Министерство образования и молодежной политики Свердловской области



ГАПОУ СО «Екатеринбургский колледж транспортного строительства»

Отчёт по учебной практике

УП 01.01

Выполнил: Гордейчик Степан Алексеевич

Группа: ПР-21

Преподаватель: Мирошниченко Г.В.

2023

Содержание

[1. Индивидуальное задание №8 (вариант - 1) 3](#_Toc138717616)

[1.1 Описание задачи 3](#_Toc138717617)

[1.2 Структура проекта 4](#_Toc138717618)

[1.3 Описание разработанных функций 32](#_Toc138717619)

[1.4 Алгоритм решения 34](#_Toc138717620)

[1.5 Используемые библиотеки 34](#_Toc138717621)

[1.6 Тестовые случаи 35](#_Toc138717622)

[1.7 Используемые инструменты 35](#_Toc138717623)

[1.8 Описание пользовательского интерфейса 35](#_Toc138717624)

[1.9 Приложение (print screen) 35](#_Toc138717625)

# Индивидуальное задание №8 (вариант - 1)

## Описание задачи

Цель задачи заключается в создании мобильного приложения на платформе Android, используя язык программирования Kotlin и Android Studio. Приложение должно состоять из четырех экранов в соответствии с предоставленным макетом.

Экран 1 - SigninUpScreen:

На этом экране пользователь может войти или зарегистрироваться. Необходимо выполнить проверку на пустоту полей и сохранить имя пользователя при повторном входе. Для сохранения имени пользователя и проверки сохраненного пароля используется механизм SharedPreference.

Экран 2 - Collection Screen:

На этом экране необходимо реализовать меню, которое предоставит пользователю доступ к различным функциям приложения.

Экран 3 - Подборка:

На этом экране отображаются фильмы с иконкой и названием. Для хранения информации о фильмах и их жанрах используется компонент Room, который представляет собой базу данных на устройстве пользователя. Пользователь должен иметь возможность выбирать, добавлять и удалять фильмы, а также сортировать их по жанру.

Экран 4:

При выборе фильма на экране 3 должна открываться информация о фильме. Экран должен отображать фото, автора, название и жанр фильма.

Требования к задаче:

* Использование языка программирования Kotlin.
* Использование Android Studio в качестве интегрированной среды разработки.
* Использование фрагментов для реализации различных экранов приложения.
* Использование компонента Room для хранения информации о фильмах и их жанрах.
* Использование компонента RecyclerView для отображения списка фильмов на экране 3.
* Использование механизма SharedPreference для сохранения имени пользователя и проверки сохраненного пароля.
* Реализация функций выбора, добавления, удаления и сортировки фильмов на экране 3.
* Правильное отображение информации о фильме на экране 4, включая фото, автора, название и жанр.
* Соответствие дизайну, предоставленному в макете, на всех экранах приложения.
* Обработка ошибок и валидация ввода данных, например, проверка на пустоту полей при входе на экране 1.

## Структура проекта

1. sign\_up\_screen.xml:

*<?xml version*="1.0" *encoding*="utf-8"*?>*<ScrollView *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*=".SignUpScreen"  
 *android:background*="@color/background">  
  
 <androidx.constraintlayout.widget.ConstraintLayout  
 *android:layout\_width*="match\_parent"  
 *android:layout\_height*="wrap\_content">  
  
 <ImageView  
 *android:id*="@+id/imageView1"  
 *android:layout\_width*="229dp"  
 *android:layout\_height*="204dp"  
 *android:layout\_marginTop*="24dp"  
 *android:src*="@drawable/movie"  
 *app:layout\_constraintEnd\_toEndOf*="parent"  
 *app:layout\_constraintStart\_toStartOf*="parent"  
 *app:layout\_constraintTop\_toTopOf*="parent" />  
  
 <ImageView  
 *android:id*="@+id/imageView2"  
 *android:layout\_width*="wrap\_content"  
 *android:layout\_height*="wrap\_content"  
 *android:layout\_marginTop*="108dp"  
 *android:src*="@drawable/frame\_movie"  
 *app:layout\_constraintEnd\_toEndOf*="parent"  
 *app:layout\_constraintStart\_toStartOf*="parent"  
 *app:layout\_constraintTop\_toTopOf*="parent" />  
  
 <EditText  
 *android:id*="@+id/nameEditText"  
 *android:layout\_width*="370dp"  
 *android:layout\_height*="50dp"  
 *android:layout\_marginTop*="55dp"  
 *android:background*="@drawable/border"  
 *android:hint*="@string/name"  
 *android:paddingLeft*="16dp"  
 *android:maxLength*="45"  
 *android:textColor*="@color/white"  
 *android:textColorHint*="@color/light\_grey"  
 *app:layout\_constraintEnd\_toEndOf*="parent"  
 *app:layout\_constraintStart\_toStartOf*="parent"  
 *app:layout\_constraintTop\_toBottomOf*="@+id/imageView2"  
 *android:singleLine*="true"  
 *tools:ignore*="MissingConstraints" />  
  
 <EditText  
 *android:id*="@+id/surnameEditText"  
 *android:layout\_width*="370dp"  
 *android:layout\_height*="50dp"  
 *android:layout\_marginTop*="18dp"  
 *android:background*="@drawable/border"  
 *android:hint*="@string/surname"  
 *android:maxLength*="45"  
 *android:paddingLeft*="16dp"  
 *android:singleLine*="true"  
 *android:textColor*="@color/white"  
 *android:textColorHint*="@color/light\_grey"  
 *app:layout\_constraintEnd\_toEndOf*="parent"  
 *app:layout\_constraintStart\_toStartOf*="parent"  
 *app:layout\_constraintTop\_toBottomOf*="@+id/nameEditText"  
 *tools:ignore*="MissingConstraints" />  
  
 <EditText  
 *android:id*="@+id/emailEditText"  
 *android:layout\_width*="370dp"  
 *android:layout\_height*="50dp"  
 *android:layout\_marginTop*="18dp"  
 *android:background*="@drawable/border"  
 *android:hint*="@string/email"  
 *android:maxLength*="45"  
 *android:paddingLeft*="16dp"  
 *android:singleLine*="true"  
 *android:textColor*="@color/white"  
 *android:textColorHint*="@color/light\_grey"  
 *app:layout\_constraintEnd\_toEndOf*="parent"  
 *app:layout\_constraintStart\_toStartOf*="parent"  
 *app:layout\_constraintTop\_toBottomOf*="@+id/surnameEditText"  
 *tools:ignore*="MissingConstraints" />  
  
 <EditText  
 *android:id*="@+id/passwordEditText"  
 *android:layout\_width*="370dp"  
 *android:layout\_height*="50dp"  
 *android:layout\_marginTop*="18dp"  
 *android:background*="@drawable/border"  
 *android:hint*="@string/password"  
 *android:password*="true"  
 *android:maxLength*="45"  
 *android:paddingLeft*="16dp"  
 *android:singleLine*="true"  
 *android:textColor*="@color/white"  
 *android:textColorHint*="@color/light\_grey"  
 *app:layout\_constraintEnd\_toEndOf*="parent"  
 *app:layout\_constraintStart\_toStartOf*="parent"  
 *app:layout\_constraintTop\_toBottomOf*="@+id/emailEditText"  
 *tools:ignore*="MissingConstraints" />  
  
 <EditText  
 *android:id*="@+id/re\_passwordEditText"  
 *android:layout\_width*="370dp"  
 *android:layout\_height*="50dp"  
 *android:layout\_marginTop*="18dp"  
 *android:background*="@drawable/border"  
 *android:hint*="@string/re\_password"  
 *android:password*="true"  
 *android:maxLength*="45"  
 *android:paddingLeft*="16dp"  
 *android:singleLine*="true"  
 *android:textColor*="@color/white"  
 *android:textColorHint*="@color/light\_grey"  
 *app:layout\_constraintEnd\_toEndOf*="parent"  
 *app:layout\_constraintStart\_toStartOf*="parent"  
 *app:layout\_constraintTop\_toBottomOf*="@+id/passwordEditText"  
 *tools:ignore*="MissingConstraints" />  
  
 <Button  
 *android:id*="@+id/registerButton"  
 *android:layout\_width*="370dp"  
 *android:layout\_height*="50dp"  
 *android:layout\_marginTop*="18dp"  
 *android:background*="@drawable/button\_border\_blue"  
 *android:hint*="@string/register"  
 *android:textColorHint*="@color/white"  
 *android:textSize*="17sp"  
 *app:layout\_constraintEnd\_toEndOf*="parent"  
 *app:layout\_constraintStart\_toStartOf*="parent"  
 *app:layout\_constraintTop\_toBottomOf*="@+id/re\_passwordEditText"  
 *tools:ignore*="MissingConstraints" />  
  
 <Button  
 *android:id*="@+id/iHaveAccountButton"  
 *android:layout\_width*="370dp"  
 *android:layout\_height*="50dp"  
 *android:layout\_marginTop*="18dp"  
 *android:background*="@drawable/border"  
 *android:hint*="@string/i\_have\_account"  
 *android:textColorHint*="@color/blue\_buttons"  
 *android:textSize*="17sp"  
 *app:layout\_constraintEnd\_toEndOf*="parent"  
 *app:layout\_constraintStart\_toStartOf*="parent"  
 *app:layout\_constraintTop\_toBottomOf*="@+id/registerButton"  
 *tools:ignore*="MissingConstraints" />  
  
 <Space  
 *android:id*="@+id/space"  
 *android:layout\_width*="match\_parent"  
 *android:layout\_height*="80dp"  
 *app:layout\_constraintTop\_toBottomOf*="@+id/iHaveAccountButton" />  
  
 </androidx.constraintlayout.widget.ConstraintLayout>  
</ScrollView>

1. SignUpScreen.kt:

*package* com.bignerdranch.android.cinemaapp  
  
*import* android.content.Context  
*import* android.content.Intent  
*import* android.content.SharedPreferences  
*import* android.os.Bundle  
*import* android.widget.Button  
*import* android.widget.EditText  
*import* android.widget.Toast  
*import* androidx.appcompat.app.AppCompatActivity  
  
*class* SignUpScreen : AppCompatActivity() {  
  
 *private lateinit var* sharedPreferences: SharedPreferences  
 *private lateinit var* editor: SharedPreferences.Editor  
 *private val* sharedPrefFile = "com.example.emailPasswordAndName"  
 *private lateinit var* nameEditText: EditText  
  
 *override fun* onCreate(savedInstanceState: Bundle?) {  
 *super*.onCreate(savedInstanceState)  
 setContentView(R.layout.*sign\_up\_screen*)  
  
 *val* iHaveAccountButton: Button = findViewById(R.id.*iHaveAccountButton*)  
 iHaveAccountButton.setOnClickListener **{** *val* intent = Intent(*this*, SignInScreen::*class*.*java*)  
 startActivity(intent)  
 **}** sharedPreferences = getSharedPreferences(sharedPrefFile, Context.*MODE\_PRIVATE*)  
 editor = sharedPreferences.edit()  
  
 nameEditText = findViewById(R.id.*nameEditText*)  
 *val* savedName = sharedPreferences.getString("name", "")  
 nameEditText.setText(savedName)  
  
 *val* registerButton: Button = findViewById(R.id.*registerButton*)  
 registerButton.setOnClickListener **{** *val* surnameEditText: EditText = findViewById(R.id.*surnameEditText*)  
 *val* rePasswordEditText: EditText = findViewById(R.id.*re\_passwordEditText*)  
 *val* emailEditText: EditText = findViewById(R.id.*emailEditText*)  
 *val* passwordEditText: EditText = findViewById(R.id.*passwordEditText*)  
  
 *if* (nameEditText.*text*.toString().isEmpty() || surnameEditText.*text*.toString().isEmpty() ||  
 emailEditText.*text*.toString().isEmpty() || passwordEditText.*text*.toString().isEmpty() ||  
 rePasswordEditText.*text*.toString().isEmpty()) {  
  
 Toast.makeText(*this*, "Заполните все поля для ввода.", Toast.*LENGTH\_SHORT*).show()  
 *return*@setOnClickListener  
 } *else if* (passwordEditText.*text*.toString() != rePasswordEditText.*text*.toString()){  
 Toast.makeText(*this*, "Пароли не совпадают.", Toast.*LENGTH\_SHORT*).show()  
 *return*@setOnClickListener  
 } *else if* (!isEmailValid(emailEditText.*text*.toString())) {  
 Toast.makeText(*this*, "Невверный ввод почты.", Toast.*LENGTH\_SHORT*).show()  
 *return*@setOnClickListener  
 }  
  
 *val* email = emailEditText.*text*.toString()  
 *val* password = passwordEditText.*text*.toString()  
 *val* name = nameEditText.*text*.toString()  
  
 editor.putString("email", email)  
 editor.putString("password", password)  
 editor.putString("name", name)  
 editor.apply()  
  
 Toast.makeText(*this*, "Вы успешно зарегистрировались.", Toast.*LENGTH\_SHORT*).show()  
  
 *val* intent = Intent(*this*, SignInScreen::*class*.*java*)  
 startActivity(intent)  
 **}** }  
  
 *fun* isEmailValid(email: String): Boolean {  
 *val* pattern = Regex("[^@]+@[^@]+\\.[^@]+")  
 *return* pattern.matches(email)  
 }  
}

1. sign\_in\_screen.xml:

*<?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*=".SignInScreen"  
 *android:background*="@color/background">  
  
 <ImageView  
 *android:id*="@+id/imageView1"  
 *android:layout\_width*="229dp"  
 *android:layout\_height*="204dp"  
 *android:layout\_marginTop*="28dp"  
 *android:src*="@drawable/movie"  
 *app:layout\_constraintEnd\_toEndOf*="parent"  
 *app:layout\_constraintStart\_toStartOf*="parent"  
 *app:layout\_constraintTop\_toTopOf*="parent" />  
  
 <ImageView  
 *android:id*="@+id/imageView2"  
 *android:layout\_width*="wrap\_content"  
 *android:layout\_height*="wrap\_content"  
 *android:layout\_marginTop*="108dp"  
 *android:src*="@drawable/frame\_movie"  
 *app:layout\_constraintEnd\_toEndOf*="parent"  
 *app:layout\_constraintStart\_toStartOf*="parent"  
 *app:layout\_constraintTop\_toTopOf*="parent" />  
  
 <EditText  
 *android:id*="@+id/emailEditText"  
 *android:layout\_width*="370dp"  
 *android:layout\_height*="50dp"  
 *android:layout\_marginTop*="70dp"  
 *android:background*="@drawable/border"  
 *android:hint*="@string/email"  
 *android:paddingLeft*="16dp"  
 *android:maxLength*="45"  
 *android:textColor*="@color/white"  
 *android:textColorHint*="@color/light\_grey"  
 *app:layout\_constraintEnd\_toEndOf*="parent"  
 *app:layout\_constraintStart\_toStartOf*="parent"  
 *app:layout\_constraintTop\_toBottomOf*="@+id/imageView2"  
 *android:singleLine*="true"  
 *tools:ignore*="MissingConstraints" />  
  
 <EditText  
 *android:id*="@+id/passwordEditText"  
 *android:layout\_width*="370dp"  
 *android:layout\_height*="50dp"  
 *android:layout\_marginTop*="18dp"  
 *android:background*="@drawable/border"  
 *android:hint*="@string/password"  
 *android:maxLength*="45"  
 *android:paddingLeft*="16dp"  
 *android:singleLine*="true"  
 *android:password*="true"  
 *android:textColor*="@color/white"  
 *android:textColorHint*="@color/light\_grey"  
 *app:layout\_constraintEnd\_toEndOf*="parent"  
 *app:layout\_constraintStart\_toStartOf*="parent"  
 *app:layout\_constraintTop\_toBottomOf*="@+id/emailEditText"  
 *tools:ignore*="MissingConstraints" />  
  
 <Button  
 *android:id*="@+id/signInButton"  
 *android:layout\_width*="370dp"  
 *android:layout\_height*="50dp"  
 *android:layout\_marginTop*="18dp"  
 *android:background*="@drawable/button\_border\_blue"  
 *android:hint*="@string/sign\_in"  
 *android:textColorHint*="@color/white"  
 *android:textSize*="17sp"  
 *app:layout\_constraintEnd\_toEndOf*="parent"  
 *app:layout\_constraintStart\_toStartOf*="parent"  
 *app:layout\_constraintTop\_toBottomOf*="@+id/passwordEditText"  
 *tools:ignore*="MissingConstraints" />  
  
 <Button  
 *android:id*="@+id/registerInButton"  
 *android:layout\_width*="370dp"  
 *android:layout\_height*="50dp"  
 *android:layout\_marginTop*="18dp"  
 *android:background*="@drawable/border"  
 *android:hint*="@string/register"  
 *android:textColorHint*="@color/blue\_buttons"  
 *android:textSize*="17sp"  
 *app:layout\_constraintEnd\_toEndOf*="parent"  
 *app:layout\_constraintStart\_toStartOf*="parent"  
 *app:layout\_constraintTop\_toBottomOf*="@+id/signInButton"  
 *tools:ignore*="MissingConstraints" />  
  
</androidx.constraintlayout.widget.ConstraintLayout>

1. SignInScreen.kt:

*package* com.bignerdranch.android.cinemaapp  
  
*import* android.content.Context  
*import* android.content.Intent  
*import* android.content.SharedPreferences  
*import* android.os.Bundle  
*import* android.widget.Button  
*import* android.widget.EditText  
*import* android.widget.Toast  
*import* androidx.appcompat.app.AppCompatActivity  
  
*class* SignInScreen : AppCompatActivity() {  
  
 *private lateinit var* sharedPreferences: SharedPreferences  
 *private lateinit var* editor: SharedPreferences.Editor  
 *private val* sharedPrefFile = "com.example.emailPasswordAndName"  
 *private lateinit var* emailEditText: EditText  
  
 *override fun* onCreate(savedInstanceState: Bundle?) {  
 *super*.onCreate(savedInstanceState)  
 setContentView(R.layout.*sign\_in\_screen*)  
  
 *val* registerButton: Button = findViewById(R.id.*registerInButton*)  
 registerButton.setOnClickListener **{** *val* intent = Intent(*this*, SignUpScreen::*class*.*java*)  
 startActivity(intent)  
 **}** sharedPreferences = getSharedPreferences(sharedPrefFile, Context.*MODE\_PRIVATE*)  
 editor = sharedPreferences.edit()  
  
 emailEditText = findViewById(R.id.*emailEditText*)  
 *val* savedEmail = sharedPreferences.getString("email", "")  
 emailEditText.setText(savedEmail)  
  
 *val* signInButton: Button = findViewById(R.id.*signInButton*)  
 signInButton.setOnClickListener **{** *val* emailEditText: EditText = findViewById(R.id.*emailEditText*)  
 *val* passwordEditText: EditText = findViewById(R.id.*passwordEditText*)  
  
 *val* email = emailEditText.*text*.toString()  
 *val* password = passwordEditText.*text*.toString()  
  
 *if* (email.isEmpty() || password.isEmpty()) {  
 Toast.makeText(*this*, "Заполните все поля", Toast.*LENGTH\_SHORT*).show()  
 } *else* {  
 *val* savedPassword = sharedPreferences.getString("password", "")  
  
 *if* (email == savedEmail && password == savedPassword) {  
 *val* intent = Intent(*this*, MainActivity::*class*.*java*)  
 startActivity(intent)  
 } *else* {  
 Toast.makeText(*this*, "Неверный email или пароль", Toast.*LENGTH\_SHORT*).show()  
 }  
 }  
 **}** }  
}

1. activity\_navigation\_set.xml:

*<?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*=".NavigationSet"  
 *android:background*="@color/background">  
  
 <TextView  
 *android:id*="@+id/setTextView"  
 *android:layout\_width*="wrap\_content"  
 *android:layout\_height*="wrap\_content"  
 *android:layout\_marginTop*="41dp"  
 *android:text*="@string/title\_set"  
 *android:textColor*="@color/white"  
 *android:textSize*="23sp"  
 *app:layout\_constraintEnd\_toEndOf*="parent"  
 *app:layout\_constraintHorizontal\_bias*="0.25"  
 *app:layout\_constraintStart\_toStartOf*="parent"  
 *app:layout\_constraintTop\_toTopOf*="parent"  
 *tools:ignore*="MissingConstraints" />  
  
 <Button  
 *android:id*="@+id/createSetButton"  
 *android:layout\_width*="18dp"  
 *android:layout\_height*="18dp"  
 *android:layout\_marginTop*="48dp"  
 *android:layout\_marginEnd*="30dp"  
 *android:background*="@drawable/plus"  
 *app:layout\_constraintEnd\_toEndOf*="parent"  
 *app:layout\_constraintTop\_toTopOf*="parent"  
 *tools:ignore*="MissingConstraints" />  
  
 <androidx.recyclerview.widget.RecyclerView  
 *android:id*="@+id/setRecyclerView"  
 *android:layout\_width*="match\_parent"  
 *android:layout\_height*="525dp"  
 *android:layout\_below*="@id/createSetButton"  
 *android:background*="@color/white"  
 *android:backgroundTint*="@color/background"  
 *app:layout\_constraintBottom\_toBottomOf*="parent"  
 *app:layout\_constraintEnd\_toEndOf*="parent"  
 *app:layout\_constraintStart\_toStartOf*="parent"  
 *app:layout\_constraintTop\_toBottomOf*="@+id/sortSpinner"  
 *app:layout\_constraintVertical\_bias*="1.0"  
 *tools:ignore*="MissingConstraints" />  
  
 <Spinner  
 *android:id*="@+id/sortSpinner"  
 *android:layout\_width*="370dp"  
 *android:layout\_height*="30dp"  
 *android:layout\_marginTop*="16dp"  
 *android:background*="@drawable/button\_border\_blue"  
 *app:layout\_constraintEnd\_toEndOf*="parent"  
 *app:layout\_constraintStart\_toStartOf*="parent"  
 *app:layout\_constraintTop\_toBottomOf*="@+id/setTextView" />  
  
</androidx.constraintlayout.widget.ConstraintLayout>

1. NavigationSetFragment.kt:

*package* com.bignerdranch.android.cinemaapp  
  
*import* android.app.Activity  
*import* android.app.AlertDialog  
*import* android.content.Intent  
*import* android.net.Uri  
*import* android.os.Bundle  
*import* android.provider.MediaStore  
*import* androidx.fragment.app.Fragment  
*import* android.view.LayoutInflater  
*import* android.view.View  
*import* android.view.ViewGroup  
*import* android.widget.\*  
*import* androidx.lifecycle.*lifecycleScope  
import* androidx.recyclerview.widget.LinearLayoutManager  
*import* androidx.recyclerview.widget.RecyclerView  
*import* androidx.room.Room  
*import* kotlinx.coroutines.Dispatchers  
*import* kotlinx.coroutines.launch  
*import* kotlinx.coroutines.withContext  
  
*class* NavigationSetFragment : Fragment() {  
 *private lateinit var* movieDao: MovieDao  
 *private lateinit var* adapter: MovieAdapter  
 *private lateinit var* selectedImageUri: Uri  
 *private lateinit var* selectedImageView: ImageView  
  
 *override fun* onCreateView(inflater: LayoutInflater, container: ViewGroup?, savedInstanceState: Bundle?): View? {  
 *val* view = inflater.inflate(R.layout.*activity\_navigation\_set*, container, *false*)  
  
 *val* spinner = view.findViewById<Spinner>(R.id.*sortSpinner*)  
 *val* spinnerAdapter = ArrayAdapter.createFromResource(  
 requireContext(),  
 R.array.*spinner\_items\_genres*,  
 android.R.layout.*simple\_spinner\_item* )  
 spinnerAdapter.setDropDownViewResource(android.R.layout.*simple\_spinner\_dropdown\_item*)  
 spinner.*adapter* = spinnerAdapter  
  
 *// сортировка по жанру* spinner.*onItemSelectedListener* = *object* : AdapterView.OnItemSelectedListener {  
 *override fun* onItemSelected(parent: AdapterView<\*>, view: View?, position: Int, id: Long) {  
 *val* genre = parent.getItemAtPosition(position) *as* String  
 loadMoviesByGenre(genre)  
 }  
 *override fun* onNothingSelected(parent: AdapterView<\*>) {}  
 }  
  
 *val* recyclerView = view.findViewById<RecyclerView>(R.id.*setRecyclerView*).apply **{** *layoutManager* = LinearLayoutManager(requireContext())  
 **}** *val* database = Room.databaseBuilder(  
 requireContext(),  
 AppDatabase::*class*.*java*,  
 "movies-dbase"  
 ).build()  
 movieDao = database.movieDao()  
  
 adapter = MovieAdapter(*emptyList*(), *viewLifecycleOwner*.*lifecycleScope*, movieDao, **{** movie **->** onMovieClicked(movie) **}**, **{** loadMovies() **}**)  
 recyclerView.*adapter* = adapter  
  
 *// загрузка списка фильмов из базы данных и установка его в адаптер* loadMovies()  
  
 *val* createSetButton = view.findViewById<Button>(R.id.*createSetButton*)  
 createSetButton.setOnClickListener **{** *// обработатка нажатия кнопки "Добавить фильм"* showAddMovieDialog()  
 **}** *return* view  
 }  
  
 *// загрузка фильмов по жанру  
 private fun* loadMoviesByGenre(genre: String) {  
 *lifecycleScope*.launch(Dispatchers.IO) **{** *val* movies = movieDao.getMoviesByGenre(genre)  
 withContext(Dispatchers.Main) **{** adapter.setData(movies)  
 **}  
 }** }  
  
 *// получение списка всех фильмов из базы данных и обновление адаптера  
 private fun* loadMovies() {  
 *lifecycleScope*.launch(Dispatchers.IO) **{** *val* movies = movieDao.getAllMovies()  
 withContext(Dispatchers.Main) **{** adapter.setData(movies)  
 **}  
 }** }  
  
 *// открытие галереи, чтобы выбрать изображение  
 private fun* openGallery() {  
 *val* intent = Intent(Intent.*ACTION\_PICK*, MediaStore.Images.Media.*EXTERNAL\_CONTENT\_URI*)  
 startActivityForResult(intent, REQUEST\_SELECT\_IMAGE)  
 }  
  
 *// обработка результата выбора галереи  
 override fun* onActivityResult(requestCode: Int, resultCode: Int, data: Intent?) {  
 *super*.onActivityResult(requestCode, resultCode, data)  
 *if* (requestCode == REQUEST\_SELECT\_IMAGE && resultCode == Activity.*RESULT\_OK* && data != *null*) {  
 *val* selectedImageUri = data.*data* selectedImageView.*visibility* = View.*VISIBLE* selectedImageView.setImageURI(selectedImageUri)  
 *if* (selectedImageUri != *null*) {  
 *this*.selectedImageUri = selectedImageUri  
 }  
 }  
 }  
  
 *// отображение диалогового окна для добавления фильма  
 private fun* showAddMovieDialog() {  
 *val* dialogView = *layoutInflater*.inflate(R.layout.*dialog\_add\_movie*, *null*)  
  
 *val* titleEditText = dialogView.findViewById<EditText>(R.id.*titleEditText*)  
 *val* authorEditText = dialogView.findViewById<EditText>(R.id.*authorEditText*)  
 *val* genreSpinner = dialogView.findViewById<Spinner>(R.id.*genreSpinner*)  
 *val* addImageButton = dialogView.findViewById<Button>(R.id.*addImageButton*)  
 selectedImageView = dialogView.findViewById(R.id.*selectedImageView*)  
 *val* addButton = dialogView.findViewById<Button>(R.id.*addButton*)  
  
 *val* alertDialog = AlertDialog.Builder(requireContext())  
 .setView(dialogView)  
 .setTitle("Добавить фильм")  
 .setCancelable(*false*)  
 .create()  
  
 addImageButton.setOnClickListener **{** openGallery()  
 **}** addButton.setOnClickListener **{** *val* title = titleEditText.*text*.toString()  
 *val* author = authorEditText.*text*.toString()  
 *val* genre = genreSpinner.*selectedItem*.toString()  
 *val* icon = selectedImageUri.toString()  
  
 *if* (title.isNotEmpty() && genre.isNotEmpty() && icon.isNotEmpty() && author.isNotEmpty()) {  
 *val* movie = Movie(icon = icon, title = title, genre = genre, author = author)  
 *lifecycleScope*.launch(Dispatchers.IO) **{** movieDao.addMovie(movie)  
 withContext(Dispatchers.Main) **{** loadMovies()  
 alertDialog.dismiss()  
 **}  
 }** } *else* {  
 Toast.makeText(requireContext(), "Заполните все поля", Toast.*LENGTH\_SHORT*).show()  
 }  
 **}** alertDialog.show()  
 }  
  
 *// обработка клика по фильму в RecyclerView  
 private fun* onMovieClicked(movie: Movie) {  
 *val* intent = Intent(requireContext(), MovieDetailsActivity::*class*.*java*)  
 intent.putExtra("movie", movie)  
 startActivityForResult(intent, REQUEST\_MOVIE\_DETAILS)  
 }  
  
 *companion object* {  
 *private const val* REQUEST\_SELECT\_IMAGE = 100  
 *private const val* REQUEST\_MOVIE\_DETAILS = 101  
 }  
}

1. create\_collection\_screen.xml:

*<?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"  
 *android:background*="@color/background">  
  
 <ImageView  
 *android:id*="@+id/backImage"  
 *android:layout\_width*="18dp"  
 *android:layout\_height*="18dp"  
 *android:layout\_marginStart*="30dp"  
 *android:layout\_marginTop*="48dp"  
 *android:src*="@drawable/arrow\_back"  
 *app:layout\_constraintStart\_toStartOf*="parent"  
 *app:layout\_constraintTop\_toTopOf*="parent"  
 *tools:ignore*="MissingConstraints" />  
  
 <TextView  
 *android:id*="@+id/titleTextView"  
 *android:layout\_width*="wrap\_content"  
 *android:layout\_height*="wrap\_content"  
 *android:layout\_marginTop*="41dp"  
 *android:text*="@string/create\_collection"  
 *android:textColor*="@color/white"  
 *android:textSize*="23sp"  
 *app:layout\_constraintEnd\_toEndOf*="parent"  
 *app:layout\_constraintHorizontal\_bias*="0.446"  
 *app:layout\_constraintStart\_toStartOf*="parent"  
 *app:layout\_constraintTop\_toTopOf*="parent"  
 *tools:ignore*="MissingConstraints" />  
  
 <ImageView  
 *android:id*="@+id/selectedIconImageView"  
 *android:layout\_width*="wrap\_content"  
 *android:layout\_height*="wrap\_content"  
 *android:layout\_marginStart*="18dp"  
 *android:layout\_marginTop*="18dp"  
 *android:src*="@drawable/ico1"  
 *app:layout\_constraintEnd\_toStartOf*="@+id/selectIconButton"  
 *app:layout\_constraintHorizontal\_bias*="0.064"  
 *app:layout\_constraintStart\_toStartOf*="parent"  
 *app:layout\_constraintTop\_toBottomOf*="@+id/nameEditText" />  
  
 <EditText  
 *android:id*="@+id/nameEditText"  
 *android:layout\_width*="370dp"  
 *android:layout\_height*="48dp"  
 *android:layout\_marginTop*="44dp"  
 *android:background*="@drawable/border"  
 *android:hint*="@string/title"  
 *android:maxLength*="20"  
 *android:paddingLeft*="16dp"  
 *android:singleLine*="true"  
 *android:textColor*="@color/light\_grey"  
 *android:textColorHint*="@color/light\_grey"  
 *app:layout\_constraintEnd\_toEndOf*="parent"  
 *app:layout\_constraintHorizontal\_bias*="0.487"  
 *app:layout\_constraintStart\_toStartOf*="parent"  
 *app:layout\_constraintTop\_toBottomOf*="@+id/titleTextView"  
 *tools:ignore*="MissingConstraints" />  
  
 <Button  
 *android:id*="@+id/selectIconButton"  
 *android:layout\_width*="274dp"  
 *android:layout\_height*="49dp"  
 *android:layout\_marginTop*="28dp"  
 *android:layout\_marginEnd*="20dp"  
 *android:background*="@drawable/border"  
 *android:text*="@string/select\_icon"  
 *android:textAllCaps*="false"  
 *android:textColor*="@color/blue\_buttons"  
 *android:textSize*="17sp"  
 *app:layout\_constraintEnd\_toEndOf*="parent"  
 *app:layout\_constraintTop\_toBottomOf*="@+id/nameEditText"  
 *tools:ignore*="MissingConstraints" />  
  
 <Button  
 *android:id*="@+id/createButton"  
 *android:layout\_width*="370dp"  
 *android:layout\_height*="wrap\_content"  
 *android:layout\_marginTop*="65dp"  
 *android:background*="@drawable/button\_border\_blue"  
 *android:text*="@string/save"  
 *android:textAllCaps*="false"  
 *android:textColor*="@color/white"  
 *android:textSize*="17sp"  
 *app:layout\_constraintEnd\_toEndOf*="parent"  
 *app:layout\_constraintStart\_toStartOf*="parent"  
 *app:layout\_constraintTop\_toBottomOf*="@+id/selectIconButton"  
 *tools:ignore*="MissingConstraints" />  
  
</androidx.constraintlayout.widget.ConstraintLayout>

1. CreateCollectionScreen.kt:

*package* com.bignerdranch.android.cinemaapp  
  
*import* android.content.Intent  
*import* androidx.appcompat.app.AppCompatActivity  
*import* android.os.Bundle  
*import* android.widget.Button  
*import* android.widget.EditText  
*import* android.widget.ImageView  
  
*// окно для создания новой коллекции  
class* CreateCollectionScreen : AppCompatActivity() {  
  
 *private val* collections = *mutableListOf*<Collection>()  
  
 *override fun* onCreate(savedInstanceState: Bundle?) {  
 *super*.onCreate(savedInstanceState)  
 setContentView(R.layout.*create\_collection\_screen*)  
 *supportActionBar*?.setDisplayHomeAsUpEnabled(*true*)  
  
 *val* backButton = findViewById<ImageView>(R.id.*backImage*)  
 backButton.setOnClickListener **{** setResult(*RESULT\_CANCELED*)  
 finish()  
 **}** *val* selectIconButton = findViewById<Button>(R.id.*selectIconButton*)  
 selectIconButton.setOnClickListener **{** *val* intent = Intent(*this*, IconSelectionScreen::*class*.*java*)  
 startActivityForResult(intent, REQUEST\_SELECT\_ICON)  
 **}** *val* createButton = findViewById<Button>(R.id.*createButton*)  
 createButton.setOnClickListener **{** *val* collectionName = findViewById<EditText>(R.id.*nameEditText*).*text*.toString()  
 *val* selectedIcon = findViewById<ImageView>(R.id.*selectedIconImageView*).*tag as?* Int  
 *if* (collectionName.isNotEmpty()) {  
 *if* (selectedIcon != *null*) {  
 *val* resultIntent = Intent()  
 resultIntent.putExtra(EXTRA\_COLLECTION\_NAME, collectionName)  
 resultIntent.putExtra(EXTRA\_SELECTED\_ICON, selectedIcon)  
 setResult(*RESULT\_OK*, resultIntent)  
 }  
 *else* {  
 *val* resultIntent = Intent()  
 resultIntent.putExtra(EXTRA\_COLLECTION\_NAME, collectionName)  
 resultIntent.putExtra(EXTRA\_SELECTED\_ICON, R.drawable.*ico1*)  
 setResult(*RESULT\_OK*, resultIntent)  
 }  
 } *else* {  
 setResult(*RESULT\_CANCELED*)  
 }  
 finish()  
 **}** }  
  
 *override fun* onActivityResult(requestCode: Int, resultCode: Int, data: Intent?) {  
 *super*.onActivityResult(requestCode, resultCode, data)  
 *if* (requestCode == REQUEST\_SELECT\_ICON && resultCode == *RESULT\_OK*) {  
 *val* selectedIcon = data?.getIntExtra(IconSelectionScreen.EXTRA\_SELECTED\_ICON, R.drawable.*ico1*)  
 *if* (selectedIcon != *null*) {  
 *val* selectedIconImageView = findViewById<ImageView>(R.id.*selectedIconImageView*)  
 selectedIconImageView.setImageResource(selectedIcon)  
 selectedIconImageView.*tag* = selectedIcon  
 }  
 }  
 }  
  
 *companion object* {  
 *const val* EXTRA\_COLLECTION\_NAME = "collection\_name"  
 *const val* EXTRA\_SELECTED\_ICON = "selected\_icon"  
 *private const val* REQUEST\_SELECT\_ICON = 1  
 }  
}

1. icon\_selection\_screen.xml:

*<?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*=".IconSelectionScreen"  
 *android:background*="@color/background">  
  
 <ImageView  
 *android:id*="@+id/backImage"  
 *android:layout\_width*="18dp"  
 *android:layout\_height*="18dp"  
 *android:layout\_marginStart*="30dp"  
 *android:layout\_marginTop*="48dp"  
 *android:src*="@drawable/arrow\_back"  
 *app:layout\_constraintStart\_toStartOf*="parent"  
 *app:layout\_constraintTop\_toTopOf*="parent"  
 *tools:ignore*="MissingConstraints" />  
  
 <TextView  
 *android:id*="@+id/textView"  
 *android:layout\_width*="wrap\_content"  
 *android:layout\_height*="wrap\_content"  
 *android:layout\_marginTop*="41dp"  
 *android:text*="@string/select\_icon"  
 *android:textColor*="@color/white"  
 *android:textSize*="23sp"  
 *app:layout\_constraintEnd\_toEndOf*="parent"  
 *app:layout\_constraintHorizontal\_bias*="0.367"  
 *app:layout\_constraintStart\_toStartOf*="parent"  
 *app:layout\_constraintTop\_toTopOf*="parent"  
 *tools:ignore*="MissingConstraints" />  
  
 <ScrollView  
 *android:layout\_width*="match\_parent"  
 *android:layout\_height*="wrap\_content"  
 *android:layout\_marginTop*="20dp"  
 *android:orientation*="vertical"  
 *app:layout\_constraintEnd\_toEndOf*="parent"  
 *app:layout\_constraintHorizontal\_bias*="1.0"  
 *app:layout\_constraintStart\_toStartOf*="parent"  
 *app:layout\_constraintTop\_toBottomOf*="@+id/textView">  
  
 <androidx.constraintlayout.widget.ConstraintLayout  
 *android:layout\_width*="match\_parent"  
 *android:layout\_height*="wrap\_content"  
 *android:orientation*="vertical">  
  
 *<!-- 1 строка -->* <ImageView  
 *android:id*="@+id/ico1"  
 *android:layout\_width*="73dp"  
 *android:layout\_height*="73dp"  
 *android:layout\_marginStart*="17dp"  
 *android:src*="@drawable/ico1"  
 *app:layout\_constraintStart\_toStartOf*="parent"  
 *app:layout\_constraintTop\_toTopOf*="parent" />  
 <ImageView  
 *android:id*="@+id/ico2"  
 *android:layout\_width*="73dp"  
 *android:layout\_height*="73dp"  
 *android:layout\_marginStart*="28dp"  
 *android:src*="@drawable/ico2"  
 *app:layout\_constraintStart\_toEndOf*="@+id/ico1"  
 *app:layout\_constraintTop\_toTopOf*="parent" />  
 <ImageView  
 *android:id*="@+id/ico3"  
 *android:layout\_width*="73dp"  
 *android:layout\_height*="73dp"  
 *android:layout\_marginStart*="28dp"  
 *android:src*="@drawable/ico3"  
 *app:layout\_constraintStart\_toEndOf*="@+id/ico2"  
 *app:layout\_constraintTop\_toTopOf*="parent" />  
 <ImageView  
 *android:id*="@+id/ico4"  
 *android:layout\_width*="73dp"  
 *android:layout\_height*="73dp"  
 *android:layout\_marginStart*="16dp"  
 *android:src*="@drawable/ico4"  
 *app:layout\_constraintEnd\_toEndOf*="parent"  
 *app:layout\_constraintHorizontal\_bias*="0.428"  
 *app:layout\_constraintStart\_toEndOf*="@+id/ico3"  
 *app:layout\_constraintTop\_toTopOf*="parent" />  
  
 *<!-- 2 строка -->* <ImageView  
 *android:id*="@+id/ico5"  
 *android:layout\_width*="73dp"  
 *android:layout\_height*="73dp"  
 *android:layout\_marginStart*="17dp"  
 *android:layout\_marginTop*="16dp"  
 *android:src*="@drawable/ico5"  
 *app:layout\_constraintStart\_toStartOf*="parent"  
 *app:layout\_constraintTop\_toBottomOf*="@+id/ico1" />  
 <ImageView  
 *android:id*="@+id/ico6"  
 *android:layout\_width*="73dp"  
 *android:layout\_height*="73dp"  
 *android:layout\_marginStart*="28dp"  
 *android:layout\_marginTop*="16dp"  
 *android:src*="@drawable/ico6"  
 *app:layout\_constraintStart\_toEndOf*="@+id/ico5"  
 *app:layout\_constraintTop\_toBottomOf*="@+id/ico2" />  
 <ImageView  
 *android:id*="@+id/ico7"  
 *android:layout\_width*="73dp"  
 *android:layout\_height*="73dp"  
 *android:layout\_marginStart*="28dp"  
 *android:layout\_marginTop*="16dp"  
 *android:src*="@drawable/ico7"  
 *app:layout\_constraintStart\_toEndOf*="@+id/ico6"  
 *app:layout\_constraintTop\_toBottomOf*="@+id/ico3" />  
 <ImageView  
 *android:id*="@+id/ico8"  
 *android:layout\_width*="73dp"  
 *android:layout\_height*="73dp"  
 *android:layout\_marginTop*="16dp"  
 *android:src*="@drawable/ico8"  
 *app:layout\_constraintEnd\_toEndOf*="parent"  
 *app:layout\_constraintHorizontal\_bias*="0.644"  
 *app:layout\_constraintStart\_toEndOf*="@+id/ico7"  
 *app:layout\_constraintTop\_toBottomOf*="@+id/ico4" />  
  
 *<!-- 3 строка -->* <ImageView  
 *android:id*="@+id/ico9"  
 *android:layout\_width*="73dp"  
 *android:layout\_height*="73dp"  
 *android:layout\_marginStart*="17dp"  
 *android:layout\_marginTop*="16dp"  
 *android:src*="@drawable/ico9"  
 *app:layout\_constraintStart\_toStartOf*="parent"  
 *app:layout\_constraintTop\_toBottomOf*="@+id/ico5" />  
 <ImageView  
 *android:id*="@+id/ico10"  
 *android:layout\_width*="73dp"  
 *android:layout\_height*="73dp"  
 *android:layout\_marginStart*="28dp"  
 *android:layout\_marginTop*="16dp"  
 *android:src*="@drawable/ico10"  
 *app:layout\_constraintStart\_toEndOf*="@+id/ico9"  
 *app:layout\_constraintTop\_toBottomOf*="@+id/ico6" />  
 <ImageView  
 *android:id*="@+id/ico11"  
 *android:layout\_width*="73dp"  
 *android:layout\_height*="73dp"  
 *android:layout\_marginStart*="28dp"  
 *android:layout\_marginTop*="16dp"  
 *android:src*="@drawable/ico11"  
 *app:layout\_constraintStart\_toEndOf*="@+id/ico10"  
 *app:layout\_constraintTop\_toBottomOf*="@+id/ico7" />  
 <ImageView  
 *android:id*="@+id/ico12"  
 *android:layout\_width*="73dp"  
 *android:layout\_height*="73dp"  
 *android:layout\_marginTop*="16dp"  
 *android:src*="@drawable/ico12"  
 *app:layout\_constraintEnd\_toEndOf*="parent"  
 *app:layout\_constraintHorizontal\_bias*="0.644"  
 *app:layout\_constraintStart\_toEndOf*="@+id/ico11"  
 *app:layout\_constraintTop\_toBottomOf*="@+id/ico8" />  
  
 *<!-- 4 строка -->* <ImageView  
 *android:id*="@+id/ico13"  
 *android:layout\_width*="73dp"  
 *android:layout\_height*="73dp"  
 *android:layout\_marginStart*="17dp"  
 *android:layout\_marginTop*="16dp"  
 *android:src*="@drawable/ico13"  
 *app:layout\_constraintStart\_toStartOf*="parent"  
 *app:layout\_constraintTop\_toBottomOf*="@+id/ico9" />  
 <ImageView  
 *android:id*="@+id/ico14"  
 *android:layout\_width*="73dp"  
 *android:layout\_height*="73dp"  
 *android:layout\_marginStart*="28dp"  
 *android:layout\_marginTop*="16dp"  
 *android:src*="@drawable/ico14"  
 *app:layout\_constraintStart\_toEndOf*="@+id/ico13"  
 *app:layout\_constraintTop\_toBottomOf*="@+id/ico10" />  
 <ImageView  
 *android:id*="@+id/ico15"  
 *android:layout\_width*="73dp"  
 *android:layout\_height*="73dp"  
 *android:layout\_marginStart*="28dp"  
 *android:layout\_marginTop*="16dp"  
 *android:src*="@drawable/ico15"  
 *app:layout\_constraintStart\_toEndOf*="@+id/ico14"  
 *app:layout\_constraintTop\_toBottomOf*="@+id/ico11" />  
 <ImageView  
 *android:id*="@+id/ico16"  
 *android:layout\_width*="73dp"  
 *android:layout\_height*="73dp"  
 *android:layout\_marginTop*="16dp"  
 *android:src*="@drawable/ico16"  
 *app:layout\_constraintEnd\_toEndOf*="parent"  
 *app:layout\_constraintHorizontal\_bias*="0.644"  
 *app:layout\_constraintStart\_toEndOf*="@+id/ico15"  
 *app:layout\_constraintTop\_toBottomOf*="@+id/ico12" />  
  
 *<!-- 5 строка -->* <ImageView  
 *android:id*="@+id/ico17"  
 *android:layout\_width*="73dp"  
 *android:layout\_height*="73dp"  
 *android:layout\_marginStart*="17dp"  
 *android:layout\_marginTop*="16dp"  
 *android:src*="@drawable/ico17"  
 *app:layout\_constraintStart\_toStartOf*="parent"  
 *app:layout\_constraintTop\_toBottomOf*="@+id/ico13" />  
 <ImageView  
 *android:id*="@+id/ico18"  
 *android:layout\_width*="73dp"  
 *android:layout\_height*="73dp"  
 *android:layout\_marginStart*="28dp"  
 *android:layout\_marginTop*="16dp"  
 *android:src*="@drawable/ico18"  
 *app:layout\_constraintStart\_toEndOf*="@+id/ico17"  
 *app:layout\_constraintTop\_toBottomOf*="@+id/ico14" />  
 <ImageView  
 *android:id*="@+id/ico19"  
 *android:layout\_width*="73dp"  
 *android:layout\_height*="73dp"  
 *android:layout\_marginStart*="28dp"  
 *android:layout\_marginTop*="16dp"  
 *android:src*="@drawable/ico19"  
 *app:layout\_constraintStart\_toEndOf*="@+id/ico18"  
 *app:layout\_constraintTop\_toBottomOf*="@+id/ico15" />  
 <ImageView  
 *android:id*="@+id/ico20"  
 *android:layout\_width*="73dp"  
 *android:layout\_height*="73dp"  
 *android:layout\_marginTop*="16dp"  
 *android:src*="@drawable/ico20"  
 *app:layout\_constraintEnd\_toEndOf*="parent"  
 *app:layout\_constraintHorizontal\_bias*="0.644"  
 *app:layout\_constraintStart\_toEndOf*="@+id/ico19"  
 *app:layout\_constraintTop\_toBottomOf*="@+id/ico16" />  
  
 *<!-- 6 строка -->* <ImageView  
 *android:id*="@+id/ico21"  
 *android:layout\_width*="73dp"  
 *android:layout\_height*="73dp"  
 *android:layout\_marginStart*="17dp"  
 *android:layout\_marginTop*="16dp"  
 *android:src*="@drawable/ico21"  
 *app:layout\_constraintStart\_toStartOf*="parent"  
 *app:layout\_constraintTop\_toBottomOf*="@+id/ico17" />  
 <ImageView  
 *android:id*="@+id/ico22"  
 *android:layout\_width*="73dp"  
 *android:layout\_height*="73dp"  
 *android:layout\_marginStart*="28dp"  
 *android:layout\_marginTop*="16dp"  
 *android:src*="@drawable/ico22"  
 *app:layout\_constraintStart\_toEndOf*="@+id/ico21"  
 *app:layout\_constraintTop\_toBottomOf*="@+id/ico18" />  
 <ImageView  
 *android:id*="@+id/ico23"  
 *android:layout\_width*="73dp"  
 *android:layout\_height*="73dp"  
 *android:layout\_marginStart*="28dp"  
 *android:layout\_marginTop*="16dp"  
 *android:src*="@drawable/ico23"  
 *app:layout\_constraintStart\_toEndOf*="@+id/ico22"  
 *app:layout\_constraintTop\_toBottomOf*="@+id/ico19" />  
 <ImageView  
 *android:id*="@+id/ico24"  
 *android:layout\_width*="73dp"  
 *android:layout\_height*="73dp"  
 *android:layout\_marginTop*="16dp"  
 *android:src*="@drawable/ico24"  
 *app:layout\_constraintEnd\_toEndOf*="parent"  
 *app:layout\_constraintHorizontal\_bias*="0.644"  
 *app:layout\_constraintStart\_toEndOf*="@+id/ico23"  
 *app:layout\_constraintTop\_toBottomOf*="@+id/ico20" />  
  
 *<!-- 7 строка -->* <ImageView  
 *android:id*="@+id/ico25"  
 *android:layout\_width*="73dp"  
 *android:layout\_height*="73dp"  
 *android:layout\_marginStart*="17dp"  
 *android:layout\_marginTop*="16dp"  
 *android:src*="@drawable/ico25"  
 *app:layout\_constraintStart\_toStartOf*="parent"  
 *app:layout\_constraintTop\_toBottomOf*="@+id/ico21" />  
 <ImageView  
 *android:id*="@+id/ico26"  
 *android:layout\_width*="73dp"  
 *android:layout\_height*="73dp"  
 *android:layout\_marginStart*="28dp"  
 *android:layout\_marginTop*="16dp"  
 *android:src*="@drawable/ico26"  
 *app:layout\_constraintStart\_toEndOf*="@+id/ico25"  
 *app:layout\_constraintTop\_toBottomOf*="@+id/ico22" />  
 <ImageView  
 *android:id*="@+id/ico27"  
 *android:layout\_width*="73dp"  
 *android:layout\_height*="73dp"  
 *android:layout\_marginStart*="28dp"  
 *android:layout\_marginTop*="16dp"  
 *android:src*="@drawable/ico27"  
 *app:layout\_constraintStart\_toEndOf*="@+id/ico26"  
 *app:layout\_constraintTop\_toBottomOf*="@+id/ico23" />  
 <ImageView  
 *android:id*="@+id/ico28"  
 *android:layout\_width*="73dp"  
 *android:layout\_height*="73dp"  
 *android:layout\_marginTop*="16dp"  
 *android:src*="@drawable/ico28"  
 *app:layout\_constraintEnd\_toEndOf*="parent"  
 *app:layout\_constraintHorizontal\_bias*="0.644"  
 *app:layout\_constraintStart\_toEndOf*="@+id/ico27"  
 *app:layout\_constraintTop\_toBottomOf*="@+id/ico24" />  
  
 *<!-- 8 строка -->* <ImageView  
 *android:id*="@+id/ico29"  
 *android:layout\_width*="73dp"  
 *android:layout\_height*="73dp"  
 *android:layout\_marginStart*="17dp"  
 *android:layout\_marginTop*="16dp"  
 *android:src*="@drawable/ico29"  
 *app:layout\_constraintStart\_toStartOf*="parent"  
 *app:layout\_constraintTop\_toBottomOf*="@+id/ico25" />  
 <ImageView  
 *android:id*="@+id/ico30"  
 *android:layout\_width*="73dp"  
 *android:layout\_height*="73dp"  
 *android:layout\_marginStart*="28dp"  
 *android:layout\_marginTop*="16dp"  
 *android:src*="@drawable/ico30"  
 *app:layout\_constraintStart\_toEndOf*="@+id/ico29"  
 *app:layout\_constraintTop\_toBottomOf*="@+id/ico26" />  
 <ImageView  
 *android:id*="@+id/ico31"  
 *android:layout\_width*="73dp"  
 *android:layout\_height*="73dp"  
 *android:layout\_marginStart*="28dp"  
 *android:layout\_marginTop*="16dp"  
 *android:src*="@drawable/ico31"  
 *app:layout\_constraintStart\_toEndOf*="@+id/ico30"  
 *app:layout\_constraintTop\_toBottomOf*="@+id/ico27" />  
 <ImageView  
 *android:id*="@+id/ico32"  
 *android:layout\_width*="73dp"  
 *android:layout\_height*="73dp"  
 *android:layout\_marginTop*="16dp"  
 *android:src*="@drawable/ico32"  
 *app:layout\_constraintEnd\_toEndOf*="parent"  
 *app:layout\_constraintHorizontal\_bias*="0.644"  
 *app:layout\_constraintStart\_toEndOf*="@+id/ico31"  
 *app:layout\_constraintTop\_toBottomOf*="@+id/ico28" />  
  
 *<!-- 9 строка -->* <ImageView  
 *android:id*="@+id/ico33"  
 *android:layout\_width*="73dp"  
 *android:layout\_height*="73dp"  
 *android:layout\_marginStart*="17dp"  
 *android:layout\_marginTop*="16dp"  
 *android:src*="@drawable/ico33"  
 *app:layout\_constraintStart\_toStartOf*="parent"  
 *app:layout\_constraintTop\_toBottomOf*="@+id/ico29" />  
 <ImageView  
 *android:id*="@+id/ico34"  
 *android:layout\_width*="73dp"  
 *android:layout\_height*="73dp"  
 *android:layout\_marginStart*="28dp"  
 *android:layout\_marginTop*="16dp"  
 *android:src*="@drawable/ico34"  
 *app:layout\_constraintStart\_toEndOf*="@+id/ico33"  
 *app:layout\_constraintTop\_toBottomOf*="@+id/ico30" />  
 <ImageView  
 *android:id*="@+id/ico35"  
 *android:layout\_width*="73dp"  
 *android:layout\_height*="73dp"  
 *android:layout\_marginStart*="28dp"  
 *android:layout\_marginTop*="16dp"  
 *android:src*="@drawable/ico35"  
 *app:layout\_constraintStart\_toEndOf*="@+id/ico34"  
 *app:layout\_constraintTop\_toBottomOf*="@+id/ico31" />  
 <ImageView  
 *android:id*="@+id/ico36"  
 *android:layout\_width*="73dp"  
 *android:layout\_height*="73dp"  
 *android:layout\_marginTop*="16dp"  
 *android:src*="@drawable/ico36"  
 *app:layout\_constraintEnd\_toEndOf*="parent"  
 *app:layout\_constraintHorizontal\_bias*="0.644"  
 *app:layout\_constraintStart\_toEndOf*="@+id/ico35"  
 *app:layout\_constraintTop\_toBottomOf*="@+id/ico32" />  
  
 <Space  
 *android:id*="@+id/space"  
 *android:layout\_width*="match\_parent"  
 *android:layout\_height*="140dp"  
 *app:layout\_constraintTop\_toBottomOf*="@+id/ico36" />  
  
 </androidx.constraintlayout.widget.ConstraintLayout>  
  
 </ScrollView>  
  
</androidx.constraintlayout.widget.ConstraintLayout>

1. IconSelectionScreen.kt:

*package* com.bignerdranch.android.cinemaapp  
  
*import* android.content.Intent  
*import* androidx.appcompat.app.AppCompatActivity  
*import* android.os.Bundle  
*import* android.widget.ImageView  
  
*class* IconSelectionScreen : AppCompatActivity() {  
 *override fun* onCreate(savedInstanceState: Bundle?) {  
 *super*.onCreate(savedInstanceState)  
 setContentView(R.layout.*icon\_selection\_screen*)  
 *supportActionBar*?.setDisplayHomeAsUpEnabled(*true*)  
  
 *val* backButton = findViewById<ImageView>(R.id.*backImage*)  
 backButton.setOnClickListener **{** setResult(*RESULT\_CANCELED*)  
 finish()  
 **}** *val* ico1 = findViewById<ImageView>(R.id.*ico1*)  
 ico1.setOnClickListener **{** selectIcon(R.drawable.*ico1*) **}** *val* ico2 = findViewById<ImageView>(R.id.*ico2*)  
 ico2.setOnClickListener **{** selectIcon(R.drawable.*ico2*) **}** *val* ico3 = findViewById<ImageView>(R.id.*ico3*)  
 ico3.setOnClickListener **{** selectIcon(R.drawable.*ico3*) **}** *val* ico4 = findViewById<ImageView>(R.id.*ico4*)  
 ico4.setOnClickListener **{** selectIcon(R.drawable.*ico4*) **}** *val* ico5 = findViewById<ImageView>(R.id.*ico5*)  
 ico5.setOnClickListener **{** selectIcon(R.drawable.*ico5*) **}** *val* ico6 = findViewById<ImageView>(R.id.*ico6*)  
 ico6.setOnClickListener **{** selectIcon(R.drawable.*ico6*) **}** *val* ico7 = findViewById<ImageView>(R.id.*ico7*)  
 ico7.setOnClickListener **{** selectIcon(R.drawable.*ico7*) **}** *val* ico8 = findViewById<ImageView>(R.id.*ico8*)  
 ico8.setOnClickListener **{** selectIcon(R.drawable.*ico8*) **}** *val* ico9 = findViewById<ImageView>(R.id.*ico9*)  
 ico9.setOnClickListener **{** selectIcon(R.drawable.*ico9*) **}** *val* ico10 = findViewById<ImageView>(R.id.*ico10*)  
 ico10.setOnClickListener **{** selectIcon(R.drawable.*ico10*) **}** *val* ico11 = findViewById<ImageView>(R.id.*ico11*)  
 ico11.setOnClickListener **{** selectIcon(R.drawable.*ico11*) **}** *val* ico12 = findViewById<ImageView>(R.id.*ico12*)  
 ico12.setOnClickListener **{** selectIcon(R.drawable.*ico12*) **}** *val* ico13 = findViewById<ImageView>(R.id.*ico13*)  
 ico13.setOnClickListener **{** selectIcon(R.drawable.*ico13*) **}** *val* ico14 = findViewById<ImageView>(R.id.*ico14*)  
 ico14.setOnClickListener **{** selectIcon(R.drawable.*ico14*) **}** *val* ico15 = findViewById<ImageView>(R.id.*ico15*)  
 ico15.setOnClickListener **{** selectIcon(R.drawable.*ico15*) **}** *val* ico16 = findViewById<ImageView>(R.id.*ico16*)  
 ico16.setOnClickListener **{** selectIcon(R.drawable.*ico16*) **}** *val* ico17 = findViewById<ImageView>(R.id.*ico17*)  
 ico17.setOnClickListener **{** selectIcon(R.drawable.*ico17*) **}** *val* ico18 = findViewById<ImageView>(R.id.*ico18*)  
 ico18.setOnClickListener **{** selectIcon(R.drawable.*ico18*) **}** *val* ico19 = findViewById<ImageView>(R.id.*ico19*)  
 ico19.setOnClickListener **{** selectIcon(R.drawable.*ico19*) **}** *val* ico20 = findViewById<ImageView>(R.id.*ico20*)  
 ico20.setOnClickListener **{** selectIcon(R.drawable.*ico20*) **}** *val* ico21 = findViewById<ImageView>(R.id.*ico21*)  
 ico21.setOnClickListener **{** selectIcon(R.drawable.*ico21*) **}** *val* ico22 = findViewById<ImageView>(R.id.*ico22*)  
 ico22.setOnClickListener **{** selectIcon(R.drawable.*ico22*) **}** *val* ico23 = findViewById<ImageView>(R.id.*ico23*)  
 ico23.setOnClickListener **{** selectIcon(R.drawable.*ico23*) **}** *val* ico24 = findViewById<ImageView>(R.id.*ico24*)  
 ico24.setOnClickListener **{** selectIcon(R.drawable.*ico24*) **}** *val* ico25 = findViewById<ImageView>(R.id.*ico25*)  
 ico25.setOnClickListener **{** selectIcon(R.drawable.*ico25*) **}** *val* ico26 = findViewById<ImageView>(R.id.*ico26*)  
 ico26.setOnClickListener **{** selectIcon(R.drawable.*ico26*) **}** *val* ico27 = findViewById<ImageView>(R.id.*ico27*)  
 ico27.setOnClickListener **{** selectIcon(R.drawable.*ico27*) **}** *val* ico28 = findViewById<ImageView>(R.id.*ico28*)  
 ico28.setOnClickListener **{** selectIcon(R.drawable.*ico28*) **}** *val* ico29 = findViewById<ImageView>(R.id.*ico29*)  
 ico29.setOnClickListener **{** selectIcon(R.drawable.*ico29*) **}** *val* ico30 = findViewById<ImageView>(R.id.*ico30*)  
 ico30.setOnClickListener **{** selectIcon(R.drawable.*ico30*) **}** *val* ico31 = findViewById<ImageView>(R.id.*ico31*)  
 ico31.setOnClickListener **{** selectIcon(R.drawable.*ico31*) **}** *val* ico32 = findViewById<ImageView>(R.id.*ico32*)  
 ico32.setOnClickListener **{** selectIcon(R.drawable.*ico32*) **}** *val* ico33 = findViewById<ImageView>(R.id.*ico33*)  
 ico33.setOnClickListener **{** selectIcon(R.drawable.*ico33*) **}** *val* ico34 = findViewById<ImageView>(R.id.*ico34*)  
 ico34.setOnClickListener **{** selectIcon(R.drawable.*ico34*) **}** *val* ico35 = findViewById<ImageView>(R.id.*ico35*)  
 ico35.setOnClickListener **{** selectIcon(R.drawable.*ico35*) **}** *val* ico36 = findViewById<ImageView>(R.id.*ico36*)  
 ico36.setOnClickListener **{** selectIcon(R.drawable.*ico36*) **}** }  
  
 *private fun* selectIcon(iconId: Int) {  
 *val* resultIntent = Intent()  
 resultIntent.putExtra(EXTRA\_SELECTED\_ICON, iconId)  
 setResult(*RESULT\_OK*, resultIntent)  
 finish()  
 }  
  
 *companion object* {  
 *const val* EXTRA\_SELECTED\_ICON = "selected\_icon"  
 }  
}

1. CollectionAdapter.kt:

*package* com.bignerdranch.android.cinemaapp  
  
*import* android.view.LayoutInflater  
*import* android.view.View  
*import* android.view.ViewGroup  
*import* android.widget.ImageView  
*import* android.widget.PopupMenu  
*import* android.widget.TextView  
*import* androidx.recyclerview.widget.RecyclerView  
  
*// Adapter для отображения коллекций  
class* CollectionAdapter(*private val* collections: List<Collection>) :  
 RecyclerView.Adapter<CollectionAdapter.CollectionViewHolder>() {  
  
 *private var* clickListener: OnCollectionClickListener? = *null  
  
 interface* OnCollectionClickListener {  
 *fun* onDeleteClicked(collection: Collection)  
 }  
  
 *fun* setOnCollectionClickListener(listener: OnCollectionClickListener) {  
 clickListener = listener  
 }  
  
 *override fun* onCreateViewHolder(parent: ViewGroup, viewType: Int): CollectionViewHolder {  
 *val* view = LayoutInflater.from(parent.*context*)  
 .inflate(R.layout.*item\_collection*, parent, *false*)  
 *return* CollectionViewHolder(view)  
 }  
  
 *override fun* onBindViewHolder(holder: CollectionViewHolder, position: Int) {  
 *val* collection = collections[position]  
 holder.bind(collection)  
 }  
  
 *override fun* getItemCount(): Int {  
 *return* collections.size  
 }  
  
 *inner class* CollectionViewHolder(itemView: View) : RecyclerView.ViewHolder(itemView) {  
  
 *private val* nameTextView: TextView = itemView.findViewById(R.id.*nameTextView*)  
 *private val* iconImageView: ImageView = itemView.findViewById(R.id.*iconImageView*)  
 *private val* deleteImageView: ImageView = itemView.findViewById(R.id.*deleteImageView*)  
  
 *fun* bind(collection: Collection) {  
 nameTextView.*text* = collection.name  
 iconImageView.setImageResource(collection.iconId)  
  
 deleteImageView.setOnClickListener **{** *val* popupMenu = PopupMenu(itemView.*context*, deleteImageView)  
 popupMenu.inflate(R.menu.*menu\_delete*)  
 popupMenu.setOnMenuItemClickListener **{** item **->** *when* (item.*itemId*) {  
 R.id.*menu\_del* -> {  
 clickListener?.onDeleteClicked(collection)  
 *true* }  
 *else* -> *false* }  
 **}** popupMenu.show()  
 **}** }  
 }  
}

1. Collection.kt:

*package* com.bignerdranch.android.cinemaapp  
  
*data class* Collection(*val* name: String, *val* iconId: Int)

1. NavigationCollectionsFragment.kt:

*package* com.bignerdranch.android.cinemaapp  
  
*import* android.app.Activity.*RESULT\_OK  
import* android.content.Context  
*import* android.content.Intent  
*import* android.os.Bundle  
*import* androidx.fragment.app.Fragment  
*import* android.view.LayoutInflater  
*import* android.view.View  
*import* android.view.ViewGroup  
*import* android.widget.Button  
*import* android.widget.PopupMenu  
*import* androidx.recyclerview.widget.LinearLayoutManager  
*import* androidx.recyclerview.widget.RecyclerView  
  
*class* NavigationCollectionsFragment : Fragment(), CollectionAdapter.OnCollectionClickListener {  
  
 *private lateinit var* collectionAdapter: CollectionAdapter  
 *private val* collections = *mutableListOf*<Collection>()  
 *private lateinit var* sharedPrefsHelper: SharedPrefsHelper  
  
 *override fun* onAttach(context: Context) {  
 *super*.onAttach(context)  
 *// инициализация SharedPrefsHelper* sharedPrefsHelper = SharedPrefsHelper(requireContext())  
 }  
  
 *override fun* onCreateView(inflater: LayoutInflater, container: ViewGroup?, savedInstanceState: Bundle?): View? {  
 *val* view = inflater.inflate(R.layout.*activity\_navigation\_collections*, container, *false*)  
  
 *// инициализация адаптера коллекций* collectionAdapter = CollectionAdapter(collections)  
 collectionAdapter.setOnCollectionClickListener(*this*)  
  
 *// установка адаптера для RecyclerView  
 val* recyclerView = view.findViewById<RecyclerView>(R.id.*collectionsRecyclerView*)  
 recyclerView.*adapter* = collectionAdapter  
  
 *val* layoutManager = LinearLayoutManager(requireContext())  
 recyclerView.*layoutManager* = layoutManager  
  
 *// Загрузка коллекций из Shared Preferences* collections.addAll(sharedPrefsHelper.loadCollections())  
 collectionAdapter.notifyDataSetChanged()  
  
 *val* createCollectionButton = view.findViewById<Button>(R.id.*createCollectionButton*)  
 createCollectionButton.setOnClickListener **{** *val* popupMenu = PopupMenu(requireContext(), createCollectionButton)  
 popupMenu.inflate(R.menu.*menu\_add*)  
 popupMenu.setOnMenuItemClickListener **{** item **->** *when* (item.*itemId*) {  
 R.id.*menu\_add* -> {  
 *// запуск активности CreateCollectionScreen для создания новой коллекции  
 val* intent = Intent(requireContext(), CreateCollectionScreen::*class*.*java*)  
 startActivityForResult(intent, REQUEST\_CREATE\_COLLECTION)  
 *true* }  
 *else* -> *false* }  
 **}** popupMenu.show()  
 **}** *return* view  
 }  
  
 *// обработка результата возвращаемого из активности  
 override fun* onActivityResult(requestCode: Int, resultCode: Int, data: Intent?) {  
 *super*.onActivityResult(requestCode, resultCode, data)  
 *if* (requestCode == REQUEST\_CREATE\_COLLECTION && resultCode == *RESULT\_OK*) {  
 *val* name = data?.getStringExtra(CreateCollectionScreen.EXTRA\_COLLECTION\_NAME)  
 *val* iconId = data?.getIntExtra(CreateCollectionScreen.EXTRA\_SELECTED\_ICON, R.drawable.*ico1*)  
 *if* (name != *null* && iconId != *null*) {  
 *val* collection = Collection(name, iconId)  
 collections.add(collection)  
 collectionAdapter.notifyDataSetChanged()  
 sharedPrefsHelper.saveCollections(collections)  
 }  
 }  
 }  
  
 *// удаление выбранной коллекции  
 override fun* onDeleteClicked(collection: Collection) {  
 collections.remove(collection)  
 collectionAdapter.notifyDataSetChanged()  
 sharedPrefsHelper.saveCollections(collections)  
 }  
  
 *// код запроса для создания коллекции  
 companion object* {  
 *private const val* REQUEST\_CREATE\_COLLECTION = 1  
 }  
}

1. MovieDao.kt:

*package* com.bignerdranch.android.cinemaapp  
  
*import* androidx.room.Dao  
*import* androidx.room.Delete  
*import* androidx.room.Insert  
*import* androidx.room.Query  
  
*// интерфейс доступа к данным (DAO) для работы с таблицей фильмов*@Dao  
*interface* MovieDao {  
 @Query("SELECT \* FROM movies")  
 *fun* getAllMovies(): List<Movie>  
  
 @Insert  
 *fun* addMovie(movie: Movie)  
  
 @Delete  
 *fun* deleteMovie(movie: Movie)  
  
 @Query("SELECT \* FROM movies WHERE genre = :genre")  
 *fun* getMoviesByGenre(genre: String): List<Movie>  
}

1. MovieAdapter.kt:

*package* com.bignerdranch.android.cinemaapp  
  
*import* android.net.Uri  
*import* android.view.LayoutInflater  
*import* android.view.View  
*import* android.view.ViewGroup  
*import* android.widget.ImageView  
*import* android.widget.PopupMenu  
*import* android.widget.TextView  
*import* androidx.lifecycle.LifecycleCoroutineScope  
*import* androidx.recyclerview.widget.RecyclerView  
*import* kotlinx.coroutines.Dispatchers  
*import* kotlinx.coroutines.launch  
*import* kotlinx.coroutines.withContext  
  
*class* MovieAdapter(  
 *private var* movies: List<Movie>,  
 *private val* lifecycleScope: LifecycleCoroutineScope,  
 *private val* movieDao: MovieDao,  
 *private val* onItemClick: (Movie) -> Unit,  
 *private val* loadMovies: (() -> Unit)? = *null*) : RecyclerView.Adapter<MovieAdapter.MovieViewHolder>() {  
  
 *override fun* onCreateViewHolder(parent: ViewGroup, viewType: Int): MovieViewHolder {  
 *val* view = LayoutInflater.from(parent.*context*).inflate(R.layout.*movie\_item*, parent, *false*)  
 *return* MovieViewHolder(view)  
 }  
  
 *override fun* onBindViewHolder(holder: MovieViewHolder, position: Int) {  
 *val* movie = movies[position]  
 holder.bind(movie)  
 }  
  
 *override fun* getItemCount(): Int = movies.size  
  
 *fun* setData(movies: List<Movie>) {  
 *this*.movies = movies  
 notifyDataSetChanged()  
 }  
  
 *inner class* MovieViewHolder(itemView: View) : RecyclerView.ViewHolder(itemView) {  
 *private val* iconImageView: ImageView = itemView.findViewById(R.id.*iconImageView*)  
 *private val* titleTextView: TextView = itemView.findViewById(R.id.*titleTextView*)  
 *private val* deleteImageView: ImageView = itemView.findViewById(R.id.*deleteImageView*)  
  
 *init* {  
 itemView.setOnClickListener **{** *val* position = *adapterPosition  
 if* (position != RecyclerView.*NO\_POSITION*) {  
 *val* movie = movies[position]  
 onItemClick(movie)  
 }  
 **}** }  
  
 *// установка данных фильма для элементов интерфейса  
 fun* bind(movie: Movie) {  
 iconImageView.setImageURI(Uri.parse(movie.icon))  
 titleTextView.*text* = movie.title  
  
 deleteImageView.setOnClickListener **{** *val* popupMenu = PopupMenu(itemView.*context*, deleteImageView)  
 popupMenu.inflate(R.menu.*menu\_delete\_film*)  
 popupMenu.setOnMenuItemClickListener **{** item **->** *when* (item.*itemId*) {  
 R.id.*menu\_del* -> {  
 *val* movie = movies[*adapterPosition*]  
 deleteMovie(movie)  
 *true* }  
 *else* -> *false* }  
 **}** popupMenu.show()  
 **}** }  
  
 *// удаление фильма из базы данных и обновление списка  
 private fun* deleteMovie(movie: Movie) {  
 lifecycleScope.launch(Dispatchers.IO) **{** movieDao.deleteMovie(movie)  
 withContext(Dispatchers.Main) **{** loadMovies?.invoke()  
 **}  
 }** }  
 }  
}

1. Movie.kt:

*package* com.bignerdranch.android.cinemaapp  
  
*import* android.os.Parcel  
*import* android.os.Parcelable  
*import* androidx.room.Entity  
*import* androidx.room.PrimaryKey  
  
*// сущность (Entity) для модели фильма*@Entity(tableName = "movies")  
*data class* Movie(  
 @PrimaryKey(autoGenerate = *true*)  
 *val* id: Int = 0,  
 *val* icon: String,  
 *val* title: String,  
 *val* genre: String,  
 *val* author: String  
  
 *// интерфейс Parcelable для класса Movie для передачи между Activity*) : Parcelable {  
 *constructor*(parcel: Parcel) : *this*(  
 parcel.readInt(),  
 parcel.readString() ?: "",  
 parcel.readString() ?: "",  
 parcel.readString() ?: "",  
 parcel.readString() ?: ""  
 )  
  
 *override fun* writeToParcel(parcel: Parcel, flags: Int) {  
 parcel.writeInt(id)  
 parcel.writeString(icon)  
 parcel.writeString(title)  
 parcel.writeString(genre)  
 parcel.writeString(author)  
 }  
  
 *override fun* describeContents(): Int {  
 *return* 0  
 }  
  
 *companion object* CREATOR : Parcelable.Creator<Movie> {  
 *override fun* createFromParcel(parcel: Parcel): Movie {  
 *return* Movie(parcel)  
 }  
  
 *override fun* newArray(size: Int): Array<Movie?> {  
 *return arrayOfNulls*(size)  
 }  
 }  
}

1. AppDatabase.kt:

*package* com.bignerdranch.android.cinemaapp  
  
*import* androidx.room.Database  
*import* androidx.room.RoomDatabase  
  
*// класс базы данных, который будет использоваться Room*@Database(entities = [Movie::*class*], version = 1)  
*abstract class* AppDatabase : RoomDatabase() {  
 *abstract fun* movieDao(): MovieDao  
}

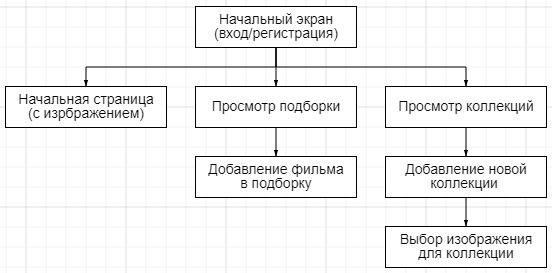
Это основные файлы проекта, которые определены в коде.

## Описание разработанных функций

Вот описание основных функций, реализованных в коде:

1. **onAttach(context: Context)**: Метод, вызываемый при присоединении фрагмента к контексту. В данном методе происходит инициализация **SharedPrefsHelper**, который используется для работы с SharedPreferences.
2. **onCreateView(inflater: LayoutInflater, container: ViewGroup?, savedInstanceState: Bundle?): View?**: Метод, вызываемый при создании пользовательского интерфейса фрагмента. В этом методе происходит инициализация и настройка элементов пользовательского интерфейса, адаптера для RecyclerView и загрузка коллекций из Shared Preferences.
3. **onActivityResult(requestCode: Int, resultCode: Int, data: Intent?)**: Метод, вызываемый для обработки результата, возвращаемого из активности. В данном методе обрабатывается результат создания новой коллекции. Если результат успешный, то создается новый объект **Collection** и добавляется в список коллекций. Затем происходит сохранение списка коллекций в Shared Preferences.
4. **onDeleteClicked(collection: Collection)**: Метод, вызываемый при нажатии на кнопку удаления выбранной коллекции. В данном методе выбранная коллекция удаляется из списка коллекций, а затем список коллекций сохраняется в Shared Preferences.
5. **onCreate(savedInstanceState: Bundle?)**: Метод, вызываемый при создании экрана для создания новой коллекции. В этом методе происходит настройка элементов пользовательского интерфейса, установка обработчиков событий для кнопок "Назад", "Выбрать иконку" и "Создать", а также обработка результата выбора иконки и создания новой коллекции.
6. **onActivityResult(requestCode: Int, resultCode: Int, data: Intent?)**: Метод, вызываемый для обработки результата, возвращаемого из активности. В данном методе обрабатывается результат выбора иконки для новой коллекции. Если результат успешный, то выбранная иконка устанавливается в **ImageView** и сохраняется в результате создания новой коллекции.
7. **onCreateView(inflater: LayoutInflater, container: ViewGroup?, savedInstanceState: Bundle?): View?**: Метод, вызываемый при создании пользовательского интерфейса фрагмента. В этом методе происходит инициализация элементов пользовательского интерфейса, адаптера для RecyclerView и базы данных. Также устанавливается слушатель выбора элемента в спиннере для сортировки фильмов по жанру.
8. **loadMoviesByGenre(genre: String)**: Метод для загрузки фильмов по выбранному жанру из базы данных и обновления адаптера с фильмами.
9. **loadMovies()**: Метод для загрузки всех фильмов из базы данных и обновления адаптера с фильмами.
10. **openGallery()**: Метод для открытия галереи, чтобы выбрать изображение для добавления фильма.
11. **onActivityResult(requestCode: Int, resultCode: Int, data: Intent?)**: Метод, вызываемый для обработки результата, возвращаемого из активности. В данном методе обрабатывается результат выбора изображения из галереи. Выбранное изображение устанавливается в **ImageView**.
12. **showAddMovieDialog()**: Метод для отображения диалогового окна, позволяющего пользователю добавить новый фильм. В этом методе настраиваются элементы пользовательского интерфейса диалогового окна, включая кнопку добавления изображения из галереи. После заполнения всех необходимых полей и выбора изображения, фильм добавляется в базу данных и обновляется список фильмов.
13. **onMovieClicked(movie: Movie)**: Метод, вызываемый при клике на фильм в RecyclerView. В данном методе открывается активность с подробной информацией о фильме.

## Алгоритм решения



## Используемые библиотеки

В проекте используются следующие библиотеки:

*// импорты для библиотек AndroidX*implementation 'androidx.appcompat:appcompat:1.4.0'  
implementation 'androidx.constraintlayout:constraintlayout:2.1.0'  
implementation 'androidx.lifecycle:lifecycle-runtime:2.3.1'  
implementation 'androidx.lifecycle:lifecycle-viewmodel-ktx:2.3.1'  
implementation 'androidx.navigation:navigation-fragment-ktx:2.4.0'  
implementation 'androidx.navigation:navigation-ui-ktx:2.4.0'  
implementation 'androidx.recyclerview:recyclerview:1.2.1'  
implementation 'com.google.android.material:material:1.5.0'  
implementation 'androidx.preference:preference-ktx:1.1.1'  
  
*// импорт для библиотеки Gson*implementation 'com.google.code.gson:gson:2.8.8'  
  
*// импорты для библиотеки Room*implementation 'androidx.room:room-runtime:2.4.0'  
implementation 'androidx.room:room-ktx:2.4.0'  
kapt 'androidx.room:room-compiler:2.4.0'  
  
*// импорты для модулей тестирования*testImplementation 'junit:junit:4.13.2'  
androidTestImplementation 'androidx.test:runner:1.4.0'  
androidTestImplementation 'androidx.test.espresso:espresso-core:3.4.0'

Это основные библиотеки, которые я использовал в проекте.

## Тестовые случаи

Имеется проверка на ввод пустых данных в поля.

## Используемые инструменты

* Язык программирования: Kotlin
* Среда разработки: Android Studio
* Фреймворк: Android SDK

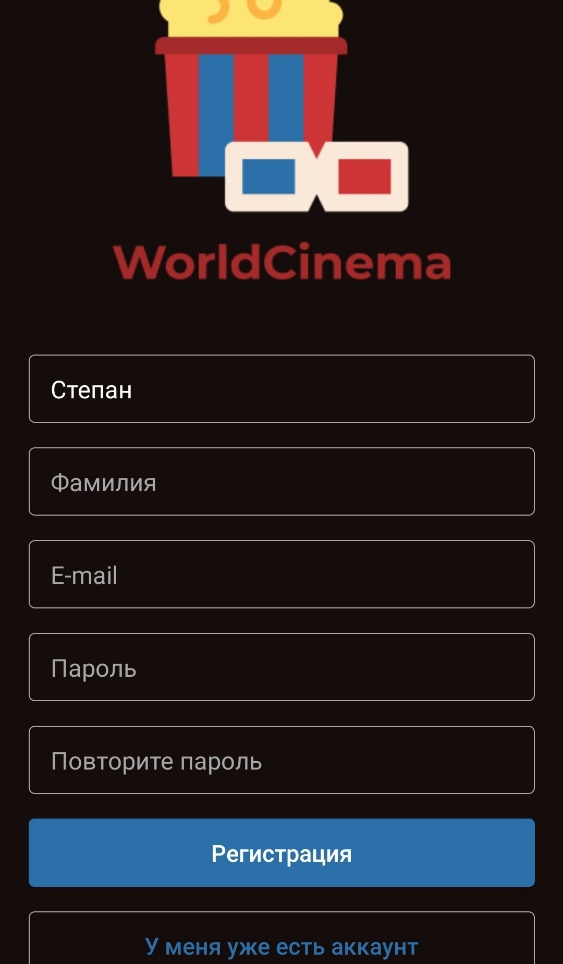
## Описание пользовательского интерфейса

Описание пользовательского интерфейса:

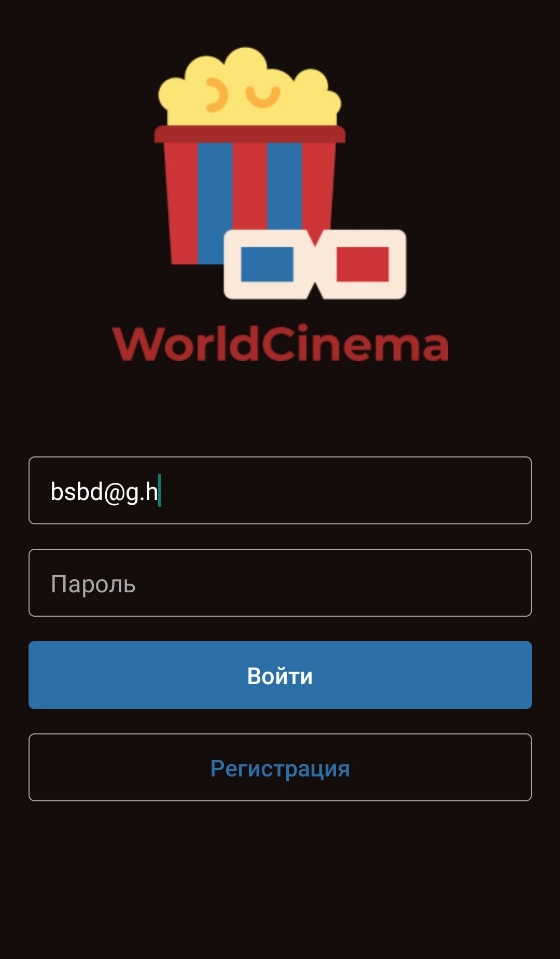
На начальном экране можно зарегистрироваться или войти, если уже была регистрация. На следующем экране определены 3 основные окна: главная, подборка и коллекции. В подборке можно добавить свой фильм, выбрав фото из галереи, введя название и автора фильма. Можно также осуществить сортировку по жанру фильма. При нажатии на кнопку добавления коллекции, появляется специальное меню. Можно ввести название и выбрать изображение для новой коллекции.

## Приложение (print screen)

1. Регистрация:



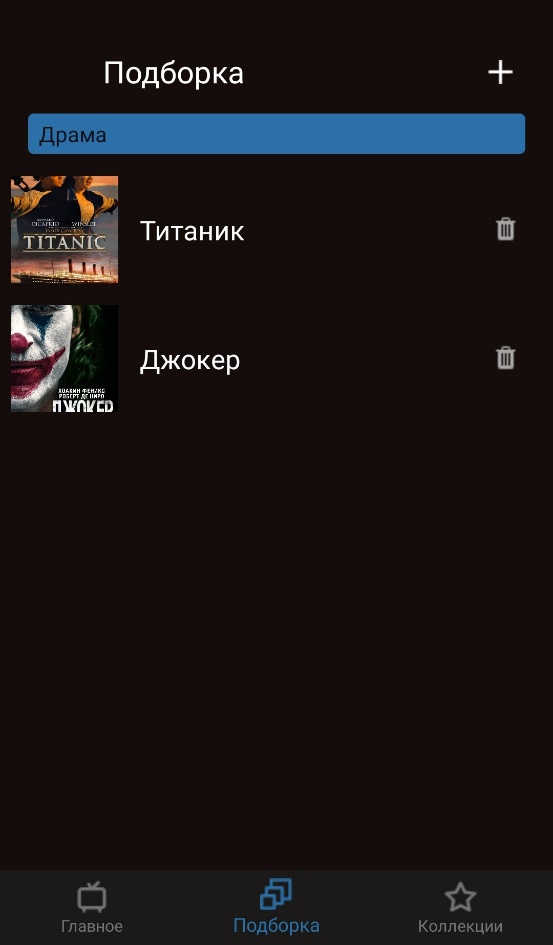
1. Вход:



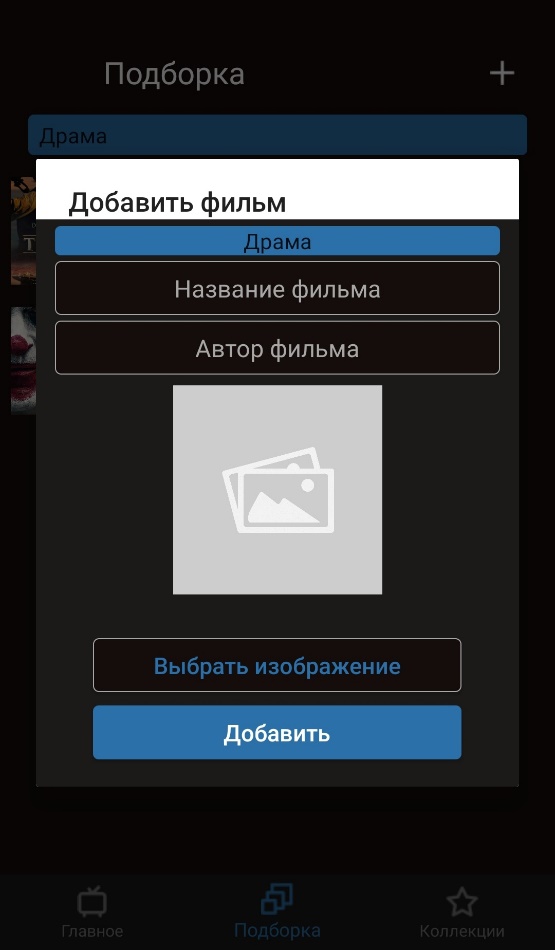
1. Главное:



1. Подборка:



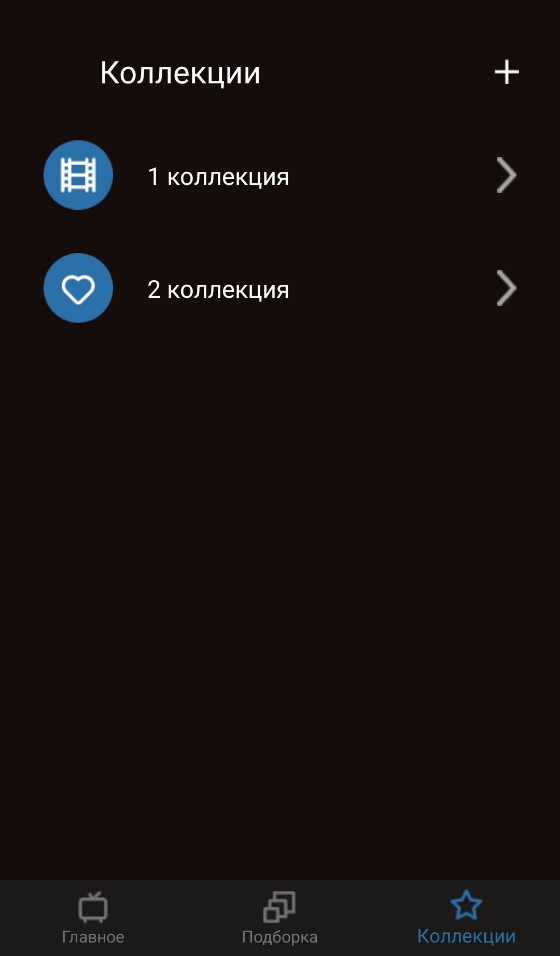
1. Добавление фильма в подборку:



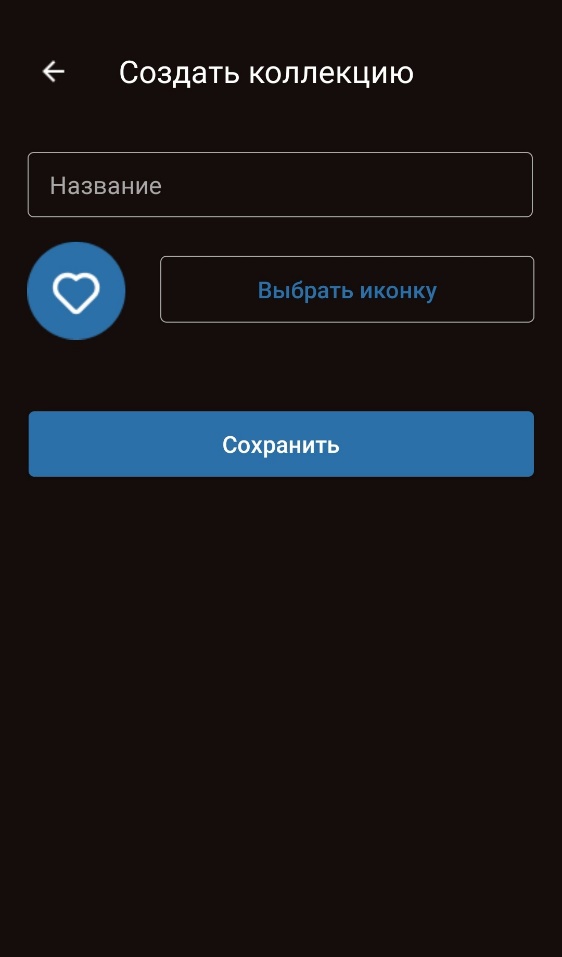
1. Детали фильма:



1. Коллекции:



1. Создание коллекции:



1. Выбор иконки для коллекции:

