Министерство образования и молодежной политики Свердловской области



ГАПОУ СО «Екатеринбургский колледж транспортного строительства»

Отчёт по программе «**Приложение для TV**»

Выполнил: Ковязин А.М

Группа: ПР-31

Преподаватель: Мирошниченко Г.В

2023

**Входные данные:**

login(String) – логин

password(String) – пароль  
search(String) – название для поиска фильма

**Выходные данные:**

**recyclerView –** список

**Toast – сообщение**

**AlertDialog - сообщение**

**Листинг программы:**

class AdapterFill(private val context: Context, private val list: MutableList<Movie>) : RecyclerView.Adapter<AdapterFill.MyVH>() {  
 override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): MyVH {  
 val root = LayoutInflater.from(context).inflate(R.layout.item, parent, false)  
 return MyVH(root)  
 }  
 fun updateList(newList: List<Movie>) {  
 list.clear()  
 list.addAll(newList)  
 }  
 inner class MyVH(itemView: View) : RecyclerView.ViewHolder(itemView) {  
 val imageView: ImageView = itemView.findViewById(R.id.questsImageView)  
 val title: TextView = itemView.findViewById(R.id.titleQuestsTextView)  
 val description: TextView = itemView.findViewById(R.id.descriptionTextView)  
 }  
  
 override fun onBindViewHolder(holder: MyVH, position: Int) {  
 Picasso.get().load(list[position].img).into(holder.imageView)  
 holder.title.setText(list[position].name)  
 holder.description.setText(list[position].info)  
 }  
  
 override fun getItemCount(): Int{  
 return list.size  
 }  
}

interface Api {  
 @GET("/")  
 fun getMovieDetails(  
 @Query("apikey") apiKey: String,  
 @Query("t") title: String,  
 @Query("plot") plot: String = "full"  
 ): Call<Search>  
}

class BaseFragment : Fragment() {  
 // *TODO: Rename and change types of parameters* private var param1: String? = null  
 private var param2: String? = null  
 var answer: MutableList<Movie> = *mutableListOf*()  
  
 override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
 arguments?.let **{** param1 = it.getString(*ARG\_PARAM1*)  
 param2 = it.getString(*ARG\_PARAM2*)  
 **}** }  
   
 override fun onCreateView(  
 inflater: LayoutInflater, container: ViewGroup?,  
 savedInstanceState: Bundle?  
 ): View? {  
 // Inflate the layout for this fragment  
  
 val fragmentLayout=inflater.inflate(R.layout.base\_fragment, container, false)  
 val navController=NavHostFragment.findNavController(this)  
  
 val rec:RecyclerView=fragmentLayout.findViewById(R.id.recyclerView)  
 rec.layoutManager= GridLayoutManager(requireContext(),3)  
 rec.adapter=QuestRecycler(requireContext(),MyObj().list)  
  
 return fragmentLayout  
  
  
 }  
  
  
  
  
 override fun onOptionsItemSelected(item: MenuItem): Boolean {  
 when (item.itemId) {  
 R.id.fragmentHoror -> {  
 findNavController().navigate(R.id.action\_fragmentGlav\_to\_fragmentHoror)  
 return true  
 }  
 R.id.fragmentDetectiv -> {  
 findNavController().navigate(R.id.action\_fragmentGlav\_to\_fragmentDetectiv)  
 return true  
 }  
 R.id.fragmentSearch -> {  
 findNavController().navigate(R.id.action\_fragmentGlav\_to\_fragmentSearch)  
 return true  
 }  
 }  
 return super.onOptionsItemSelected(item)}  
  
  
  
}

class DetectiveFragment : Fragment() {  
 // TODO: Rename and change types of parameters  
 private var param1: String? = null  
 private var param2: String? = null  
 private lateinit var recyclerView: RecyclerView  
 var answer: MutableList<Movie> = mutableListOf()  
  
 override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
 arguments?.let **{** param1 = it.getString(ARG\_PARAM1)  
 param2 = it.getString(ARG\_PARAM2)  
 **}** }  
  
 override fun onCreateView(  
 inflater: LayoutInflater, container: ViewGroup?,  
 savedInstanceState: Bundle?  
 ): View? {  
 // Inflate the layout for this fragment  
 val fragmentLayout = inflater.inflate(R.layout.detective\_fragment, container, false)  
 val navController = NavHostFragment.findNavController(this)  
 val fetchMoviesTask = FetchMoviesTask(fragmentLayout)  
 fetchMoviesTask.execute()  
  
 return fragmentLayout  
 }  
 private inner class FetchMoviesTask(val fragmentLayout: View) : AsyncTask<Void, Void, String>() {  
 override fun doInBackground(vararg params: Void?): String? {  
 val apiKey = "43181e24"  
  
 try {  
  
 val apiUrl = "https://www.omdbapi.com/?apikey=$apiKey&s=detective&type=movie&plot=full"  
  
  
 val url = URL(apiUrl)  
 val connection = url.openConnection() as HttpURLConnection  
 val scanner = Scanner(connection.inputStream, "UTF-8").useDelimiter("\\A")  
 return if (scanner.hasNext()) scanner.next() else null  
 } catch (e: Exception) {  
 Log.e(ContentValues.TAG, "Ошибка при выполнении запроса: ${e.message}")  
 return null  
 }  
 }  
  
 override fun onPostExecute(result: String?) {  
 super.onPostExecute(result)  
  
  
 if (result != null) {  
 val gson = Gson()  
 val data = gson.fromJson(result, Map::class.java)  
  
  
 if (data.containsKey("Search")) {  
 val top5HorrorMovies = (data["Search"] as List<Map<String, String>>).take(5)  
 top5HorrorMovies.forEach **{** movie **->** val img = "${movie["Poster"]}"  
 val name = "${movie["Title"]}"  
 val info = "Год выпуска: ${movie["Year"]}"  
 val wr: Movie = Movie( name, img, info)  
 answer.add(wr)  
 if(answer.size == 5){  
 val rec: RecyclerView = fragmentLayout.findViewById(R.id.recyclerView2)  
 rec.layoutManager = GridLayoutManager(requireContext(), 3)  
 rec.adapter = AdapterFill(requireContext(), MyObjTwo(answer).list)  
 Log.d(ContentValues.TAG, name + " " + img + " " + info)  
 }  
  
 **}** } else {  
 Log.e(ContentValues.TAG, "Не удалось получить данные. Пожалуйста, проверьте ваш API-ключ и запрос.")  
 }  
 }  
 }  
  
 }

class HorrorFragment : Fragment() {  
 // *TODO: Rename and change types of parameters* private var param1: String? = null  
 private var param2: String? = null  
 private lateinit var recyclerView: RecyclerView  
 var answer: MutableList<Movie> = *mutableListOf*()  
  
  
 override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
 arguments?.let **{** param1 = it.getString(*ARG\_PARAM1*)  
 param2 = it.getString(*ARG\_PARAM2*)  
 **}** }  
  
 override fun onCreateView(  
 inflater: LayoutInflater, container: ViewGroup?,  
 savedInstanceState: Bundle?  
 ): View? {  
 // Inflate the layout for this fragment  
  
 val fragmentLayout = inflater.inflate(R.layout.horror\_fragment, container, false)  
 val navController = NavHostFragment.findNavController(this)  
 val fetchMoviesTask = FetchMoviesTask(fragmentLayout)  
 fetchMoviesTask.execute()  
  
 return fragmentLayout  
  
 }  
 private inner class FetchMoviesTask(val fragmentLayout: View) : AsyncTask<Void, Void, String>() {  
 override fun doInBackground(vararg params: Void?): String? {  
 val apiKey = "43181e24"  
  
 try {  
  
 val apiUrl = "https://www.omdbapi.com/?apikey=$apiKey&s=horror&type=movie&plot=full"  
  
  
 val url = URL(apiUrl)  
 val connection = url.openConnection() as HttpURLConnection  
 val scanner = Scanner(connection.inputStream, "UTF-8").useDelimiter("\\A")  
 return if (scanner.hasNext()) scanner.next() else null  
 } catch (e: Exception) {  
 Log.e(TAG, "Ошибка при выполнении запроса: ${e.message}")  
 return null  
 }  
 }  
  
 override fun onPostExecute(result: String?) {  
 super.onPostExecute(result)  
  
  
 if (result != null) {  
 val gson = Gson()  
 val data = gson.fromJson(result, Map::class.java)  
  
  
 if (data.containsKey("Search")) {  
 val top5HorrorMovies = (data["Search"] as List<Map<String, String>>).take(5)  
 top5HorrorMovies.forEach **{** movie **->** val img = "${movie["Poster"]}"  
 val name = "${movie["Title"]}"  
 val info = "Год создания: ${movie["Year"]}"  
 val wr: Movie = Movie( name, img, info)  
 answer.add(wr)  
 if(answer.size == 5){  
 val rec: RecyclerView = fragmentLayout.findViewById(R.id.recyclerView2)  
 rec.layoutManager = GridLayoutManager(requireContext(), 3)  
 rec.adapter = AdapterFill(requireContext(), MyObjTwo(answer).list)  
 Log.d(TAG, name + " " + img + " " + info)  
 }  
  
 **}** } else {  
 Log.e(TAG, "Ошибка")  
 }  
 }  
 }  
  
 }

class MainActivity : Activity() {  
 override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
 setContentView(R.layout.*activity\_main*)  
 val timer=object :CountDownTimer(3000,1000){  
 override fun onTick(maillisUntilFinished: Long) {  
  
 }  
  
 override fun onFinish() {  
 val intent = Intent(this@MainActivity, SignInActivity::class.java)  
 startActivity(intent)  
 finish()  
 }  
  
 }  
 timer.start()  
 }

data class Movie(  
 val name: String,  
 val img: String,  
 val info: String  
)

class MyObjTwo(private val answer: List<Movie>){ val list = arrayListOf(  
 Movie(answer[0].name, answer[0].img, answer[0].info),  
 Movie(answer[1].name, answer[1].img, answer[1].info),  
 Movie(answer[2].name, answer[2].img, answer[2].info),  
 Movie(answer[3].name, answer[3].img, answer[3].info),  
 Movie(answer[4].name, answer[4].img, answer[4].info),  
 )}

class QuestRecycler(private val context: Context, private val list: ArrayList<Quests>) : RecyclerView.Adapter<QuestRecycler.MyVH>() {  
 override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): MyVH {  
 val root = LayoutInflater.from(context).inflate(R.layout.item, parent, false)  
 return MyVH(root)  
 }  
  
 inner class MyVH(itemView: View) : RecyclerView.ViewHolder(itemView) {  
 val imageView: ImageView = itemView.findViewById(R.id.questsImageView)  
 val title: TextView = itemView.findViewById(R.id.titleQuestsTextView)  
 val description: TextView = itemView.findViewById(R.id.descriptionTextView)  
 }  
  
 override fun onBindViewHolder(holder: MyVH, position: Int) {  
 holder.imageView.setImageResource(list[position].image)  
 holder.title.setText(list[position].title)  
 holder.description.setText(list[position].text)  
 }  
  
 override fun getItemCount(): Int{  
 return list.size  
 }

data class Quests(val image:Int, val title:String, val text:String)  
class MyObj{ val list = arrayListOf(  
 Quests(R.drawable.*kazan*, "Welcome to Kazan", "Now you must go to Kazan Expo and find next key. And there are many more interesting information."),  
 Quests(R.drawable.*kazan*, "Welcome to Kazan", "Now you must go to Kazan Expo and find next key. And there are many more interesting information."),  
 Quests(R.drawable.*kazan*, "Welcome to Kazan", "Now you must go to Kazan Expo and find next key. And there are many more interesting information."),  
 Quests(R.drawable.*kazan*, "Welcome to Kazan", "Now you must go to Kazan Expo and find next key. And there are many more interesting information."),  
 Quests(R.drawable.*kazan*, "Welcome to Kazan", "Now you must go to Kazan Expo and find next key. And there are many more interesting information."),  
 Quests(R.drawable.*kazan*, "Welcome to Kazan", "Now you must go to Kazan Expo and find next key. And there are many more interesting information."),  
)  
}

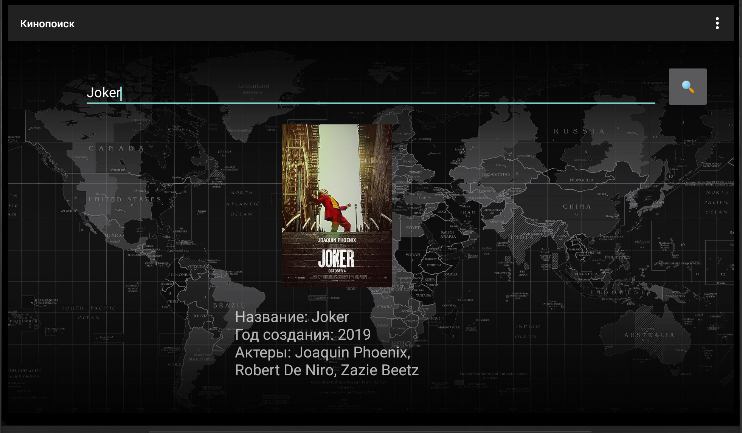
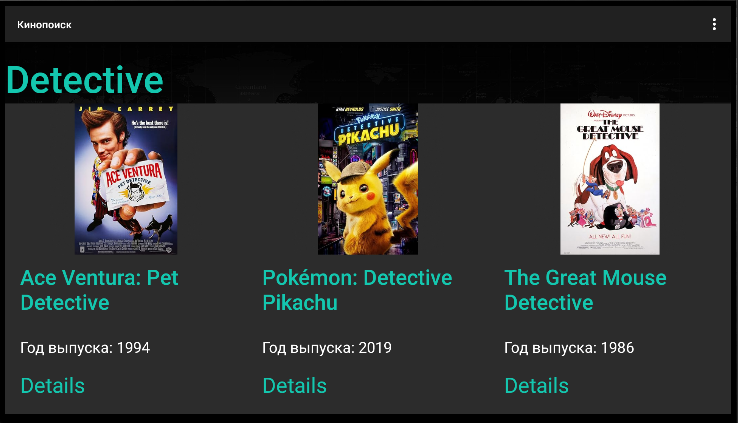
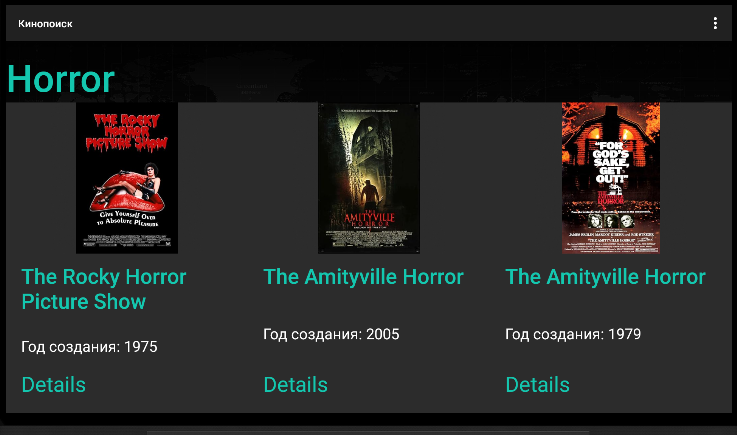
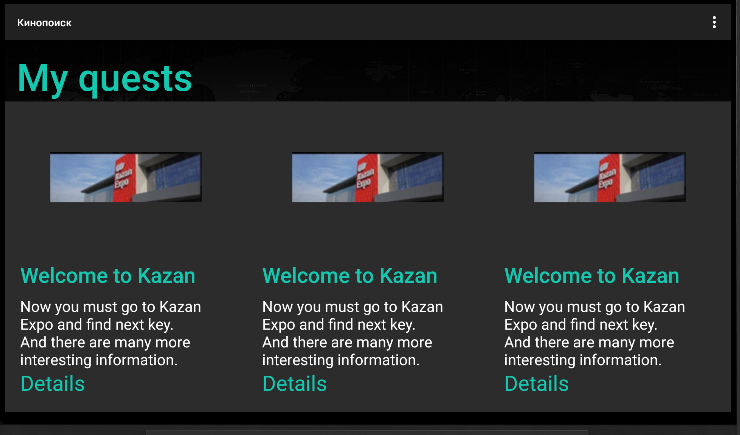
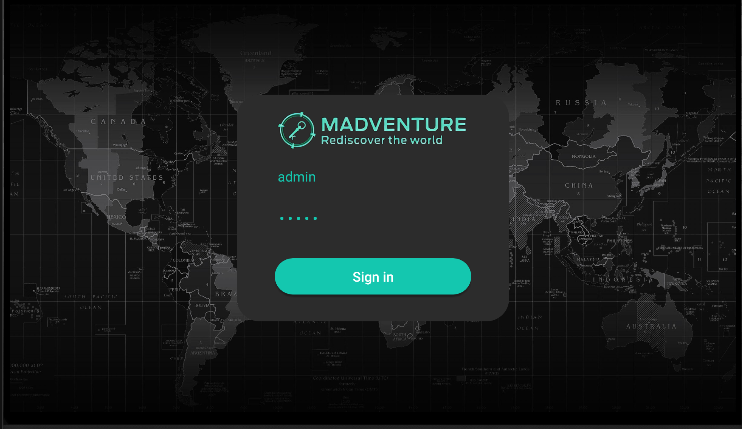
class QuestsActivity : AppCompatActivity() {  
  
 private lateinit var navController: NavController  
  
 override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
 setContentView(R.layout.*activity\_quests*)  
  
 val host: NavHostFragment = supportFragmentManager  
 .findFragmentById(R.id.*fragmentGlav*) as NavHostFragment? ?: return  
 navController = host.navController  
  
  
   
  
 }  
  
  
 override fun onCreateOptionsMenu(menu: Menu): Boolean {  
 menuInflater.inflate(R.menu.*menu*, menu)  
 return true  
 }  
  
  
  
 override fun onOptionsItemSelected(item: MenuItem): Boolean {  
  
 return item.onNavDestinationSelected(navController)  
 || super.onOptionsItemSelected(item)  
 }

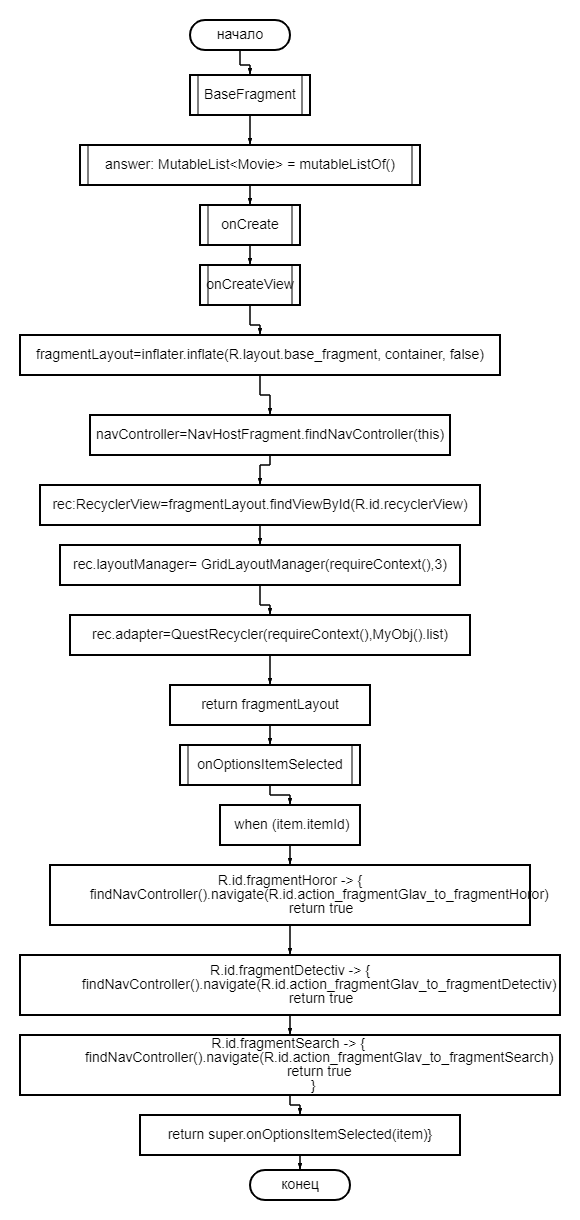
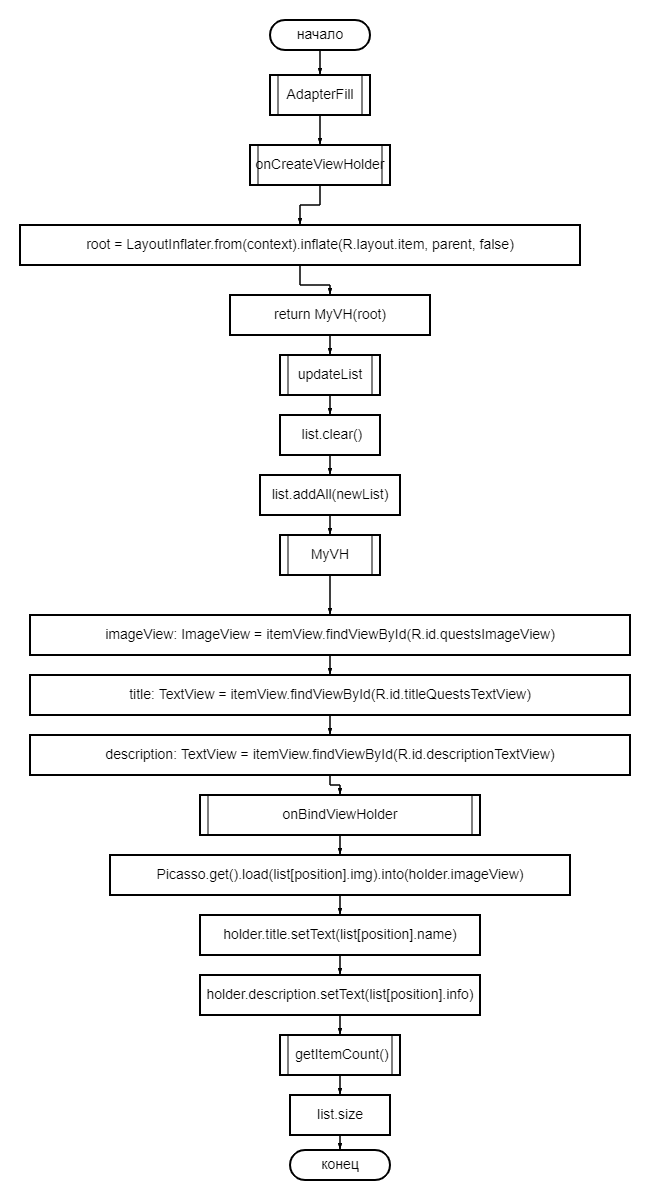
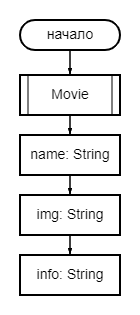
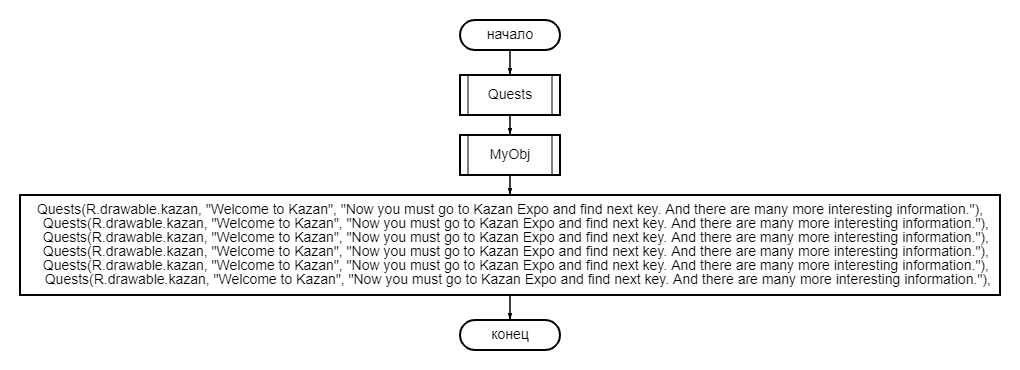
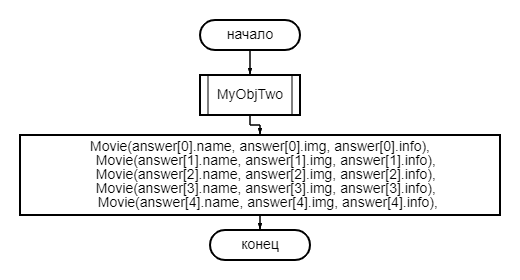
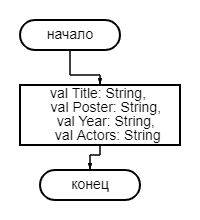
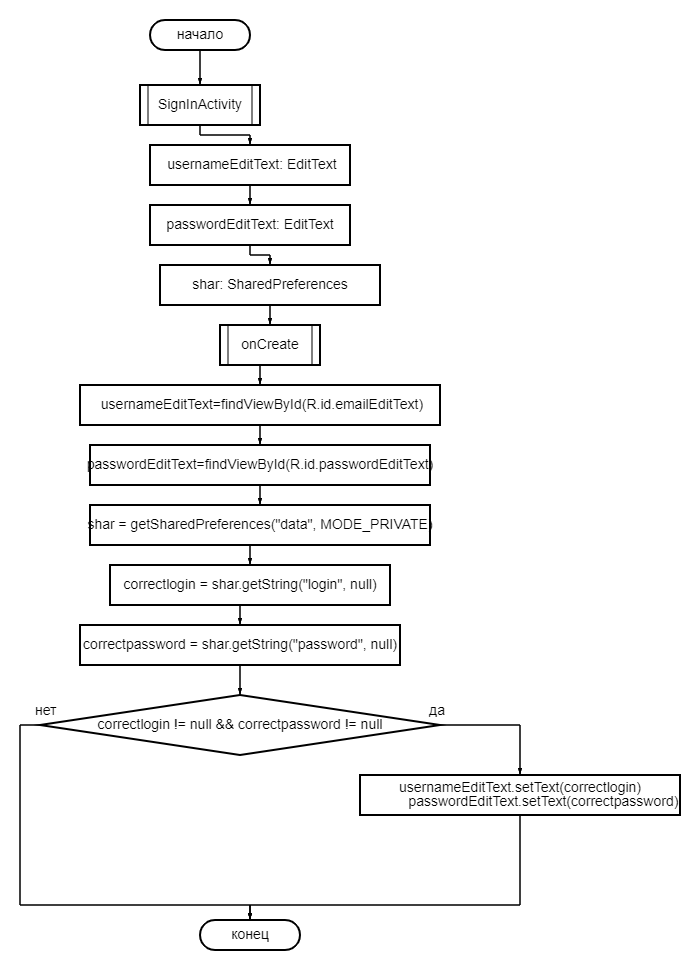
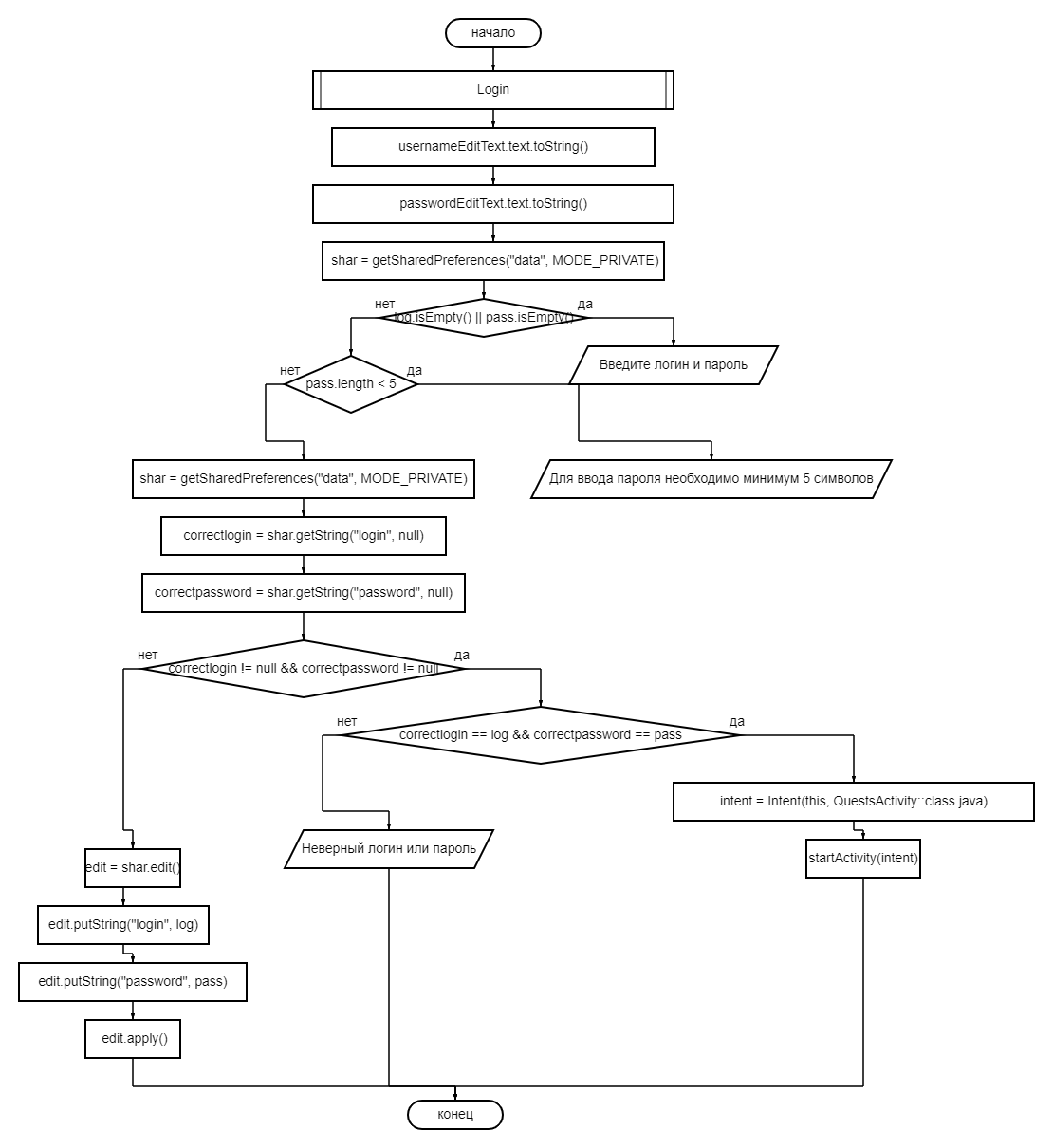
data class Search (  
 val Title: String,  
 val Poster: String,  
 val Year: String,  
 val Actors: String  
  
)

class SearchFragment : Fragment() {  
 // TODO: Rename and change types of parameters  
 private var param1: String? = null  
 private var param2: String? = null  
 private val apiKey = "43181e24"  
 private val api: Api  
 private lateinit var editTxt: EditText  
 private lateinit var img: ImageView  
 private lateinit var writeTxt: TextView  
 private lateinit var button : Button  
  
 override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
 arguments?.let **{** param1 = it.getString(ARG\_PARAM1)  
 param2 = it.getString(ARG\_PARAM2)  
 **}** }  
  
 @SuppressLint("MissingInflatedId")  
 override fun onCreateView(  
 inflater: LayoutInflater, container: ViewGroup?,  
 savedInstanceState: Bundle?  
 ): View? {  
 // Inflate the layout for this fragment  
 val fragmentLayout = inflater.inflate(R.layout.search\_fragment, container, false)  
 val navController = NavHostFragment.findNavController(this)  
 editTxt = fragmentLayout.findViewById(R.id.seachText)  
 img = fragmentLayout.findViewById(R.id.setImg)  
 writeTxt = fragmentLayout.findViewById(R.id.setTxt)  
 button = fragmentLayout.findViewById(R.id.buttonSerch)  
 button.setOnClickListener **{** val searchTxt = editTxt.text  
 if(searchTxt.isNotEmpty()){  
 output(searchTxt.toString(), img, writeTxt)  
  
 }  
 else{  
 Toast.makeText(requireContext(), "Проверьте ввод", Toast.LENGTH\_SHORT).show()  
 }  
 **}** return fragmentLayout  
 }  
 init {  
 val retrofit = Retrofit.Builder()  
 .baseUrl("https://www.omdbapi.com/")  
 .addConverterFactory(GsonConverterFactory.create())  
 .build()  
  
 api = retrofit.create(Api::class.java)  
 }  
  
 private fun output(nameFilm: String, img: ImageView, txt: TextView) {  
 val call = api.getMovieDetails(apiKey, nameFilm)  
  
 call.enqueue(object : Callback<Search> {  
 override fun onResponse(call: Call<Search>, response: Response<Search>) {  
 if (response.isSuccessful) {  
 val movieSearch = response.body()  
  
 if (movieSearch != null) {  
 val title = movieSearch.Title  
 val posterUrl = movieSearch.Poster  
 val year = movieSearch.Year  
 val actors = movieSearch.Actors  
 Log.d("MovieDetails", "$response")  
 Log.d("MovieDetails", "Title: $title, Poster URL: $posterUrl")  
  
 Picasso.get().load(posterUrl).into(img)  
  
 if (title != null) {  
 txt.text = "Название: $title \nГод создания: $year\nАктеры: $actors"  
 } else {  
 Toast.makeText(requireContext(), "Некорректное название фильма", Toast.LENGTH\_SHORT).show()  
 return // Дополнительная обработка в случае некорректного названия фильма  
 }  
 } else {  
 Toast.makeText(requireContext(), "Фильм не найден", Toast.LENGTH\_SHORT).show()  
 }  
 } else {  
 Toast.makeText(requireContext(), "Проверьте ввод", Toast.LENGTH\_SHORT).show()  
 }  
 }  
  
 override fun onFailure(call: Call<Search>, t: Throwable) {  
 // Обработка ошибок  
 Log.e("MovieDetails", "Failed to retrieve movie details", t)  
 }  
 })  
 }

class SignInActivity : Activity() {  
 private lateinit var usernameEditText: EditText  
 private lateinit var passwordEditText: EditText  
 lateinit var shar: SharedPreferences  
  
 override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
 setContentView(R.layout.activity\_sign\_in)  
 usernameEditText=findViewById(R.id.emailEditText)  
 passwordEditText=findViewById(R.id.passwordEditText)  
 shar = getSharedPreferences("data", MODE\_PRIVATE)  
 var correctlogin = shar.getString("login", null)  
 var correctpassword = shar.getString("password", null)  
 if (correctlogin != null && correctpassword != null)  
 {  
 usernameEditText.setText(correctlogin)  
 passwordEditText.setText(correctpassword)  
 }  
 }  
 fun Login(view: View){  
  
  
 var log = usernameEditText.text.toString()  
 var pass = passwordEditText.text.toString()  
 shar = getSharedPreferences("data", MODE\_PRIVATE)  
 if (log.isEmpty() || pass.isEmpty())  
 {  
 Toast.makeText(this, "Введите логин и пароль", Toast.LENGTH\_SHORT).show()  
 }  
 else if (pass.length < 5)  
 {  
 Toast.makeText(this, "Для ввода пароля необходимо минимум 5 символов", Toast.LENGTH\_SHORT).show()  
 }  
 else {  
 shar = getSharedPreferences("data", MODE\_PRIVATE)  
 var correctlogin = shar.getString("login", null)  
 var correctpassword = shar.getString("password", null)  
 if (correctlogin != null && correctpassword != null) {  
 if (correctlogin == log && correctpassword == pass) {  
 val intent = Intent(this, QuestsActivity::class.java)  
 startActivity(intent)  
 } else {  
 Toast.makeText(this, "Неверный логин или пароль", Toast.LENGTH\_SHORT).show()  
 }  
 } else {  
 var edit = shar.edit()  
 edit.putString("login", log)  
 edit.putString("password", pass)  
 edit.apply()  
 }

Приложение:



Блок-схема:  
  




**Тестовые ситуации:**

**1)** Проверка на пустоту (if else)  
**2)** Проверка (try catch)  
**3)** Проверка на правильность ввода(if else)

**Вывод:** При выполнении данной практической работы я научился работать с ANDROID TV.