Министерство образования и молодежной политики Свердловской области



ГАПОУ СО «Екатеринбургский колледж транспортного строительства»

Отчёт по программе «**Практическое занятие 22\_2**»

Выполнил: Юнусов Руслан Александрович

Группа: ПР-33

Преподаватель: Мирошниченко Г.В

2024

**Задание:**

Разработать моб. Приложение, информация в приложении должна быть запрошена через API и сохраняться в SharedPreferences, в формате JSON. Приложение должно содержать:

* Экран с авторизацией
* Экран с вводом данных о персонаже (его имя или id). И в нем же сохранять в базу данных.

**Входные и выходные данные**

Pref – SharedPreferences, поле класса для сохранения данных

Login – editText, Вводимое текстовое поле, ввод логина

Password – editText, Вводимое текстовое поле, ввод пароля

Signin – Button, кнопка, Кнопка входа

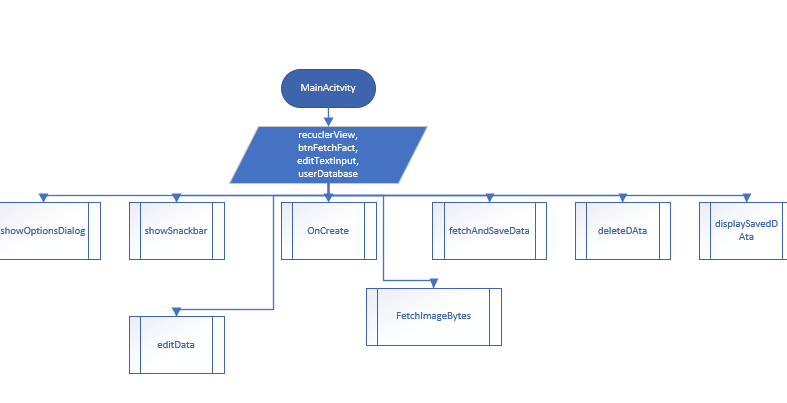
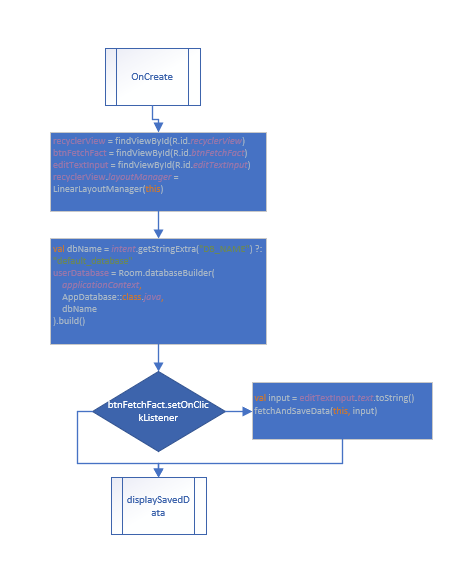
userDatabase – AppDatabase, класс, класс базы данных авторизированных пользователей

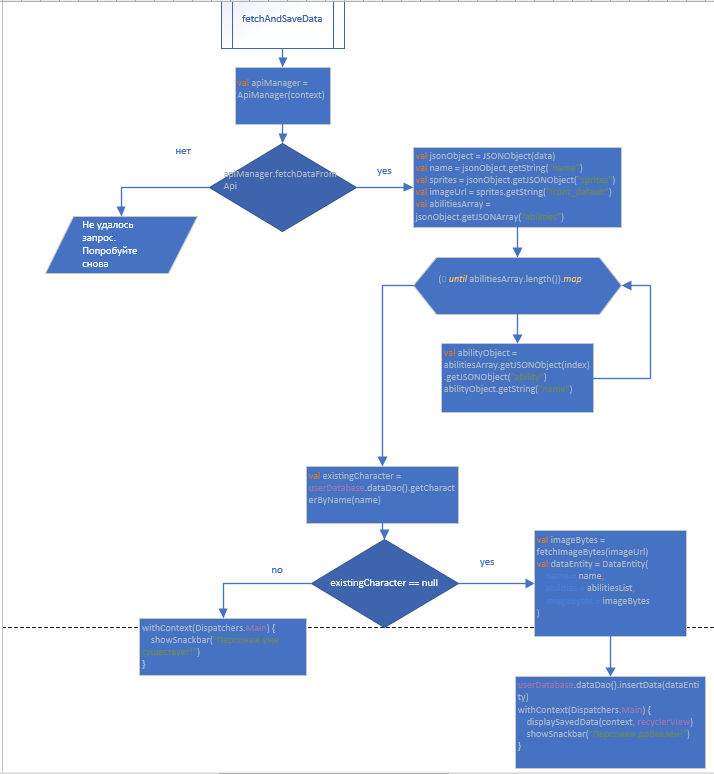
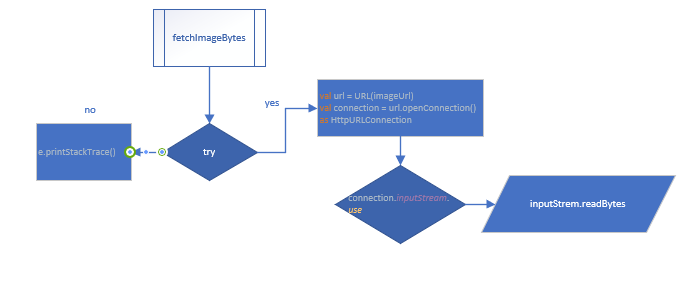
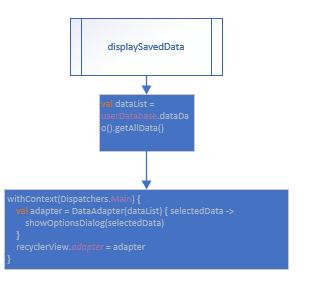
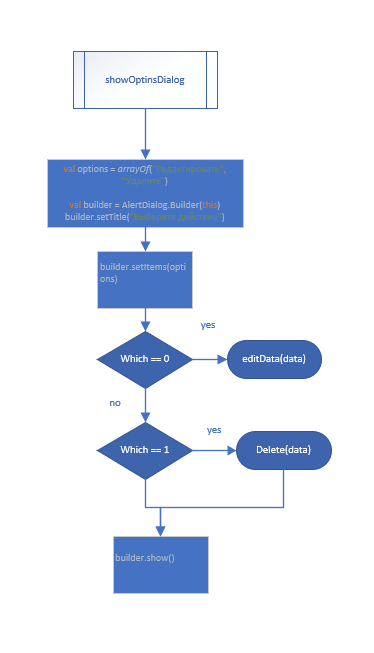
recyclerview – RecyclerView, макет для вывода всех персонажей в базе данных

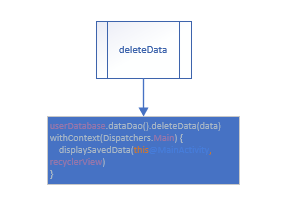
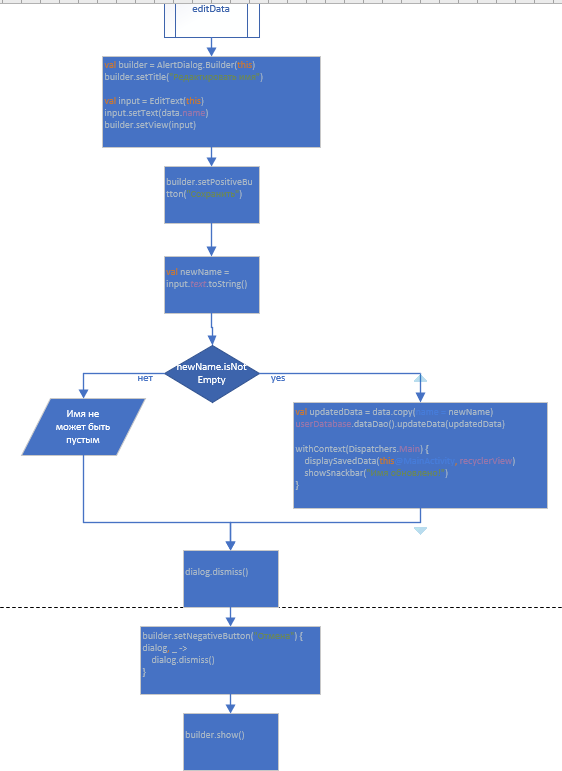
binFetchFact – Button, кнопка, кнопка добавления персонажа

editTextInput – EditText, вводимое текстовое поле, ввод имя персонажа

**Блок-схемы**

**** ****  

**** ****

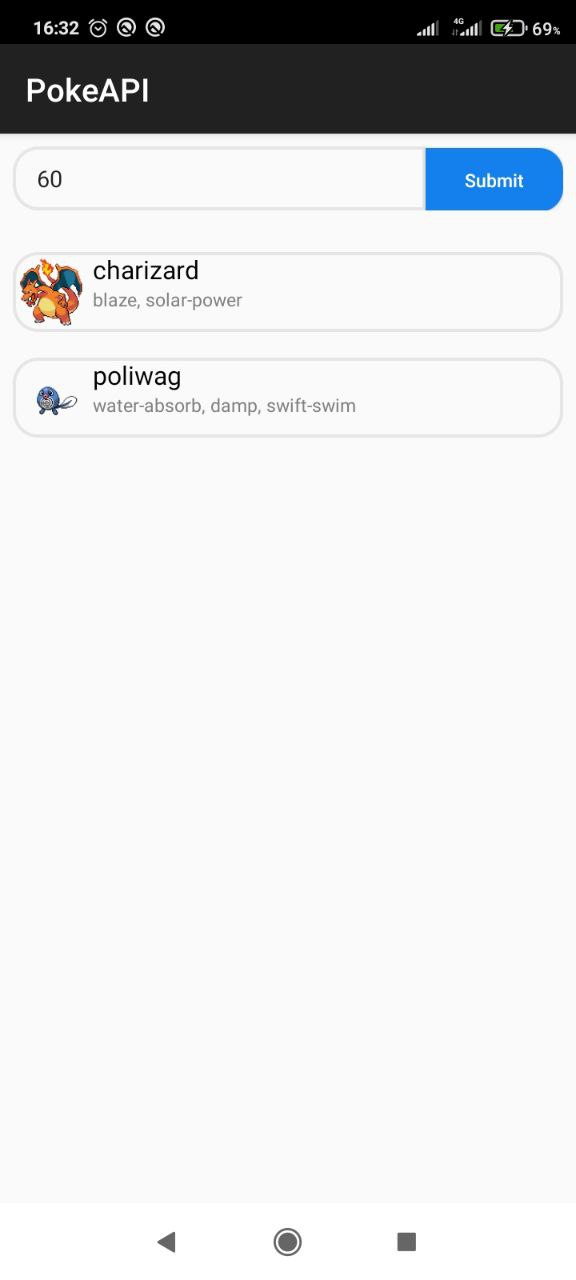
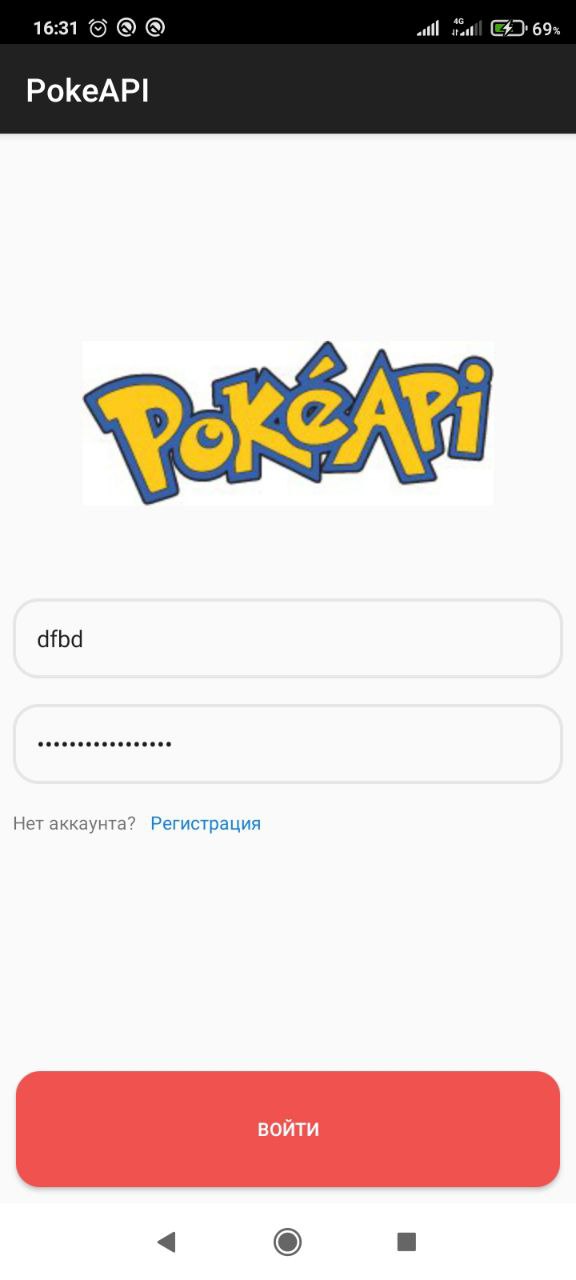
**Листинг программы (если есть)**

package com.example.prakt22\_2  
  
import android.content.Context  
import android.os.Bundle  
import android.widget.Button  
import android.widget.EditText  
import androidx.appcompat.app.AppCompatActivity  
import androidx.recyclerview.widget.LinearLayoutManager  
import androidx.recyclerview.widget.RecyclerView  
import com.example.prakt22\_2.API.ApiManager  
import com.example.prakt22\_2.Data.AppDatabase  
import com.example.prakt22\_2.Data.DataAdapter  
import com.example.prakt22\_2.Data.DataEntity  
import kotlinx.coroutines.CoroutineScope  
import kotlinx.coroutines.Dispatchers  
import kotlinx.coroutines.launch  
import kotlinx.coroutines.withContext  
import org.json.JSONObject  
import androidx.appcompat.app.AlertDialog  
import androidx.room.Room  
import com.google.android.material.snackbar.Snackbar  
import java.net.HttpURLConnection  
import java.net.URL  
  
class MainActivity : AppCompatActivity() {  
 private lateinit var recyclerView: RecyclerView  
 private lateinit var btnFetchFact: Button  
 private lateinit var editTextInput: EditText  
 private lateinit var userDatabase: AppDatabase  
  
 override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
 setContentView(R.layout.*activity\_main*)  
  
 recyclerView = findViewById(R.id.*recyclerView*)  
 btnFetchFact = findViewById(R.id.*btnFetchFact*)  
 editTextInput = findViewById(R.id.*editTextInput*)  
  
 recyclerView.*layoutManager* = LinearLayoutManager(this)  
  
 val dbName = *intent*.getStringExtra("DB\_NAME") ?: "default\_database"  
 userDatabase = Room.databaseBuilder(  
 *applicationContext*,  
 AppDatabase::class.*java*,  
 dbName  
 ).build()  
  
 btnFetchFact.setOnClickListener **{** val input = editTextInput.*text*.toString()  
 fetchAndSaveData(this, input)  
 **}** displaySavedData(this, recyclerView)  
 }  
  
 private fun fetchAndSaveData(context: Context, input: String) {  
 val apiManager = ApiManager(context)  
  
 apiManager.fetchDataFromApi(input,  
 **{** data **->** val jsonObject = JSONObject(data)  
 val name = jsonObject.getString("name")  
 val sprites = jsonObject.getJSONObject("sprites")  
 val imageUrl = sprites.getString("front\_default")  
  
 val abilitiesArray = jsonObject.getJSONArray("abilities")  
 val abilitiesList = (0 *until* abilitiesArray.length()).*map* **{** index **->** val abilityObject = abilitiesArray.getJSONObject(index).getJSONObject("ability")  
 abilityObject.getString("name")  
 **}**.*toTypedArray*()  
  
 *CoroutineScope*(Dispatchers.IO).*launch* **{** val existingCharacter = userDatabase.dataDao().getCharacterByName(name)  
 if (existingCharacter == null) {  
 val imageBytes = fetchImageBytes(imageUrl)  
 val dataEntity = DataEntity(  
 name = name,  
 abilities = abilitiesList,  
 imageBytes = imageBytes  
 )  
 userDatabase.dataDao().insertData(dataEntity)  
 withContext(Dispatchers.Main) **{** displaySavedData(context, recyclerView)  
 showSnackbar("Персонаж добавлен!")  
 **}** } else {  
 withContext(Dispatchers.Main) **{** showSnackbar("Персонаж уже существует!")  
 **}** }  
 **}  
 }**,  
 **{** error **->** showSnackbar("Не удалось найти запрос. Попробуйте снова.")  
 **}** )  
 }  
  
  
 private fun showSnackbar(message: String) {  
 Snackbar.make(findViewById(android.R.id.*content*), message, Snackbar.*LENGTH\_LONG*).show()  
 }  
  
 private fun fetchImageBytes(imageUrl: String): ByteArray? {  
 return try {  
 val url = URL(imageUrl)  
 val connection = url.openConnection() as HttpURLConnection  
 connection.*inputStream*.*use* **{** inputStream **->** return inputStream.*readBytes*()  
 **}** } catch (e: Exception) {  
 e.printStackTrace()  
 null  
 }  
 }  
  
 private fun displaySavedData(context: Context, recyclerView: RecyclerView) {  
 *CoroutineScope*(Dispatchers.IO).*launch* **{** val dataList = userDatabase.dataDao().getAllData()  
  
 withContext(Dispatchers.Main) **{** val adapter = DataAdapter(dataList) **{** selectedData **->** showOptionsDialog(selectedData)  
 **}** recyclerView.*adapter* = adapter  
 **}  
 }** }  
  
 private fun showOptionsDialog(data: DataEntity) {  
 val options = *arrayOf*("Редактировать", "Удалить")  
  
 val builder = AlertDialog.Builder(this)  
 builder.setTitle("Выберите действие")  
 builder.setItems(options) **{** dialog, which **->** when (which) {  
 0 -> editData(data)  
 1 -> deleteData(data)  
 }  
 **}** builder.show()  
 }  
  
 private fun deleteData(data: DataEntity) {  
 *CoroutineScope*(Dispatchers.IO).*launch* **{** userDatabase.dataDao().deleteData(data)  
 withContext(Dispatchers.Main) **{** displaySavedData(this@MainActivity, recyclerView)  
 **}  
 }** }  
  
 private fun editData(data: DataEntity) {  
 val builder = AlertDialog.Builder(this)  
 builder.setTitle("Редактировать имя")  
   
 val input = EditText(this)  
 input.setText(data.name)  
 builder.setView(input)  
  
 builder.setPositiveButton("Сохранить") **{** dialog, \_ **->** val newName = input.*text*.toString()  
 if (newName.*isNotEmpty*()) {  
 *CoroutineScope*(Dispatchers.IO).*launch* **{** val updatedData = data.copy(name = newName)  
 userDatabase.dataDao().updateData(updatedData)  
  
 withContext(Dispatchers.Main) **{** displaySavedData(this@MainActivity, recyclerView)  
 showSnackbar("Имя обновлено!")  
 **}  
 }** } else {  
 showSnackbar("Имя не может быть пустым!")  
 }  
 dialog.dismiss()  
 **}** builder.setNegativeButton("Отмена") **{** dialog, \_ **->** dialog.dismiss()  
 **}** builder.show()  
 }  
  
}

package com.example.prakt22\_2  
  
import android.content.Intent  
import android.content.SharedPreferences  
import androidx.appcompat.app.AppCompatActivity  
import android.os.Bundle  
import android.widget.Button  
import android.widget.EditText  
import android.widget.TextView  
import androidx.room.Room  
import com.example.prakt22\_2.Data.AppDatabase  
import com.google.android.material.snackbar.Snackbar  
import kotlinx.coroutines.CoroutineScope  
import kotlinx.coroutines.Dispatchers  
import kotlinx.coroutines.launch  
import kotlinx.coroutines.withContext  
  
class Login : AppCompatActivity() {  
 private lateinit var pref: SharedPreferences  
 private lateinit var login: EditText  
 private lateinit var password: EditText  
 private lateinit var signin: Button  
 private lateinit var regist: TextView  
 private lateinit var userDatabase: AppDatabase  
  
 override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
 setContentView(R.layout.*login*)  
  
 login = findViewById(R.id.*login*)  
 password = findViewById(R.id.*password*)  
 signin = findViewById(R.id.*registration*)  
 regist = findViewById(R.id.*register\_click*)  
  
 userDatabase = Room.databaseBuilder(  
 *applicationContext*,  
 AppDatabase::class.*java*,  
 "user\_database"  
 ).build()  
  
  
 regist.setOnClickListener **{** var intent = Intent(this, Registratio::class.*java*)  
 startActivity(intent)  
 **}** signin.setOnClickListener **{** if (login.*text*.*isNotEmpty*() && password.*text*.*isNotEmpty*()) {  
 val userLogin = login.*text*.toString()  
 val userPassword = password.*text*.toString()  
  
 if (userPassword.length <= 8) {  
  
 *CoroutineScope*(Dispatchers.IO).*launch* **{** val userDao = userDatabase.userDao()  
 val user = userDao.getUserByUsername(userLogin)  
  
 withContext(Dispatchers.Main) **{** if (user == null || user.password != userPassword) {  
 showSnackbar("Неверный логин или пароль")  
 } else {  
 showSnackbar("Вход выполнен")  
 val dbName = "${userLogin}\_database"  
 val intent = Intent(this@Login, MainActivity::class.*java*)  
 intent.putExtra("DB\_NAME", dbName)  
 startActivity(intent)  
  
 }  
 **}  
 }** }  
 else {  
 showSnackbar("Пароль не должен превышать 8 символов")  
 }  
 } else {  
 showSnackbar("Обнаружены пустые поля")  
 }  
 **}** }  
  
 private fun showSnackbar(message: String) {  
 Snackbar.make(findViewById(android.R.id.*content*), message, Snackbar.*LENGTH\_LONG*).show()  
 }  
}

package com.example.prakt22\_2  
  
import android.content.Intent  
import android.content.SharedPreferences  
import androidx.appcompat.app.AppCompatActivity  
import android.os.Bundle  
import android.widget.Button  
import android.widget.EditText  
import androidx.room.Room  
import com.example.prakt22\_2.Data.AppDatabase  
import com.example.prakt22\_2.Data.UserEntity  
import com.google.android.material.snackbar.Snackbar  
import kotlinx.coroutines.CoroutineScope  
import kotlinx.coroutines.Dispatchers  
import kotlinx.coroutines.launch  
import kotlinx.coroutines.withContext  
  
class Registratio : AppCompatActivity() {  
 private lateinit var pref: SharedPreferences  
 private lateinit var login: EditText  
 private lateinit var password: EditText  
 private lateinit var signin: Button  
 private lateinit var userDatabase: AppDatabase  
  
 override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
 setContentView(R.layout.*activity\_registratio*)  
 login = findViewById(R.id.*login*)  
 password = findViewById(R.id.*password*)  
 signin = findViewById(R.id.*registration*)  
  
 userDatabase = Room.databaseBuilder(  
 *applicationContext*,  
 AppDatabase::class.*java*,  
 "user\_database"  
 ).build()  
  
 signin.setOnClickListener **{** if (login.*text*.*isNotEmpty*() && password.*text*.*isNotEmpty*()) {  
 val userLogin = login.*text*.toString()  
 val userPassword = password.*text*.toString()  
  
 if (userPassword.length <= 8) {  
  
 *CoroutineScope*(Dispatchers.IO).*launch* **{** val userDao = userDatabase.userDao()  
 val existingUser = userDao.getUserByUsername(userLogin)  
  
 if (existingUser == null) {  
 val newUser = UserEntity(username = userLogin, password = userPassword)  
 userDao.insertUser(newUser)  
  
 withContext(Dispatchers.Main) **{** showSnackbar("Регистрация успешна")  
 val dbName = "${userLogin}\_database"  
 val intent = Intent(this@Registratio, MainActivity::class.*java*)  
 intent.putExtra("DB\_NAME", dbName)  
 startActivity(intent)  
 **}** } else {  
 withContext(Dispatchers.Main) **{** showSnackbar("Пользователь уже существует")  
 **}** }  
 **}** }else {  
 showSnackbar("Пароль не должен превышать 8 символов")  
 }  
 } else {  
 showSnackbar("Обнаружены пустые поля")  
 }  
 **}** }  
  
 private fun showSnackbar(message: String) {  
 Snackbar.make(findViewById(android.R.id.*content*), message, Snackbar.*LENGTH\_LONG*).show()  
 }  
}

**Скрин-шот экранов**

****

**Вывод**

Я освоился с библиотекой Room, научился работать с api