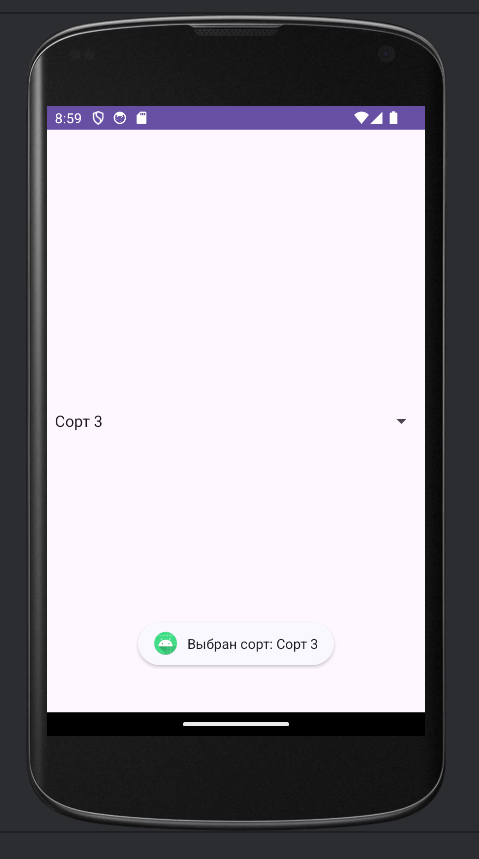


Рисунок 8. Проверка списка выбора сортов