**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ**

**УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ**

**ГОМЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ИМЕНИ П. О. СУХОГО**

Факультет автоматизированных и информационных систем

Кафедра «Информатика»

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ № 3

по дисциплине «Разработка приложений для мобильных устройств»

на тему: «Жизненный цикл Активности, создание пользовательского интерфейса»

Выполнил: студент гр. ИП-42

Суховенко Э.С.

Принял: преподаватель

Фролова Е.В.

Гомель 2022

**Лабораторная работа №3**

**Жизненный цикл Активности, создание пользовательского интерфейса.**

**Цель работы:** изучить методы жизненного цикла активности и получить практические навыки по созданию пользовательского интерфейса.

Лабораторная работа выполнена на языке программирования Java.

**Задание.**

Задание является продолжением задания 3 из лабораторной работы №2.

Разработать Android приложение, выполняющее следующие функции:

* Чтение информации из текстового файла и отображение списка объектов (без подробностей) на главном экране при запуске приложения. При загрузке данных должно появляться диалоговое окно ProgressDialog.
* Вывод подробной информации о выбранном объекте в отдельном фрагменте, который появляется при перелистывании страницы.
* Поиск информации в соответствии с вариантом задания при выборе соответствующего пункта главного меню. Результаты поиска должны отображаться в списочной активности. Ввод необходимых данных для поиска должен производиться в диалоговом окне.
* Добавление или редактирование данных по соответствующей команде главного меню. Ввод данных осуществлять в отдельном экране. Использовать для ввода AutoCompleteTextView.
* Удаление выбранного объекта по команде меню. Перед удалением должно появляться диалоговое окно для подтверждения или отмены удаления.
* Сохранение информации в текстовом файле по команде меню.
* Смена стиля приложения. Выбор темы осуществлять с помощью контекстного меню. Создать не менее двух собственных тем.

**В приложении должны быть реализованы паттерны MVP и Репозиторий.**

**Вариант 12**

Product: id, Наименование, UPC, Производитель, Цена, Срок хранения, Количество.

a) список товаров для заданного наименования;

b) список товаров для заданного наименования, цена которых не превосходит заданную.

**Код программы:**

package com.uni.lab3.activities;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;

import android.os.Bundle;

import android.widget.AutoCompleteTextView;

import android.widget.Button;

import com.uni.lab3.R;

import com.uni.lab3.model.Product;

import com.uni.lab3.themes.Themes;

import com.uni.lab3.themes.ThemesUtils;

public class AddEditForm extends AppCompatActivity {

MainActivity.AddEditFormVariants variant;

AutoCompleteTextView nameAutoComplete;

AutoCompleteTextView upcAutoComplete;

AutoCompleteTextView producerAutoComplete;

AutoCompleteTextView priceAutoComplete;

AutoCompleteTextView expirationDateAutoComplete;

AutoCompleteTextView countAutoComplete;

Product product = new Product();

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

if (getIntent() != null) {

Themes theme = (Themes) getIntent().getSerializableExtra(MainActivity.*CURRENT\_THEME*);

setTheme(ThemesUtils.*getThemeId*(theme));

}

setContentView(R.layout.*activity\_add\_edit\_form*);

variant = (MainActivity.AddEditFormVariants) getIntent().getSerializableExtra(MainActivity.*ADD\_EDIT\_VARIANT*);

Button addEditButton = findViewById(R.id.*addEditButton*);

addEditButton.setText(variant == MainActivity.AddEditFormVariants.*ADD* ? "add" : "edit");

nameAutoComplete = findViewById(R.id.*nameAutoComplete*);

upcAutoComplete = findViewById(R.id.*upcAutoComplete*);

producerAutoComplete = findViewById(R.id.*producerAutoComplete*);

priceAutoComplete = findViewById(R.id.*priceAutoComplete*);

expirationDateAutoComplete = findViewById(R.id.*expirationDateAutoComplete*);

countAutoComplete = findViewById(R.id.*countAutoComplete*);

if (getIntent().getSerializableExtra(MainActivity.*PRODUCT*) != null) {

product = (Product) getIntent().getSerializableExtra(MainActivity.*PRODUCT*);

}

nameAutoComplete.setText(product.getName());

upcAutoComplete.setText(String.*valueOf*(product.getUniversalProductCode()));

producerAutoComplete.setText(product.getProducer());

priceAutoComplete.setText(String.*valueOf*(product.getPrice()));

expirationDateAutoComplete.setText(product.getExpirationDate());

countAutoComplete.setText(String.*valueOf*(product.getCount()));

addEditButton.setOnClickListener(view -> onSubmit());

}

public void onSubmit() {

product.setName(nameAutoComplete.getText().toString());

product.setUniversalProductCode(Integer.*parseInt*(upcAutoComplete.getText().toString()));

product.setProducer(producerAutoComplete.getText().toString());

product.setPrice(Integer.*parseInt*(priceAutoComplete.getText().toString()));

product.setExpirationDate(expirationDateAutoComplete.getText().toString());

product.setCount(Integer.*parseInt*(countAutoComplete.getText().toString()));

Intent resultIntent = new Intent();

resultIntent.putExtra(MainActivity.*PRODUCT*, product);

final int resultId = variant == MainActivity.AddEditFormVariants.*ADD*

? MainActivity.*ADD\_PRODUCT\_RESULT*

: MainActivity.*EDIT\_PRODUCT\_RESULT*;

this.setResult(resultId, resultIntent);

this.finish();

}

}

package com.uni.lab3.activities;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;

import android.os.Bundle;

import android.text.Editable;

import android.text.TextWatcher;

import android.widget.Button;

import android.widget.EditText;

import com.uni.lab3.R;

import com.uni.lab3.themes.Themes;

import com.uni.lab3.themes.ThemesUtils;

public class ConstrainedSearch extends AppCompatActivity {

private String currentName = "";

private String lastMaxPriceString = "";

private int currentMaxPrice = 0;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

if (getIntent() != null) {

Themes theme = (Themes) getIntent().getSerializableExtra(MainActivity.CURRENT\_THEME);

setTheme(ThemesUtils.getThemeId(theme));

}

setContentView(R.layout.activity\_constrained\_search);

EditText nameInput = findViewById(R.id.nameInput);

EditText maxPriceInput = findViewById(R.id.maxPriceInput);

nameInput.addTextChangedListener(new TextWatcher() {

@Override

public void beforeTextChanged(CharSequence charSequence, int i, int i1, int i2) {}

@Override

public void onTextChanged(CharSequence charSequence, int i, int i1, int i2) {

currentName = charSequence.toString();

}

@Override

public void afterTextChanged(Editable editable) {}

});

maxPriceInput.addTextChangedListener(new TextWatcher() {

@Override

public void beforeTextChanged(CharSequence charSequence, int i, int i1, int i2) {

if (isInteger(charSequence.toString())) {

lastMaxPriceString = charSequence.toString();

currentMaxPrice = Integer.parseInt(charSequence.toString());

}

}

@Override

public void onTextChanged(CharSequence charSequence, int i, int i1, int i2) {

if (!isInteger(charSequence.toString()) && charSequence.length() > 0) {

int previousSelection = maxPriceInput.getSelectionEnd() - 1;

maxPriceInput.setText(lastMaxPriceString);

maxPriceInput.setSelection(Math.min(previousSelection, lastMaxPriceString.length()));

} else {

lastMaxPriceString = charSequence.toString();

currentMaxPrice = Integer.parseInt(charSequence.toString());

}

if (charSequence.length() == 0) {

lastMaxPriceString = "";

currentMaxPrice = 0;

}

}

@Override

public void afterTextChanged(Editable editable) {}

});

Button constrainedSearchButton = findViewById(R.id.constrainedSearchButton);

constrainedSearchButton.setOnClickListener(view -> {

this.setResult(MainActivity.CONSTRAINED\_SEARCH\_RESULT, getResultIntent());

this.finish();

});

}

public Intent getResultIntent() {

Intent resultIntent = new Intent();

resultIntent.putExtra("name", currentName);

resultIntent.putExtra("maxPrice", currentMaxPrice);

return resultIntent;

}

public static boolean isInteger(String strNum) {

if (strNum == null) {

return false;

}

try {

double d = Integer.parseInt(strNum);

} catch (NumberFormatException e) {

return false;

}

return true;

}

}

package com.uni.lab3.activities;

import static com.uni.lab3.themes.Themes.\*;

import android.annotation.SuppressLint;

import android.app.SearchManager;

import android.content.Intent;

import android.os.Bundle;

import androidx.annotation.NonNull;

import androidx.annotation.Nullable;

import androidx.appcompat.app.AppCompatActivity;

import androidx.fragment.app.DialogFragment;

import androidx.fragment.app.Fragment;

import androidx.fragment.app.FragmentTransaction;

import com.uni.lab3.IO.productsWriter.ProductsWriter;

import com.uni.lab3.R;

import com.uni.lab3.fragments.DeleteAlertDialog;

import com.uni.lab3.fragments.FullProductInfoFragment;

import com.uni.lab3.fragments.ProductSelectorDialog;

import com.uni.lab3.model.Product;

import com.uni.lab3.model.ProductsRepository;

import com.uni.lab3.IO.productsReader.ProductsReader;

import com.uni.lab3.themes.Themes;

import com.uni.lab3.themes.ThemesUtils;

import android.util.Log;

import android.view.Menu;

import android.view.MenuItem;

import android.widget.ArrayAdapter;

import android.widget.ListView;

import android.widget.TextView;

import java.io.BufferedReader;

import java.io.BufferedWriter;

import java.io.IOException;

import java.io.InputStreamReader;

import java.io.OutputStreamWriter;

import java.util.Arrays;

import java.util.Locale;

public class MainActivity extends AppCompatActivity implements DeleteAlertDialog.DialogListener, ProductSelectorDialog.DialogListener {

public enum SelectProductIdReasons {

DELETE,

UPDATE

}

public enum AddEditFormVariants {

ADD,

EDIT

}

public final static String ADD\_EDIT\_VARIANT = "add\_edit\_variant";

public final static String PRODUCTS\_REPOSITORY = "productsRepository";

public final static String CURRENT\_THEME = "currentTheme";

public final static String REASON = "reason";

public final static String PRODUCT = "product";

public final static int CONSTRAINED\_SEARCH\_RESULT = 1;

public final static int ADD\_PRODUCT\_RESULT = 2;

public final static int EDIT\_PRODUCT\_RESULT = 3;

private ProductsRepository productsRepository;

private Themes currentTheme = GREEN;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

if (savedInstanceState != null) {

productsRepository = (ProductsRepository) savedInstanceState.getSerializable(PRODUCTS\_REPOSITORY);

currentTheme = (Themes) savedInstanceState.getSerializable(CURRENT\_THEME);

}

setTheme(ThemesUtils.getThemeId(currentTheme));

setContentView(R.layout.activity\_main);

ListView productsListView = findViewById(R.id.productsListView);

productsListView.setOnItemClickListener((adapterView, view, i, l) -> {

Fragment previousFragment = getFullInfoProductFragment();

if (previousFragment != null) {

removeFullProductInfoFragment(previousFragment);

}

Bundle bundle = new Bundle();

bundle.putSerializable(PRODUCT, productsRepository.getByIndex(i));

getSupportFragmentManager().beginTransaction()

.setReorderingAllowed(true)

.add(R.id.fragmentContainerView, FullProductInfoFragment.class, bundle, FullProductInfoFragment.FRAGMENT\_TAG)

.commit();

});

if (productsRepository != null) {

setProductsInListView(productsRepository.getAll());

} else {

loadProducts();

}

}

@Override

public boolean onCreateOptionsMenu(Menu menu) {

getMenuInflater().inflate(R.menu.navigation\_menu, menu);

return true;

}

@SuppressLint("NonConstantResourceId")

@Override

public boolean onOptionsItemSelected(MenuItem item) {

switch (item.getItemId()) {

case R.id.load\_products: {

loadProducts();

return true;

}

case R.id.save\_products: {

saveProducts();

return true;

}

case R.id.search:

onSearchRequested();

return true;

case R.id.search\_with\_price: {

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

intent.putExtra(CURRENT\_THEME, currentTheme);

startActivityForResult(intent, CONSTRAINED\_SEARCH\_RESULT);

return true;

}

case R.id.get\_all: {

applySearchQuery("");

setMaxPriceQueryText("");

return true;

}

*// actions*

case R.id.create\_product: {

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

intent.putExtra(CURRENT\_THEME, currentTheme);

intent.putExtra(ADD\_EDIT\_VARIANT, AddEditFormVariants.ADD);

startActivityForResult(intent, ADD\_PRODUCT\_RESULT);

return true;

}

case R.id.update\_product: {

showProductSelectDialog(SelectProductIdReasons.UPDATE);

return true;

}

case R.id.delete\_product: {

showProductSelectDialog(SelectProductIdReasons.DELETE);

return true;

}

*// themes*

case R.id.default\_theme: {

currentTheme = DEFAULT;

recreate();

return true;

}

case R.id.red\_theme: {

currentTheme = RED;

recreate();

return true;

}

case R.id.green\_theme: {

currentTheme = GREEN;

recreate();

return true;

}

}

return super.onOptionsItemSelected(item);

}

@Override

protected void onActivityResult(int requestCode, int resultCode, @Nullable Intent data) {

super.onActivityResult(requestCode, resultCode, data);

if (data == null) return;

if (requestCode == CONSTRAINED\_SEARCH\_RESULT) {

String name = data.getStringExtra("name");

int maxPrice = data.getIntExtra("maxPrice", 0);

applySearchWithPriceQuery(name, maxPrice);

}

if (data.getSerializableExtra(PRODUCT) == null) return;

Product product = (Product) data.getSerializableExtra(PRODUCT);

switch (requestCode) {

case ADD\_PRODUCT\_RESULT: {

productsRepository.add(product);

break;

}

case EDIT\_PRODUCT\_RESULT: {

productsRepository.update(product);

break;

}

default:

throw new IllegalStateException("Unexpected value: " + requestCode);

}

setProductsInListView(productsRepository.getAll());

}

@Override

public void onSaveInstanceState(@NonNull Bundle savedInstanceState) {

super.onSaveInstanceState(savedInstanceState);

savedInstanceState.putSerializable(PRODUCTS\_REPOSITORY, productsRepository);

savedInstanceState.putSerializable(CURRENT\_THEME, currentTheme);

}

@Override

protected void onNewIntent(Intent intent) {

super.onNewIntent(intent);

if (Intent.ACTION\_SEARCH.equals(intent.getAction())) {

String query = intent.getStringExtra(SearchManager.QUERY);

applySearchQuery(query.toLowerCase(Locale.ROOT));

}

}

public void setProductsInListView(Product[] products) {

ArrayAdapter<String> arrayAdapter = new ArrayAdapter<>(this, R.layout.list\_item, Arrays.stream(products).map(Product::toShortString).toArray(String[]::new));

ListView productsListView = findViewById(R.id.productsListView);

productsListView.setAdapter(arrayAdapter);

}

public Fragment getFullInfoProductFragment() {

return getSupportFragmentManager().findFragmentByTag(FullProductInfoFragment.FRAGMENT\_TAG);

}

public void removeFullProductInfoFragment(Fragment fragment) {

getSupportFragmentManager().beginTransaction()

.remove(fragment)

.commit();

}

public void applySearchQuery(String searchQuery) {

setSearchQueryText(searchQuery);

setProductsInListView(Arrays.stream(productsRepository.getAll()).filter(product -> product.getName().toLowerCase(Locale.ROOT).contains(searchQuery)).toArray(Product[]::new));

}

public void applySearchWithPriceQuery(String name, int maxPrice) {

setSearchQueryText(name);

setMaxPriceQueryText(Integer.toString(maxPrice));

setProductsInListView(Arrays.stream(productsRepository.getAll()).filter(product ->

product.getName().toLowerCase(Locale.ROOT).contains(name.toLowerCase(Locale.ROOT))

&& product.getPrice() <= maxPrice)

.toArray(Product[]::new)

);

}

public void setSearchQueryText(String text) {

TextView searchQueryTextView = findViewById(R.id.searchQueryTextView);

if (text.isEmpty()) {

searchQueryTextView.setText("");

} else {

searchQueryTextView.setText(String.*format*("Name: %s", text));

}

}

public void setMaxPriceQueryText(String text) {

TextView maxPriceTextView = findViewById(R.id.*maxPriceTextView*);

if (text.isEmpty()) {

maxPriceTextView.setText("");

} else {

maxPriceTextView.setText(String.*format*("Max price: %s", text));

}

}

private void loadProducts() {

try {

ProductsReader productsReader = new ProductsReader(

MainActivity.this,

new BufferedReader(new InputStreamReader(getAssets().open("products.json"))

),

(products) -> {

productsRepository = new ProductsRepository(products);

setProductsInListView(products);

});

productsReader.execute();

} catch (IOException e) {

e.printStackTrace();

}

}

private void saveProducts() {

try {

ProductsWriter productsWriter = new ProductsWriter(

MainActivity.this,

new BufferedWriter(new OutputStreamWriter(this.openFileOutput("products.json", MODE\_PRIVATE))

),

productsRepository.getAll(),

(result) -> {});

productsWriter.execute();

} catch (IOException e) {

e.printStackTrace();

}

}

@Override

public void onConfirmDeleteProductId(int productId) {

productsRepository.removeById(productId);

setProductsInListView(productsRepository.getAll());

}

public void removeDeleteDialogFragments() {

FragmentTransaction ft = getSupportFragmentManager().beginTransaction();

DialogFragment selectProductIdDialog = (DialogFragment) getSupportFragmentManager().findFragmentByTag(DeleteAlertDialog.SELECT\_DIALOG);

DialogFragment confirmationDialog = (DialogFragment) getSupportFragmentManager().findFragmentByTag(DeleteAlertDialog.DELETE\_CONFIRMATION\_DIALOG);

if (selectProductIdDialog != null) {

selectProductIdDialog.dismiss();

}

if (confirmationDialog != null) {

confirmationDialog.dismiss();

}

ft.commit();

}

@Override

public void onSelectProductId(int selectedProductId, SelectProductIdReasons reason) {

switch (reason) {

case UPDATE: {

FragmentTransaction ft = getSupportFragmentManager().beginTransaction();

DialogFragment selectProductIdDialog = (DialogFragment) getSupportFragmentManager().findFragmentByTag(DeleteAlertDialog.SELECT\_DIALOG);

if (selectProductIdDialog != null) {

selectProductIdDialog.dismiss();

}

ft.commit();

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

intent.putExtra(CURRENT\_THEME, currentTheme);

intent.putExtra(ADD\_EDIT\_VARIANT, AddEditFormVariants.EDIT);

intent.putExtra(PRODUCT, productsRepository.getById(selectedProductId));

startActivityForResult(intent, EDIT\_PRODUCT\_RESULT);

break;

}

case DELETE: {

DeleteAlertDialog deleteAlertDialog = new DeleteAlertDialog();

Bundle bundle = new Bundle();

bundle.putInt(DeleteAlertDialog.PRODUCT\_ID, selectedProductId);

bundle.putString(DeleteAlertDialog.ALERT\_TITLE, "Confirm deleting");

bundle.putString(DeleteAlertDialog.ALERT\_MESSAGE, "Delete product with id:");

deleteAlertDialog.setArguments(bundle);

FragmentTransaction ft = getSupportFragmentManager().beginTransaction();

deleteAlertDialog.show(ft, DeleteAlertDialog.DELETE\_CONFIRMATION\_DIALOG);

break;

}

default:

throw new IllegalStateException("Unexpected value: " + reason);

}

}

public void showProductSelectDialog(SelectProductIdReasons reason) {

if (productsRepository.length() == 0) return;

ProductSelectorDialog dialogFragment = new ProductSelectorDialog();

Bundle bundle = new Bundle();

bundle.putSerializable(REASON, reason);

bundle.putSerializable(ProductSelectorDialog.PRODUCT\_IDS, Arrays.stream(productsRepository.getAll()).map(Product::getId).mapToInt(i -> i).toArray());

dialogFragment.setArguments(bundle);

FragmentTransaction ft = getSupportFragmentManager().beginTransaction();

ft.addToBackStack(null);

dialogFragment.show(ft, ProductSelectorDialog.SELECT\_DIALOG);

}

}

package com.uni.lab3.fragments;

import android.app.AlertDialog;

import android.app.Dialog;

import android.os.Bundle;

import androidx.annotation.NonNull;

import androidx.annotation.Nullable;

import androidx.fragment.app.DialogFragment;

import android.view.LayoutInflater;

import android.view.View;

import android.view.ViewGroup;

import com.uni.lab3.R;

public class DeleteAlertDialog extends DialogFragment {

public static String *NOT\_ALERT\_DIALOG* = "notAlertDialog";

public static String *FULLSCREEN* = "fullScreen";

public static String *ALERT\_TITLE* = "alertTitle";

public static String *ALERT\_MESSAGE* = "alertMessage";

public static String *PRODUCT\_ID* = "productId";

public static String *PRODUCT\_IDS* = "productIds";

public static String *OK* = "Ok";

public static String *CANCEL* = "Cancel";

public static String *SELECT\_DIALOG* = "selectDialog";

public static String *DELETE\_CONFIRMATION\_DIALOG* = "deleteConfirmationDialog";

DialogListener dialogListener;

@NonNull

@Override

public Dialog onCreateDialog(@Nullable Bundle savedInstanceState) {

dialogListener = (DialogListener) getActivity();

AlertDialog.Builder builder = new AlertDialog.Builder(getActivity());

if (getArguments() != null) {

builder.setTitle(getArguments().getString(*ALERT\_TITLE*));

builder.setMessage(getArguments().getString(*ALERT\_MESSAGE*) + " " + getArguments().getInt(*PRODUCT\_ID*));

builder.setPositiveButton(*OK*, (dialog, which) -> {

dialogListener.onConfirmDeleteProductId(getArguments().getInt(*PRODUCT\_ID*));

dialogListener.removeDeleteDialogFragments();

});

builder.setNegativeButton(*CANCEL*, (dialog, which) -> dismiss());

}

return builder.create();

}

@Override

public View onCreateView(LayoutInflater inflater, ViewGroup container,

Bundle savedInstanceState) {

return inflater.inflate(R.layout.*fragment\_delete\_dialog*, container, false);

}

public interface DialogListener {

void onConfirmDeleteProductId(int productId);

void removeDeleteDialogFragments();

}

}

package com.uni.lab3.fragments;

import android.os.Bundle;

import androidx.annotation.NonNull;

import androidx.annotation.Nullable;

import androidx.fragment.app.Fragment;

import android.view.LayoutInflater;

import android.view.View;

import android.view.ViewGroup;

import android.widget.Button;

import android.widget.TextView;

import com.uni.lab3.activities.MainActivity;

import com.uni.lab3.R;

import com.uni.lab3.model.Product;

public class FullProductInfoFragment extends Fragment {

public static String *FRAGMENT\_TAG* = "fullProductInfoFragment";

public FullProductInfoFragment() {}

@Override

public void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

}

@Override

public void onViewCreated(@NonNull View view, @Nullable Bundle savedInstanceState) {

super.onViewCreated(view, savedInstanceState);

MainActivity mainActivity = (MainActivity) getActivity();

Bundle arguments = getArguments();

if (mainActivity != null && arguments != null) {

Product product = (Product) getArguments().getSerializable("product");

TextView fullProductInfoTextView = view.findViewById(R.id.*fullProductInfoTextView*);

fullProductInfoTextView.setText(product.toString());

Button clearButton = view.findViewById(R.id.*clearButton*);

clearButton.setOnClickListener((v) -> mainActivity.removeFullProductInfoFragment(this));

}

}

@Override

public View onCreateView(LayoutInflater inflater, ViewGroup container,

Bundle savedInstanceState) {

*// Inflate the layout for this fragment*

return inflater.inflate(R.layout.*fragment\_full\_product\_info*, container, false);

}

}

package com.uni.lab3.fragments;

import android.app.Dialog;

import android.os.Bundle;

import android.view.LayoutInflater;

import android.view.View;

import android.view.ViewGroup;

import android.widget.ArrayAdapter;

import android.widget.Button;

import android.widget.Spinner;

import androidx.annotation.NonNull;

import androidx.annotation.Nullable;

import androidx.fragment.app.DialogFragment;

import com.uni.lab3.R;

import com.uni.lab3.activities.MainActivity;

import java.util.Arrays;

public class ProductSelectorDialog extends DialogFragment {

public static String *PRODUCT\_IDS* = "productIds";

public static String *SELECT\_DIALOG* = "selectDialog";

ProductSelectorDialog.DialogListener dialogListener;

MainActivity.SelectProductIdReasons reason;

@NonNull

@Override

public Dialog onCreateDialog(@Nullable Bundle savedInstanceState) {

dialogListener = (ProductSelectorDialog.DialogListener) getActivity();

return super.onCreateDialog(savedInstanceState);

}

@Override

public void onViewCreated(@NonNull View view, @Nullable Bundle savedInstanceState) {

super.onViewCreated(view, savedInstanceState);

final Spinner productIdSpinner = view.findViewById(R.id.*productIdSpinner*);

final Button selectProductButton = view.findViewById(R.id.*selectProductButton*);

final Button cancelSelectButton = view.findViewById(R.id.*cancelSelectButton*);

if (getArguments() != null

&& getArguments().getSerializable(*PRODUCT\_IDS*) != null

&& getArguments().getSerializable(MainActivity.*REASON*) != null) {

reason = (MainActivity.SelectProductIdReasons) getArguments().getSerializable(MainActivity.*REASON*);

String[] productIds = Arrays.*stream*(((int[]) getArguments().getSerializable(*PRODUCT\_IDS*))).mapToObj(Integer::*toString*).toArray(String[]::new);

ArrayAdapter<String> arrayAdapter = new ArrayAdapter<>(getContext(), R.layout.*list\_item*, productIds);

productIdSpinner.setAdapter(arrayAdapter);

*// select dialog*

selectProductButton.setOnClickListener(view1 -> dialogListener.onSelectProductId(Integer.*parseInt*(productIdSpinner.getSelectedItem().toString()), reason));

cancelSelectButton.setOnClickListener(view12 -> dismiss());

}

}

@Override

public View onCreateView(LayoutInflater inflater, ViewGroup container,

Bundle savedInstanceState) {

return inflater.inflate(R.layout.*fragment\_select\_dialog*, container, false);

}

public interface DialogListener {

void onSelectProductId(int selectedProductId, MainActivity.SelectProductIdReasons reason);

}

}

package com.uni.lab3.IO.productsReader;

import android.app.ProgressDialog;

import android.content.Context;

import android.os.AsyncTask;

import com.google.gson.Gson;

import com.uni.lab3.model.Product;

import java.io.BufferedReader;

import java.io.IOException;

public class ProductsReader extends AsyncTask<Void, Void, Product[]> {

private final ProgressDialog progressDialog;

private final BufferedReader bufferedReader;

private final ProductsReaderHandlerCallback onStopHandler;

public ProductsReader(Context context, BufferedReader bufferedReader, ProductsReaderHandlerCallback onStopHandler) {

progressDialog = new ProgressDialog(context);

this.bufferedReader = bufferedReader;

this.onStopHandler = onStopHandler;

}

@Override

protected void onPreExecute() {

super.onPreExecute();

progressDialog.setTitle("Please wait");

progressDialog.setMessage("Data is loading");

progressDialog.show();

}

@Override

protected Product[] doInBackground(Void... arg0) {

StringBuilder stringBuilder = new StringBuilder();

try {

String line;

while ((line = bufferedReader.readLine()) != null) {

stringBuilder.append(line);

}

Thread.sleep(1500);

} catch (IOException | InterruptedException e) {

e.printStackTrace();

}

String text = stringBuilder.toString();

return new Gson().fromJson(text, Product[].class);

}

@Override

protected void onPostExecute(Product[] result) {

super.onPostExecute(result);

onStopHandler.handleProductsAfterLoading(result);

progressDialog.dismiss();

}

}

package com.uni.lab3.IO.productsReader;

import com.uni.lab3.model.Product;

public interface ProductsReaderHandlerCallback {

void handleProductsAfterLoading(Product[] products);

}

package com.uni.lab3.IO.productsWriter;

import android.app.ProgressDialog;

import android.content.Context;

import android.os.AsyncTask;

import com.google.gson.Gson;

import com.uni.lab3.model.Product;

import java.io.BufferedWriter;

import java.io.IOException;

public class ProductsWriter extends AsyncTask<Void, Void, Boolean> {

private final ProgressDialog progressDialog;

private final BufferedWriter bufferedWriter;

private final Product[] products;

private final ProductsWriterHandlerCallback onStopHandler;

public ProductsWriter(Context context, BufferedWriter bufferedWriter, Product[] products, ProductsWriterHandlerCallback onStopHandler) {

progressDialog = new ProgressDialog(context);

this.bufferedWriter = bufferedWriter;

this.products = products;

this.onStopHandler = onStopHandler;

}

@Override

protected void onPreExecute() {

super.onPreExecute();

progressDialog.setTitle("Please wait");

progressDialog.setMessage("Data is saving");

progressDialog.show();

}

@Override

protected Boolean doInBackground(Void... arg0) {

try {

bufferedWriter.write(new Gson().toJson(products));

bufferedWriter.flush();

Thread.*sleep*(1500);

} catch (IOException | InterruptedException e) {

e.printStackTrace();

return false;

}

return true;

}

@Override

protected void onPostExecute(Boolean result) {

super.onPostExecute(result);

onStopHandler.handleResultAfterSaving(result);

progressDialog.dismiss();

}

}

package com.uni.lab3.model;

import androidx.annotation.NonNull;

import com.google.gson.annotations.SerializedName;

import java.io.Serializable;

public class Product implements Serializable {

@SerializedName("id")

private int id;

@SerializedName("name")

private String name;

@SerializedName("universalProductCode")

private int universalProductCode;

@SerializedName("producer")

private String producer;

@SerializedName("price")

private int price;

@SerializedName("expirationDate")

private String expirationDate;

@SerializedName("count")

private int count;

public Product() {

this.name = "Sample name";

this.universalProductCode = 0;

this.producer = "Sample producer";

this.price = 0;

this.expirationDate = "01.01.2000";

this.count = 0;

}

public Product(String name, int universalProductCode, String producer, int price, String expirationDate, int count) {

this.name = name;

this.universalProductCode = universalProductCode;

this.producer = producer;

this.price = price;

this.expirationDate = expirationDate;

this.count = count;

}

@NonNull

public Product clone() {

Product product = new Product(

getName(),

getUniversalProductCode(),

getProducer(),

getPrice(),

getExpirationDate(),

getCount()

);

product.setId(getId());

return product;

}

public int getId() {

return id;

}

public void setId(int id) {

this.id = id;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public int getUniversalProductCode() {

return universalProductCode;

}

public void setUniversalProductCode(int universalProductCode) {

this.universalProductCode = universalProductCode;

}

public String getProducer() {

return producer;

}

public void setProducer(String producer) {

this.producer = producer;

}

public int getPrice() {

return price;

}

public void setPrice(int price) {

this.price = price;

}

public String getExpirationDate() {

return expirationDate;

}

public void setExpirationDate(String expirationDate) {

this.expirationDate = expirationDate;

}

public int getCount() {

return count;

}

public void setCount(int count) {

this.count = count;

}

public String toShortString() {

return "Product{" +

"id=" + id +

", name='" + name + '\'' +

", price=" + price +

'}';

}

@NonNull

@Override

public String toString() {

return "Product{" +

"id=" + id +

", name='" + name + '\'' +

", universalProductCode=" + universalProductCode +

", producer='" + producer + '\'' +

", price=" + price +

", expirationDate='" + expirationDate + '\'' +

", count=" + count +

'}';

}

}

package com.uni.lab3.model;

import androidx.annotation.NonNull;

import java.io.Serializable;

import java.util.Arrays;

import java.util.LinkedList;

import java.util.List;

import java.util.stream.Collectors;

import java.util.stream.IntStream;

public class ProductsRepository implements Serializable {

private List<Product> products;

public ProductsRepository(Product[] products) {

this.products = new LinkedList<>(Arrays.*asList*(products));

}

public Product[] getAll() {

return products.stream().toArray(Product[]::new);

}

public Product getByIndex(int index) {

return products.get(index).clone();

}

public Product getById(int id) {

return products.stream().filter(product -> product.getId() == id).findFirst().get().clone();

}

public void add(Product product) {

product.setId(getNextId());

addByIndex(length(), product);

}

public void addByIndex(int index, Product product) {

products.add(index, product);

}

public void update(Product product) {

int index = IntStream.*range*(0, products.size())

.filter(i -> product.getId() == products.get(i).getId())

.findFirst().orElse(-1);

remove(product);

products.add(index, product);

}

public void remove(@NonNull Product product) {

removeById(product.getId());

}

public void removeById(int id) {

products = products.stream().filter(product -> product.getId() != id).collect(Collectors.*toList*());

}

public int getNextId() {

return Math.*max*(0, products.stream().map(Product::getId).mapToInt(i -> i).max().orElse(0)) + 1;

}

public int length() {

return products.size();

}

}

package com.uni.lab3.themes;

public enum Themes {

*RED*,

*GREEN*,

*DEFAULT*

}

package com.uni.lab3.themes;

import com.uni.lab3.R;

public class ThemesUtils {

public static int getThemeId(Themes theme) {

switch (theme) {

case *RED*: {

return R.style.*Theme\_Lab3\_red*;

}

case *GREEN*: {

return R.style.*Theme\_Lab3\_green*;

}

default: {

return R.style.*Theme\_Lab3*;

}

}

}

}

package com.uni.lab3.IO.productsWriter;

public interface ProductsWriterHandlerCallback {

void handleResultAfterSaving(boolean result);

}

*<?*xml version="1.0" encoding="utf-8"*?>*

<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:app="http://schemas.android.com/apk/res-auto"

xmlns:tools="http://schemas.android.com/tools"

android:id="@+id/constraintLayout"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent">

<AutoCompleteTextView

android:id="@+id/upcAutoComplete"

android:layout\_width="195dp"

android:layout\_height="48dp"

android:layout\_marginEnd="76dp"

android:minHeight="48dp"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintTop\_toBottomOf="@+id/nameAutoComplete"

tools:ignore="DuplicateSpeakableTextCheck" />

<AutoCompleteTextView

android:id="@+id/nameAutoComplete"

android:layout\_width="195dp"

android:layout\_height="48dp"

android:layout\_marginTop="120dp"

android:layout\_marginEnd="76dp"

android:minHeight="48dp"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintTop\_toTopOf="parent"

tools:ignore="DuplicateSpeakableTextCheck" />

<TextView

android:id="@+id/upcLabelForAutoComplete"

android:layout\_width="100dp"

android:layout\_height="48dp"

android:gravity="center"

android:labelFor="@id/upcAutoComplete"

android:text="@string/universal\_product\_code"

app:layout\_constraintEnd\_toStartOf="@+id/upcAutoComplete"

app:layout\_constraintTop\_toBottomOf="@+id/nameLabelForAutoComplete" />

<TextView

android:id="@+id/nameLabelForAutoComplete"

android:layout\_width="100dp"

android:layout\_height="48dp"

android:gravity="center"

android:labelFor="@id/nameAutoComplete"

android:text="@string/name"

app:layout\_constraintEnd\_toStartOf="@+id/nameAutoComplete"

app:layout\_constraintTop\_toTopOf="@+id/nameAutoComplete" />

<AutoCompleteTextView

android:id="@+id/expirationDateAutoComplete"

android:layout\_width="195dp"

android:layout\_height="48dp"

android:layout\_marginEnd="76dp"

android:minHeight="48dp"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintTop\_toBottomOf="@+id/priceAutoComplete"

tools:ignore="DuplicateSpeakableTextCheck" />

<AutoCompleteTextView

android:id="@+id/priceAutoComplete"

android:layout\_width="195dp"

android:layout\_height="48dp"

android:layout\_marginEnd="76dp"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintTop\_toBottomOf="@+id/producerAutoComplete"

tools:ignore="DuplicateSpeakableTextCheck" />

<AutoCompleteTextView

android:id="@+id/producerAutoComplete"

android:layout\_width="195dp"

android:layout\_height="48dp"

android:layout\_marginEnd="76dp"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintTop\_toBottomOf="@+id/upcAutoComplete"

tools:ignore="DuplicateSpeakableTextCheck" />

<AutoCompleteTextView

android:id="@+id/countAutoComplete"

android:layout\_width="195dp"

android:layout\_height="48dp"

android:layout\_marginEnd="76dp"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintTop\_toBottomOf="@+id/expirationDateAutoComplete"

tools:ignore="DuplicateSpeakableTextCheck" />

<TextView

android:id="@+id/priceLabelForAutoComplete"

android:layout\_width="100dp"

android:layout\_height="48dp"

android:layout\_marginEnd="194dp"

android:gravity="center"

android:labelFor="@id/priceAutoComplete"

android:text="@string/price"

app:layout\_constraintEnd\_toEndOf="@+id/priceAutoComplete"

app:layout\_constraintTop\_toBottomOf="@+id/producerLabelForAutoComplete" />

<TextView

android:id="@+id/producerLabelForAutoComplete"

android:layout\_width="100dp"

android:layout\_height="48dp"

android:gravity="center"

android:labelFor="@id/producerAutoComplete"

android:text="@string/producer"

app:layout\_constraintEnd\_toStartOf="@+id/producerAutoComplete"

app:layout\_constraintTop\_toBottomOf="@+id/upcLabelForAutoComplete" />

<TextView

android:id="@+id/countLabelForAutoComplete"

android:layout\_width="100dp"

android:layout\_height="48dp"

android:layout\_marginEnd="194dp"

android:gravity="center"

android:labelFor="@id/countAutoComplete"

android:text="@string/count"

app:layout\_constraintEnd\_toEndOf="@+id/countAutoComplete"

app:layout\_constraintTop\_toBottomOf="@+id/expirationDateLabelForAutoComplete" />

<TextView

android:id="@+id/expirationDateLabelForAutoComplete"

android:layout\_width="100dp"

android:layout\_height="48dp"

android:layout\_marginEnd="194dp"

android:gravity="center"

android:labelFor="@id/expirationDateAutoComplete"

android:text="@string/expiration\_date"

app:layout\_constraintEnd\_toEndOf="@+id/expirationDateAutoComplete"

app:layout\_constraintTop\_toBottomOf="@+id/priceLabelForAutoComplete" />

<Button

android:id="@+id/addEditButton"

android:layout\_width="100dp"

android:layout\_height="48dp"

android:layout\_marginStart="141dp"

android:layout\_marginTop="100dp"

android:layout\_marginEnd="176dp"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toBottomOf="@+id/countAutoComplete"

tools:ignore="SpeakableTextPresentCheck" />

</androidx.constraintlayout.widget.ConstraintLayout>

*<?*xml version="1.0" encoding="utf-8"*?>*

<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:app="http://schemas.android.com/apk/res-auto"

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

tools:context=".activities.ConstrainedSearch">

<Button

android:id="@+id/constrainedSearchButton"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginTop="160dp"

android:text="@string/search"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toBottomOf="@+id/maxPriceInput" />

<TextView

android:id="@+id/textView2"

android:layout\_width="50dp"

android:layout\_height="48dp"

android:layout\_marginTop="60dp"

android:gravity="center"

android:text="@string/name"

app:layout\_constraintEnd\_toStartOf="@+id/nameInput"

app:layout\_constraintTop\_toBottomOf="@+id/textView"

android:labelFor="@+id/nameInput" />

<TextView

android:id="@+id/textView"

android:layout\_width="113dp"

android:layout\_height="51dp"

android:layout\_marginTop="68dp"

android:gravity="center"

android:text="@string/search\_by\_name\_and\_max\_price"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toTopOf="parent" />

<TextView

android:id="@+id/textView3"

android:layout\_width="50dp"

android:layout\_height="48dp"

android:layout\_marginStart="56dp"

android:layout\_marginTop="28dp"

android:gravity="center"

android:text="@string/max\_price"

app:layout\_constraintEnd\_toStartOf="@+id/maxPriceInput"

app:layout\_constraintHorizontal\_bias="1.0"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toBottomOf="@+id/textView2"

android:labelFor="@+id/maxPriceInput" />

<EditText

android:id="@+id/nameInput"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginTop="60dp"

android:ems="10"

android:inputType="textPersonName"

android:minHeight="48dp"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toBottomOf="@+id/textView"

tools:ignore="SpeakableTextPresentCheck"

android:autofillHints="" />

<EditText

android:id="@+id/maxPriceInput"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginTop="32dp"

android:ems="10"

android:inputType="textPersonName"

android:minHeight="48dp"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toBottomOf="@+id/nameInput"

tools:ignore="SpeakableTextPresentCheck"

android:autofillHints="" />

</androidx.constraintlayout.widget.ConstraintLayout>

*<?*xml version="1.0" encoding="utf-8"*?>*

<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:app="http://schemas.android.com/apk/res-auto"

xmlns:tools="http://schemas.android.com/tools"

android:id="@+id/coordinatorLayout"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

tools:context=".activities.MainActivity">

<ListView

android:id="@+id/productsListView"

android:layout\_width="280dp"

android:layout\_height="300dp"

android:layout\_marginTop="136dp"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintHorizontal\_bias="0.503"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toTopOf="parent">

</ListView>

<androidx.fragment.app.FragmentContainerView

android:id="@+id/fragmentContainerView"

android:layout\_width="280dp"

android:layout\_height="250dp"

android:layout\_marginTop="28dp"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintHorizontal\_bias="0.496"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toBottomOf="@+id/productsListView" />

<TextView

android:id="@+id/searchQueryTextView"

android:layout\_width="280dp"

android:layout\_height="31dp"

android:layout\_marginTop="28dp"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toTopOf="parent" />

<TextView

android:id="@+id/maxPriceTextView"

android:layout\_width="280dp"

android:layout\_height="31dp"

android:layout\_marginTop="24dp"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintHorizontal\_bias="0.503"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toBottomOf="@+id/searchQueryTextView" />

</androidx.constraintlayout.widget.ConstraintLayout>

*<?*xml version="1.0" encoding="utf-8"*?>*

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:app="http://schemas.android.com/apk/res-auto"

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:orientation="vertical"

android:padding="10dp">

<androidx.constraintlayout.widget.ConstraintLayout

android:layout\_width="match\_parent"

android:layout\_height="75dp">

<Spinner

android:id="@+id/productIdSpinner"

android:layout\_width="121dp"

android:layout\_height="48dp"

android:layout\_marginStart="16dp"

android:layout\_marginTop="16dp"

android:layout\_marginEnd="16dp"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintStart\_toEndOf="@+id/productIdSpinnerLabel"

app:layout\_constraintTop\_toTopOf="parent"

tools:ignore="DuplicateSpeakableTextCheck" />

<TextView

android:id="@+id/productIdSpinnerLabel"

android:layout\_width="212dp"

android:layout\_height="43dp"

android:layout\_margin="16dp"

android:layout\_marginStart="16dp"

android:layout\_marginTop="20dp"

android:gravity="center"

android:labelFor="@id/productIdSpinner"

android:text="@string/please\_select\_productid\_to\_delete"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toTopOf="parent" />

</androidx.constraintlayout.widget.ConstraintLayout>

<androidx.constraintlayout.widget.ConstraintLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content">

<Button

android:id="@+id/deleteProductButton"

android:layout\_width="100dp"

android:layout\_height="48dp"

android:layout\_marginTop="4dp"

android:layout\_marginEnd="1dp"

android:text="@string/delete"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintHorizontal\_bias="1.0"

app:layout\_constraintStart\_toEndOf="@+id/cancelDeletionButton"

app:layout\_constraintTop\_toTopOf="parent" />

<Button

android:id="@+id/cancelDeletionButton"

android:layout\_width="100dp"

android:layout\_height="48dp"

android:layout\_marginStart="39dp"

android:layout\_marginTop="4dp"

android:text="@string/cancel"

app:layout\_constraintEnd\_toStartOf="@+id/deleteProductButton"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toTopOf="parent" />

</androidx.constraintlayout.widget.ConstraintLayout>

</LinearLayout>

*<?*xml version="1.0" encoding="utf-8"*?>*

<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:app="http://schemas.android.com/apk/res-auto"

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="250dp"

tools:context=".fragments.FullProductInfoFragment">

<TextView

android:id="@+id/fullProductInfoTextView"

android:layout\_width="0dp"

android:layout\_height="190dp"

android:gravity="center"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toTopOf="parent" />

<Button

android:id="@+id/clearButton"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginTop="8dp"

android:text="@string/clear"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toBottomOf="@+id/fullProductInfoTextView"

app:layout\_constraintVertical\_bias="1.0" />

</androidx.constraintlayout.widget.ConstraintLayout>

*<?*xml version="1.0" encoding="utf-8"*?>*

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:app="http://schemas.android.com/apk/res-auto"

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:orientation="vertical"

android:padding="10dp">

<androidx.constraintlayout.widget.ConstraintLayout

android:layout\_width="match\_parent"

android:layout\_height="75dp">

<Spinner

android:id="@+id/productIdSpinner"

android:layout\_width="120dp"

android:layout\_height="48dp"

android:layout\_marginStart="16dp"

android:layout\_marginTop="16dp"

android:layout\_marginEnd="16dp"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintStart\_toEndOf="@+id/productIdSpinnerLabel"

app:layout\_constraintTop\_toTopOf="parent"

tools:ignore="DuplicateSpeakableTextCheck" />

<TextView

android:id="@+id/productIdSpinnerLabel"

android:layout\_width="225dp"

android:layout\_height="43dp"

android:layout\_margin="16dp"

android:layout\_marginStart="4dp"

android:layout\_marginTop="16dp"

android:gravity="center"

android:labelFor="@id/productIdSpinner"

android:text="@string/please\_select\_product\_id"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toTopOf="parent" />

</androidx.constraintlayout.widget.ConstraintLayout>

<androidx.constraintlayout.widget.ConstraintLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content">

<Button

android:id="@+id/selectProductButton"

android:layout\_width="100dp"

android:layout\_height="48dp"

android:layout\_marginTop="4dp"

android:layout\_marginEnd="1dp"

android:text="@android:string/ok"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintHorizontal\_bias="1.0"

app:layout\_constraintStart\_toEndOf="@+id/cancelSelectButton"

app:layout\_constraintTop\_toTopOf="parent"

tools:ignore="TouchTargetSizeCheck" />

<Button

android:id="@+id/cancelSelectButton"

android:layout\_width="100dp"

android:layout\_height="48dp"

android:layout\_marginStart="39dp"

android:layout\_marginTop="4dp"

android:text="@string/cancel"

app:layout\_constraintEnd\_toStartOf="@+id/selectProductButton"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toTopOf="parent"

tools:ignore="TouchTargetSizeCheck" />

</androidx.constraintlayout.widget.ConstraintLayout>

</LinearLayout>

*<?*xml version="1.0" encoding="utf-8"*?>*

<TextView xmlns:android="http://schemas.android.com/apk/res/android"

android:layout\_width="match\_parent"

android:layout\_height="48dp"

android:gravity="center|fill\_horizontal|center\_vertical"

android:textAlignment="center" />

*<?*xml version="1.0" encoding="utf-8"*?>*

<menu xmlns:android="http://schemas.android.com/apk/res/android"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent">

<item

android:title="@string/io">

<menu>

<item

android:id="@+id/load\_products"

android:title="@string/load" />

<item

android:id="@+id/save\_products"

android:title="@string/save" />

</menu>

</item>

<item android:title="@string/search">

<menu>

<item

android:id="@+id/search"

android:title="@string/search\_by\_name" />

<item

android:id="@+id/search\_with\_price"

android:title="@string/search\_by\_name\_and\_price" />

</menu>

</item>

<item

android:id="@+id/crud\_menu"

android:title="@string/actions"

>

<menu

android:layout\_width="match\_parent"

android:layout\_height="match\_parent">

<item

android:id="@+id/get\_all"

android:title="@string/get\_all" />

<item

android:id="@+id/create\_product"

android:title="@string/create" />

<item

android:id="@+id/update\_product"

android:title="@string/update" />

<item

android:id="@+id/delete\_product"

android:title="@string/delete" />

</menu>

</item>

<item

android:id="@+id/theme\_menu"

android:title="@string/theme"

>

<menu

android:layout\_width="match\_parent"

android:layout\_height="match\_parent">

<item

android:id="@+id/default\_theme"

android:title="@string/default\_theme" />

<item

android:id="@+id/red\_theme"

android:title="@string/red" />

<item

android:id="@+id/green\_theme"

android:title="@string/green" />

</menu>

</item>

</menu>

<resources>

<string name="app\_name">lab3</string>

<string name="action\_settings">Settings</string>

*<!-- Strings used for fragments for navigation -->*

<string name="first\_fragment\_label">First Fragment</string>

<string name="second\_fragment\_label">Second Fragment</string>

<string name="next">Next</string>

<string name="previous">Previous</string>

<string name="hello\_first\_fragment">Hello first fragment</string>

<string name="hello\_second\_fragment">Hello second fragment. Arg: %1$s</string>

<string name="button">Button</string>

<string name="finish">finish</string>

*<!-- TODO: Remove or change this placeholder text -->*

<string name="hello\_blank\_fragment">Hello blank fragment</string>

<string name="clear">Clear</string>

<string name="search">Search</string>

<string name="app\_label">app\_label</string>

<string name="search\_hint">search\_hint</string>

<string name="get\_all">Get all</string>

<string name="max\_price">Max price</string>

<string name="search\_by\_name\_and\_max\_price">Search by name and max price</string>

<string name="name">Name</string>

<string name="search\_by\_name\_and\_price">Search by name and price</string>

<string name="theme">Theme</string>

<string name="default\_theme">Default</string>

<string name="red">Red</string>

<string name="green">Green</string>

<string name="create">Create</string>

<string name="update">Update</string>

<string name="delete">Delete</string>

<string name="actions">Actions</string>

<string name="save">Save</string>

<string name="please\_select\_productid\_to\_delete">Please select productId to delete</string>

<string name="please\_select\_productid\_to\_update">Please select productId to update</string>

<string name="cancel">Cancel</string>

<string name="count">Count</string>

<string name="expiration\_date">Expiration date</string>

<string name="price">Price</string>

<string name="universal\_product\_code">Universal Product Code</string>

<string name="producer">Producer</string>

<string name="please\_select\_product\_id">Please select productId</string>

<string name="io">IO</string>

<string name="load">Load</string>

<string name="search\_by\_name">Search by name</string>

</resources>

<resources>

*<!-- Base application theme. -->*

<style name="Theme.Lab3" parent="Theme.MaterialComponents.DayNight.DarkActionBar">

*<!-- Primary brand color. -->*

<item name="colorPrimary">@color/purple\_500</item>

<item name="colorPrimaryVariant">@color/purple\_700</item>

<item name="colorOnPrimary">@color/white</item>

*<!-- Secondary brand color. -->*

<item name="colorSecondary">@color/teal\_200</item>

<item name="colorSecondaryVariant">@color/teal\_700</item>

<item name="colorOnSecondary">@color/black</item>

*<!-- Status bar color. -->*

<item name="android:statusBarColor">?attr/colorPrimaryVariant</item>

*<!-- Customize your theme here. -->*

</style>

<style name="Theme.Lab3.red" parent="Theme.MaterialComponents.DayNight.DarkActionBar">

<item name="colorPrimary">@color/red\_200</item>

</style>

<style name="Theme.Lab3.green" parent="Theme.MaterialComponents.DayNight.DarkActionBar">

<item name="colorPrimary">@color/green\_200</item>

</style>

<style name="Theme.Lab3.NoActionBar">

<item name="windowActionBar">true</item>

<item name="windowNoTitle">false</item>

</style>

<style name="Theme.Lab3.AppBarOverlay" parent="ThemeOverlay.AppCompat.Dark.ActionBar" />

<style name="Theme.Lab3.PopupOverlay" parent="ThemeOverlay.AppCompat.Light" />

<color name="red\_200">#F44336</color>

<color name="green\_200">#52F436</color>

</resources>

*<?*xml version="1.0" encoding="utf-8"*?>*

<searchable xmlns:android="http://schemas.android.com/apk/res/android"

android:label="@string/app\_label"

android:hint="@string/search\_hint" >

</searchable>

[

{

"id": 0,

"name": "Guitar",

"universalProductCode": 438,

"producer": "China",

"price": 200,

"expirationDate": "02.12.2023",

"count": 2

},

{

"id": 1,

"name": "Laptop",

"universalProductCode": 251,

"producer": "Germany",

"price": 500,

"expirationDate": "01.01.2030",

"count": 10

},

{

"id": 2,

"name": "Table",

"universalProductCode": 193,

"producer": "Belarus",

"price": 100,

"expirationDate": "02.01.2032",

"count": 1

},

{

"id": 3,

"name": "Iphone",

"universalProductCode": 101,

"producer": "USA",

"price": 1500,

"expirationDate": "26.07.2025",

"count": 100

},

{

"id": 4,

"name": "IPad",

"universalProductCode": 102,

"producer": "USA",

"price": 2000,

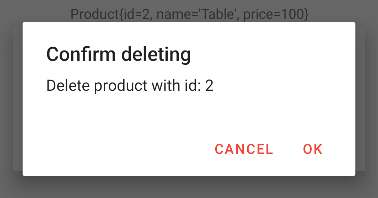
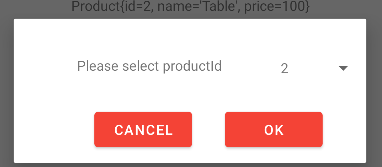
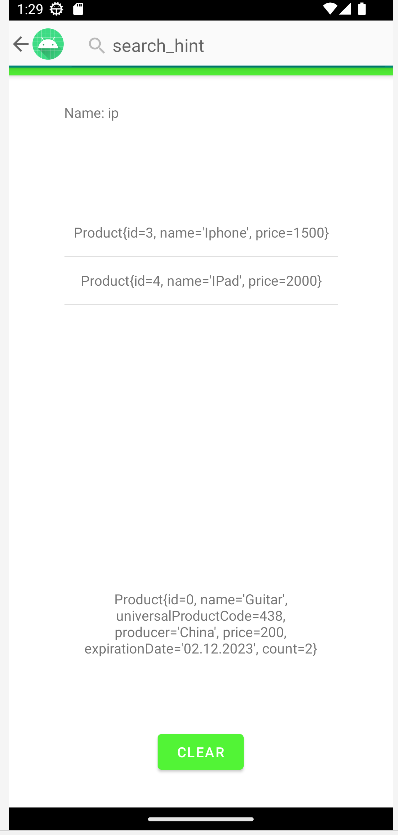
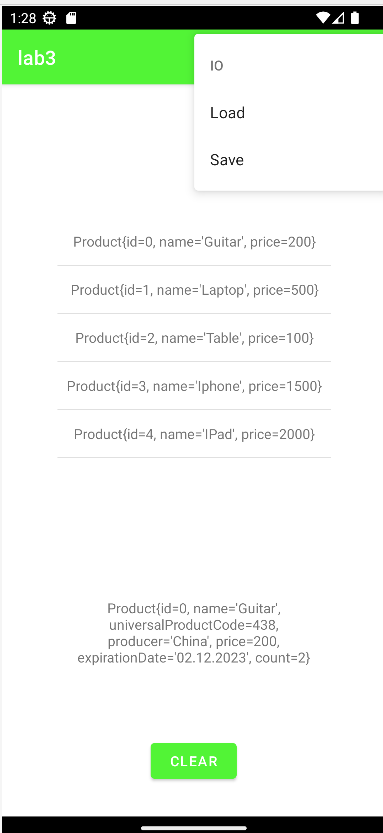
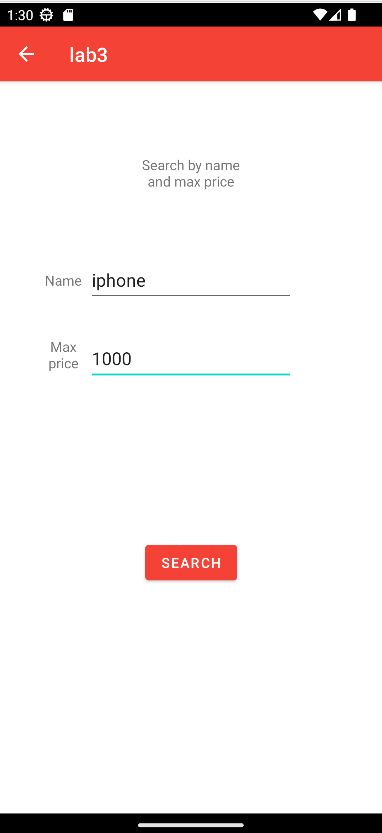
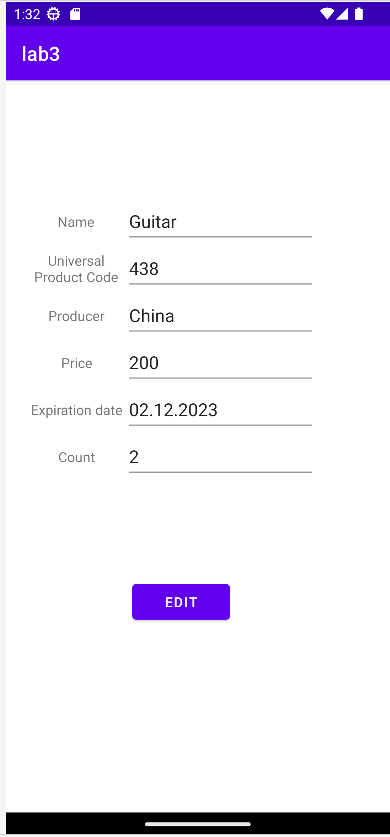
"expirationDate": "26.08.2031",

"count": 100

}

]

**Результаты**



**Вывод:** изучил методы жизненного цикла активности и получил практические навыки по созданию пользовательского интерфейса.