Министерство образования и молодежной политики Свердловской области



ГАПОУ СО «Екатеринбургский колледж транспортного строительства»

Отчёт по программе «Практическое занятие 22\_2

Сетевые взаимодействия - работа с API.»

Выполнил: Оберюхтина Елена

Группа: ПР-31

Преподаватель: Мирошниченко Г. В.

2023

**Цель:**

1. Научиться работать API
2. Повторить работу с форматом данных JSON и базой ROOM на языке Kotlin.

**Задание (вариант 2 – Приложение «Факты о персонаже» Poke API ( https://pokeapi.co/ ). Персонаж выбирается по его имени или id пользователем приложения.):**

Разработать мобильное приложение, информация в приложении должна быть запрошена через API и сохраняться в room.

Приложение должно содержать:

* приветственный экран с авторизацией пользователя, проверкой;
* ввода на пустоту и сохранением информации о пользователе через механизм SharedPreferences;
* содержать форму ввода данных;
* экран для вывода сохраненной информации, с использованием библиотеки ROOM, возможностью добавления, удаления и редактирования введенных данных, с сохранением в базе данных (ROOM или SQL);
* стилизация приложения в соответствии с вариантом;
* изменить название, тему и иконку приложения (должны соответствовать названию);
* предусмотреть портретную ориентацию

**Входные данные:**

* username (String) – введенный пользователем логин;
* password (String) – введенный пользователем пароль;
* pocemon (String) – введенный пользователем цифра для факта.

**Выходные данные:**

* InfoSet, в котором отображаются вся информация;
* Сообщения пользователю об ошибках.

**Листинг программы:**

Dao

package com.example.pract22\_2  
  
import androidx.room.\*  
import androidx.room.Dao  
import kotlinx.coroutines.flow.Flow  
  
@Dao  
interface Dao {  
 //запись  
 @Insert  
 fun insertItem(item: Item)  
  
 //получение всей бд  
 @Query("SELECT \* FROM items")  
 fun getAllItem(): Flow<List<Item>>  
 //удаление  
 @Delete  
 suspend fun deleteItem(item: Item)  
 //обновление  
 @Update  
 suspend fun updateItem(item: Item)  
}

Item

package com.example.pract22\_2  
import androidx.room.ColumnInfo  
import androidx.room.Entity  
import androidx.room.PrimaryKey  
  
@Entity (tableName = "items")  
  
//коллоны  
data class Item(  
  
 //ид  
 @PrimaryKey(autoGenerate = true)  
 var id: Int? = null,  
 //имя  
 @ColumnInfo(name = "name")  
 var name: String,  
 //рост  
 @ColumnInfo(name = "height")  
 var height: String,  
 //вес  
 @ColumnInfo(name = "weight ")  
 var weight: String,  
 //способность  
 @ColumnInfo(name = "ability")  
 var abiliti: String,  
  
 )

ItemAdapter

package com.example.pract22\_2  
  
import android.content.Context  
import android.view.LayoutInflater  
import android.view.View  
import android.view.ViewGroup  
import android.widget.BaseAdapter  
import android.widget.ImageButton  
import android.widget.TextView  
  
class ItemAdapter(  
 context: Context,  
 private val itemList: List<Item>,  
 private val onDeleteClick: (Item) -> Unit,  
 private val onEditClick: (Item) -> Unit  
) : BaseAdapter() {  
 private val inflater: LayoutInflater = LayoutInflater.from(context)  
  
 override fun getCount(): Int {  
 return itemList.size  
 }  
  
 override fun getItem(position: Int): Any {  
 return itemList[position]  
 }  
  
 override fun getItemId(position: Int): Long {  
 return position.toLong()  
 }  
  
 override fun getView(position: Int, convertView: View?, parent: ViewGroup?): View {  
 val view: View  
 val holder: ViewHolder  
  
 if (convertView == null) {  
 view = inflater.inflate(R.layout.*activity\_item\_list\_row*, parent, false)  
 holder = ViewHolder(  
 view.findViewById(R.id.*titleTextView*),  
 view.findViewById(R.id.*descriptionTextView*),  
 view.findViewById(R.id.*editButton*),  
 view.findViewById(R.id.*deleteButton*),  
 view.findViewById(R.id.*weightText*),  
 view.findViewById(R.id.*abilityText*)  
 )  
 view.*tag* = holder  
 } else {  
 view = convertView  
 holder = view.*tag* as ViewHolder  
 }  
  
 val item = itemList[position]  
 holder.titleTextView.*text* = item.name  
 holder.descriptionTextView.*text* = item.height  
 holder.weightText.*text* = item.weight  
 holder.abilityText.*text* = item.abiliti  
  
 // Обработка нажатий на кнопку "Редактировать"  
 holder.editButton.setOnClickListener **{** onEditClick(item)  
 **}** // Обработка нажатий на кнопку "Удалить"  
 holder.deleteButton.setOnClickListener **{** onDeleteClick(item)  
 **}** return view  
 }  
  
 private class ViewHolder(  
 val titleTextView: TextView,  
 val descriptionTextView: TextView,  
 val editButton: ImageButton,  
 val deleteButton: ImageButton,  
 val weightText: TextView,  
 val abilityText: TextView  
 )  
}

Item\_list\_row

package com.example.pract22\_2  
  
import androidx.appcompat.app.AppCompatActivity  
import android.os.Bundle  
  
class item\_list\_row : AppCompatActivity() {  
 override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
 setContentView(R.layout.*activity\_item\_list\_row*)  
 }  
}

MainActivity

package com.example.pract22\_2  
  
import android.content.Context  
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.Toast  
import androidx.appcompat.app.AlertDialog  
  
class MainActivity : AppCompatActivity() {  
 private lateinit var usernameEditText: EditText  
 private lateinit var passwordEditText: EditText  
 private lateinit var sharedPreferences: SharedPreferences  
 override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
 setContentView(R.layout.*activity\_main*)  
 val registerButton = findViewById<Button>(R.id.*registrbut*)  
 registerButton.setOnClickListener **{** val intent = Intent(this, RegistrActivity::class.*java*)  
 startActivity(intent)  
 finish()  
 **}** usernameEditText = findViewById(R.id.*login*)  
 passwordEditText = findViewById(R.id.*pass*)  
 val loginButton: Button = findViewById(R.id.*enterbut*)  
 sharedPreferences = getSharedPreferences("my\_app\_prefs", Context.*MODE\_PRIVATE*)  
 val savedUsername = sharedPreferences.getString("login", "")  
 val savedPassword = sharedPreferences.getString("password", "")  
 // установка сохраненного имя пользователя в EditText  
 usernameEditText.setText(savedUsername)  
  
 loginButton.setOnClickListener **{** val username = usernameEditText.*text*.toString().*trim*()  
 val password = passwordEditText.*text*.toString().*trim*()  
  
 if (username.*isEmpty*()) {  
 Toast.makeText(this@MainActivity, "Введите логин", Toast.*LENGTH\_SHORT*).show()  
 if (password.*isEmpty*()) {  
 Toast.makeText(this@MainActivity, "Введите пароль", Toast.*LENGTH\_SHORT*).show()  
 }  
 }  
 if (password.*isEmpty*()) {  
 Toast.makeText(this@MainActivity, "Введите пароль", Toast.*LENGTH\_SHORT*).show()  
 }  
  
 if (savedUsername == username && savedPassword == password) {  
 val intent = Intent(this, PokemonActivity::class.*java*)  
 intent.putExtra("saved\_username", username)  
 startActivity(intent)  
 finish()  
 } else {  
 val alertDialogBuilder = AlertDialog.Builder(this)  
 alertDialogBuilder.setTitle("Ошибка")  
 alertDialogBuilder.setMessage("Неверный логин или пароль.")  
 alertDialogBuilder.setPositiveButton("OK") **{** \_, \_ **-> }** val alertDialog = alertDialogBuilder.create()  
 alertDialog.show()  
 }  
 **}** }  
}

MainDb

package com.example.pract22\_2  
  
import android.content.Context  
import androidx.room.Database  
import androidx.room.Room  
import androidx.room.RoomDatabase  
  
  
@Database(entities = [Item::class], version = 1)  
abstract class MainDb : RoomDatabase() {  
  
 abstract fun getDao(): Dao  
  
 companion object{  
 fun getDb(context: Context): MainDb{  
 return Room.databaseBuilder(  
 context.*applicationContext*, MainDb::class.*java*, "test.db"  
 ).build()  
 }  
 }  
}

PokemonActivity

package com.example.pract22\_2  
  
import android.content.Intent  
import androidx.appcompat.app.AppCompatActivity  
import android.os.Bundle  
import android.util.Log  
import android.view.View  
import android.widget.EditText  
import android.widget.TextView  
import android.widget.Toast  
import androidx.appcompat.app.AlertDialog  
import com.android.volley.Request  
import com.android.volley.toolbox.StringRequest  
import com.android.volley.toolbox.Volley  
import org.json.JSONException  
import org.json.JSONObject  
  
class PokemonActivity : AppCompatActivity() {  
 override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
 setContentView(R.layout.*activity\_pokemon*)  
 }  
 private fun getResualt(poce:String)  
 {  
 if (poce.*isNotEmpty*()) {  
 val url = "https://pokeapi.co/api/v2/pokemon/${poce}/"  
  
 val queue = Volley.newRequestQueue(this)  
  
 val stringRequest = StringRequest(  
 Request.Method.*GET*, url,  
 **{** response **->** try {  
 val jsonObject = JSONObject(response)  
 val name = jsonObject.getString("name")  
 val height = jsonObject.getInt("height")  
 val weight = jsonObject.getInt("weight")  
 val abilities = jsonObject.getJSONArray("abilities")  
 val abilityList = *mutableListOf*<String>()  
 for (i in 0 *until* abilities.length()) {  
 val abilityObject = abilities.getJSONObject(i)  
 val ability = abilityObject.getJSONObject("ability")  
 val abilityName = ability.getString("name")  
 abilityList.add(abilityName)  
 }  
 Log.d("MyLog", "Name: $name")  
 Log.d("MyLog", "Height: $height")  
 Log.d("MyLog", "Weight: $weight")  
 Log.d("MyLog", "Abilities: ${abilityList.*joinToString*(", ")}")  
  
 val textInfoSet: TextView = findViewById(R.id.*textinfo*)  
 textInfoSet.*text* =  
 "Name: $name\nHeight: $height\nWeight: $weight\nAbilities: ${  
 abilityList.*joinToString*(", ")  
 }"  
 //запись в бд  
 val db = MainDb.getDb(this)  
 val item = Item(  
 null,  
 name,  
 height.toString(),  
 weight.toString(),  
 abilityList.*joinToString*()  
 )  
 Thread **{** db.getDao().insertItem(item)  
 **}**.start()  
  
 } catch (e: JSONException) {  
 Log.e("MyLog", "JSON Parsing error: ${e.message}")  
 }  
 **}**,  
 **{** error **->** Log.d("MyLog", "Volley error: ${error.message}")  
 **}**)  
  
 queue.add(stringRequest)  
 }  
 else{ Toast.makeText(this@PokemonActivity, "Заполните поле", Toast.*LENGTH\_SHORT*).show()  
 }  
 }  
 fun getResultBut(view: View) {  
 val findpocemon: EditText =findViewById(R.id.*editText*)  
 val pocemon = findpocemon.*text*.toString()  
  
 getResualt(pocemon)  
 }  
 fun perehod(view: View) {  
 val intent = Intent(this, ScreenStatistics::class.*java*)  
 startActivity(intent)  
 }  
}

RegistrActivity

package com.example.pract22\_2  
  
import android.app.AlertDialog  
import android.content.Context  
import android.content.Intent  
import android.content.SharedPreferences  
import androidx.appcompat.app.AppCompatActivity  
import android.os.Bundle  
import android.view.View  
import android.widget.Button  
import android.widget.EditText  
import android.widget.Toast  
  
class RegistrActivity : AppCompatActivity() {  
  
 private lateinit var sharedPreferences: SharedPreferences  
 override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
 setContentView(R.layout.*activity\_registr*)  
 val toRegisterButton = findViewById<Button>(R.id.*registrbutone*)  
 sharedPreferences = getSharedPreferences("my\_app\_prefs", Context.*MODE\_PRIVATE*)  
 toRegisterButton.setOnClickListener **{** val loginTextView = findViewById<EditText>(R.id.*login1*)  
 val passwordTextView = findViewById<EditText>(R.id.*pass1*)  
  
 val username = loginTextView.*text*.toString().*trim*()  
 val password = passwordTextView.*text*.toString().*trim*()  
  
 if (username.*isEmpty*()) {  
 Toast.makeText(this@RegistrActivity, "Введите логин", Toast.*LENGTH\_SHORT*).show()  
 if (password.*isEmpty*()) {  
 Toast.makeText(this@RegistrActivity, "Введите пароль", Toast.*LENGTH\_SHORT*).show()  
 }  
 }  
 if (password.*isEmpty*()) {  
 Toast.makeText(this@RegistrActivity, "Введите пароль", Toast.*LENGTH\_SHORT*).show()  
 }  
 else{ // сохранение имя пользователя и пароля в SharedPreferences  
 sharedPreferences.edit().putString("login", username).apply()  
 sharedPreferences.edit().putString("password", password).apply()  
  
 val alertDialogBuilder = AlertDialog.Builder(this)  
 alertDialogBuilder.setTitle("Регистрация завершена")  
 alertDialogBuilder.setMessage("Вы успешно зарегистрировались.")  
 alertDialogBuilder.setPositiveButton("OK") **{** \_, \_ **->** val intent = Intent(this, MainActivity::class.*java*)  
 startActivity(intent)  
 finish()  
 **}** val alertDialog = alertDialogBuilder.create()  
 alertDialog.show()}  
  
 **}** }  
  
 fun btback(view: View) {  
 val intent = Intent(this, MainActivity::class.*java*)  
 startActivity(intent)  
  
 }

ScreenStatistics

package com.example.pract22\_2  
  
import android.app.AlertDialog  
import android.content.Intent  
import androidx.appcompat.app.AppCompatActivity  
import android.os.Bundle  
import android.view.View  
import android.widget.EditText  
import android.widget.LinearLayout  
import android.widget.ListView  
import androidx.lifecycle.asLiveData  
import kotlinx.coroutines.CoroutineScope  
import kotlinx.coroutines.Dispatchers  
import kotlinx.coroutines.launch  
import kotlinx.coroutines.withContext  
  
class ScreenStatistics : AppCompatActivity() {  
  
 private val dataList = *mutableListOf*<Item>()  
  
 override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
 setContentView(R.layout.*activity\_screen\_statistics*)  
 printInfo()  
 }  
 private fun printInfo() {  
 val textList = findViewById<ListView>(R.id.*listView*)  
 dataList.clear()  
 val db = MainDb.getDb(this)  
 db.getDao().getAllItem().*asLiveData*().observe(this) **{** list **->** val updatedDataList = *mutableListOf*<Item>()  
 list.*forEach* **{** item **->** val newItem = Item(id = item.id, name = item.name, height = item.height, weight = item.weight, abiliti = item.abiliti)  
 updatedDataList.add(newItem)  
 **}** // После завершения цикла, обновите dataList  
 dataList.clear()  
 dataList.addAll(updatedDataList)  
 val itemAdapter = ItemAdapter(this, dataList,  
 onDeleteClick = **{** itemToDelete **->** // Здесь выполняйте удаление элемента из базы данных и обновление списка  
 val db = MainDb.getDb(this)  
 *CoroutineScope*(Dispatchers.IO).*launch* **{** try {  
 db.getDao().deleteItem(itemToDelete)  
  
 printInfo()  
 } catch (e: Exception) {  
 }  
 **}** printInfo()  
 printInfo()  
 **}**,  
 onEditClick = **{** position **->** val db = MainDb.getDb(this)  
 var bdNewText: Array<String> = *arrayOf*("id", "name", "height", "weight", "ability")  
 *CoroutineScope*(Dispatchers.IO).*launch* **{** try {  
 bdNewText= *arrayOf*(position.id.*toString*(), position.name, position.height, position.weight, position.abiliti)  
 } catch (e: Exception) {  
 }  
 **}** //изменения через AlertDiolog  
 val builder = AlertDialog.Builder(this)  
 builder.setTitle("Измените данные")  
 // Создайте контейнер LinearLayout для размещения EditText  
 val layout = LinearLayout(this)  
 val EditNumber = EditText(this)  
 val EditFakt = EditText(this)  
 val EditWeight = EditText(this)  
 val EditAbility = EditText(this)  
 EditNumber.*hint* = "Введите имя покемона"  
 EditFakt.*hint* = "Введите рост"  
 EditWeight.*hint* = "Введите вес"  
 EditAbility.*hint* = "Введите способности"  
 EditNumber.setText(bdNewText[1])  
 EditFakt.setText(bdNewText[2])  
 EditWeight.setText(bdNewText[3])  
 EditAbility.setText(bdNewText[4])  
 layout.addView(EditNumber)  
 layout.addView(EditFakt)  
 layout.addView(EditWeight)  
 layout.addView(EditAbility)  
 builder.setView(layout)  
 builder.setPositiveButton("OK") **{** dialog, \_ **->** var number = EditNumber.*text*.toString()  
 var height = EditFakt.*text*.toString()  
 var weight = EditWeight.*text*.toString()  
 var ability = EditAbility.*text*.toString()  
 //запись в бд  
 //проверка  
 if(number.*isEmpty*()){  
 number = bdNewText[1];  
 }  
 if(height.*isEmpty*()){  
 height = bdNewText[2];  
 }  
 if(weight.*isEmpty*()){  
 weight = bdNewText[3];  
 }  
 if(ability.*isEmpty*()){  
 ability = bdNewText[4];  
 }  
 val item = Item(bdNewText[0].*toInt*(), number, height, weight, ability)  
 *CoroutineScope*(Dispatchers.IO).*launch* **{** db.getDao().updateItem(item)  
 **}** printInfo()  
 **}** builder.setNegativeButton("Отмена") **{** dialog, \_ **->** dialog.cancel()  
 **}** val dialog = builder.create()  
 dialog.show()  
 printInfo()  
  
 **}**)  
 textList.*adapter* = itemAdapter  
 itemAdapter.notifyDataSetChanged()  
  
 **}** }  
  
 fun ButCleaAll(view: View) {  
 val db = MainDb.getDb(this)  
 *CoroutineScope*(Dispatchers.IO).*launch* **{** // Очистить все таблицы в базе данных  
 db.clearAllTables()  
 withContext(Dispatchers.Main) **{  
 }  
 }** printInfo()  
 }  
 fun AddNewButton(view: View) {  
 val builder = AlertDialog.Builder(this)  
 builder.setTitle("Введите данные")  
 val layout = LinearLayout(this)  
 layout.*orientation* = LinearLayout.*VERTICAL* val EditNumber = EditText(this)  
 val EditFakt = EditText(this)  
 val EditWeight = EditText(this)  
 val EditAbility = EditText(this)  
 EditNumber.*hint* = "Введите имя покемона"  
 EditFakt.*hint* = "Введите рост"  
 EditWeight.*hint* = "Введите вес"  
 EditAbility.*hint* = "Введите способности"  
 layout.addView(EditNumber)  
 layout.addView(EditFakt)  
 layout.addView(EditWeight)  
 layout.addView(EditAbility)  
 builder.setView(layout)  
 builder.setPositiveButton("OK") **{** dialog, \_ **->** val number = EditNumber.*text*.toString()  
 val height = EditFakt.*text*.toString()  
 val weight = EditWeight.*text*.toString()  
 val ability = EditAbility.*text*.toString()  
 if(number.*isEmpty*() || height.*isEmpty*() || weight.*isEmpty*() || ability.*isEmpty*()){  
 val builder = AlertDialog.Builder(this)  
 builder.setTitle("Не сохранено")  
 .setMessage("Необходимо заполнить все поля")  
 .setPositiveButton("ОК") **{** dialog, id **->** dialog.cancel()  
 **}** builder.create()  
 builder.show()  
 }  
 else{  
 val db = MainDb.getDb(this)  
 val item = Item(null, number, weight, height, ability)  
 Thread**{** db.getDao().insertItem(item)  
 **}**.start()  
 dialog.dismiss()  
 printInfo()  
 }  
 **}** builder.setNegativeButton("Отмена") **{** dialog, \_ **->** dialog.cancel()  
 **}** val dialog = builder.create()  
 dialog.show()  
 }  
 fun ButBack(view: View) {  
 val intent = Intent(this, PokemonActivity::class.*java*)  
 startActivity(intent)  
 }  
  
  
}

activity\_item\_list\_row

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout  
 xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".item\_list\_row">  
 <LinearLayout  
 android:layout\_width="300dp"  
 android:layout\_height="wrap\_content"  
 android:orientation="vertical">  
 <TextView  
 android:textColor="@color/black"  
 android:id="@+id/titleTextView"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:textSize="18sp"  
 android:textStyle="bold"  
 android:paddingStart="16dp"  
 android:paddingTop="8dp"  
 android:paddingEnd="16dp"  
 android:paddingBottom="4dp" />  
 <LinearLayout  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:orientation="horizontal">  
 <TextView  
 android:textColor="@color/black"  
 android:text="Рост:"  
 android:textSize="14sp"  
 android:paddingStart="16dp"  
 android:paddingTop="4dp"  
 android:paddingEnd="16dp"  
 android:paddingBottom="8dp"  
 android:layout\_width="100dp"  
 android:layout\_height="wrap\_content"></TextView>  
 <TextView  
 android:textColor="@color/black"  
 android:id="@+id/descriptionTextView"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:textSize="14sp"  
 android:paddingStart="16dp"  
 android:paddingTop="4dp"  
 android:paddingEnd="16dp"  
 android:paddingBottom="8dp" />  
  
 </LinearLayout>  
 <LinearLayout  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:orientation="horizontal">  
 <TextView  
 android:textColor="@color/black"  
 android:text="Вес:"  
 android:textSize="14sp"  
 android:paddingStart="16dp"  
 android:paddingTop="4dp"  
 android:paddingEnd="16dp"  
 android:paddingBottom="8dp"  
 android:layout\_width="100dp"  
 android:layout\_height="wrap\_content"></TextView>  
 <TextView  
 android:textColor="@color/black"  
 android:id="@+id/weightText"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:textSize="14sp"  
 android:paddingStart="16dp"  
 android:paddingTop="4dp"  
 android:paddingEnd="16dp"  
 android:paddingBottom="8dp" />  
  
 </LinearLayout>  
 <LinearLayout  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:orientation="horizontal">  
 <TextView  
 android:textColor="@color/black"  
 android:text="Способности: "  
 android:textSize="14sp"  
 android:paddingStart="16dp"  
 android:paddingTop="4dp"  
 android:paddingEnd="16dp"  
 android:paddingBottom="8dp"  
 android:layout\_width="130dp"  
 android:layout\_height="wrap\_content"></TextView>  
 <TextView  
 android:textColor="@color/black"  
 android:id="@+id/abilityText"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:textSize="14sp"  
 android:paddingStart="16dp"  
 android:paddingTop="4dp"  
 android:paddingEnd="16dp"  
 android:paddingBottom="8dp" />  
  
 </LinearLayout>  
  
 </LinearLayout>  
  
 <ImageButton  
 android:scaleType="centerCrop"  
 android:id="@+id/editButton"  
 android:src="@drawable/edit"  
 android:background="@color/yelow"  
 android:layout\_width="30dp"  
 android:layout\_height="30dp"  
 android:layout\_marginTop="10dp"  
 android:layout\_marginRight="10dp">  
  
 </ImageButton>  
 <ImageButton  
 android:scaleType="centerCrop"  
 android:id="@+id/deleteButton"  
 android:background="@color/yelow"  
 android:src="@drawable/delet"  
 android:layout\_width="30dp"  
 android:layout\_height="30dp"  
 android:layout\_marginTop="10dp"  
 >  
  
 </ImageButton>  
</LinearLayout>

activity\_main

<?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=".MainActivity"  
 android:background="@drawable/font"  
 >  
  
 <TextView  
 android:id="@+id/textView"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="@string/glav"  
 android:textColor="@color/black"  
 android:textSize="20dp"  
 android:textStyle="bold"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintVertical\_bias="0.093" />  
  
 <EditText  
 android:id="@+id/login"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="48dp"  
 android:hint="@string/login"  
 android:textColorHint="@color/black"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.0"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/textView" />  
  
 <EditText  
 android:id="@+id/pass"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="28dp"  
 android:hint="@string/password"  
 android:textColorHint="@color/black"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.0"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/login" />  
  
 <Button  
 android:id="@+id/enterbut"  
 android:layout\_width="200dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="44dp"  
 android:background="@drawable/button"  
 android:text="@string/enter"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.497"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/pass" />  
  
 <Button  
 android:id="@+id/registrbut"  
 android:layout\_width="220dp"  
 android:layout\_height="wrap\_content"  
 android:background="@drawable/button"  
 android:text="@string/registration"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.516"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/enterbut"  
 app:layout\_constraintVertical\_bias="0.087" />  
  
</androidx.constraintlayout.widget.ConstraintLayout>

Activity\_pokemon

<?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=".PokemonActivity"  
 android:background="@drawable/font">  
 <TextView  
 android:id="@+id/textView2"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="40dp"  
 android:text="@string/glav"  
 android:textColor="@color/black"  
 android:textSize="20dp"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent" />  
  
 <EditText  
 android:id="@+id/editText"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="32dp"  
 android:hint="@string/namepers"  
 android:textColorHint="@color/black"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.0"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/textView2" />  
  
 <TextView  
 android:id="@+id/textinfo"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="130dp"  
 android:layout\_marginTop="48dp"  
 android:text=""  
 android:textColor="@color/black"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.498"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/editText" />  
  
 <Button  
 android:id="@+id/button"  
 android:layout\_width="200dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="40dp"  
 android:background="@drawable/button"  
 android:onClick="getResultBut"  
 android:text="@string/butinfo"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.497"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/textinfo" />  
  
 <Button  
 android:layout\_width="200dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="20dp"  
 android:background="@drawable/button"  
 android:onClick="perehod"  
 android:text="@string/buthistori"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.497"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/button"  
 app:layout\_constraintVertical\_bias="0.0" />  
</androidx.constraintlayout.widget.ConstraintLayout>

activity\_registr

<?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="@drawable/font"  
 tools:context=".RegistrActivity">  
  
 <ImageButton  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:background="@color/yelow"  
 android:onClick="btback"  
 android:src="@drawable/back"  
 app:layout\_constraintBottom\_toTopOf="@+id/login1"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.0"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintVertical\_bias="0.0" />  
  
 <EditText  
 android:id="@+id/login1"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="116dp"  
 android:hint="@string/login"  
 android:textColorHint="@color/black"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.0"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent" />  
  
 <EditText  
 android:id="@+id/pass1"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="40dp"  
 android:hint="@string/password"  
 android:textColorHint="@color/black"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.0"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/login1" />  
  
 <Button  
 android:id="@+id/registrbutone"  
 android:layout\_width="220dp"  
 android:layout\_height="wrap\_content"  
 android:background="@drawable/button"  
 android:text="@string/registration"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.497"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/pass1"  
 app:layout\_constraintVertical\_bias="0.405" />  
</androidx.constraintlayout.widget.ConstraintLayout>

activity\_screen\_statistics

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical"  
 tools:context=".ScreenStatistics"  
 android:background="@drawable/font">  
  
 <ImageButton  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:src="@drawable/back"  
 android:background="@color/yelow"  
 android:onClick="ButBack"  
 />  
 <TextView  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Истоия поиска"  
 android:textSize="20dp"  
 android:textColor="@color/black"  
 android:textAlignment="center"  
 />  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="horizontal">  
  
 <Button  
 android:background="@drawable/button"  
 android:id="@+id/clearButton"  
 android:layout\_width="0dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_weight="1"  
 android:text="очистить"  
 android:textColor="@color/white"  
 android:layout\_margin="4dp"  
 android:onClick="ButCleaAll"/>  
  
  
 <Button  
 android:textColor="@color/white"  
 android:background="@drawable/button"  
 android:id="@+id/AddButton"  
 android:layout\_width="0dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_weight="1"  
 android:text="добавить"  
 android:layout\_margin="4dp"  
 android:onClick="AddNewButton"/>  
 </LinearLayout>  
 <ListView  
 android:id="@+id/listView"  
 android:layout\_width="match\_parent"  
 android:layout\_height="0dp"  
 android:layout\_weight="1" />  
  
  
</LinearLayout>

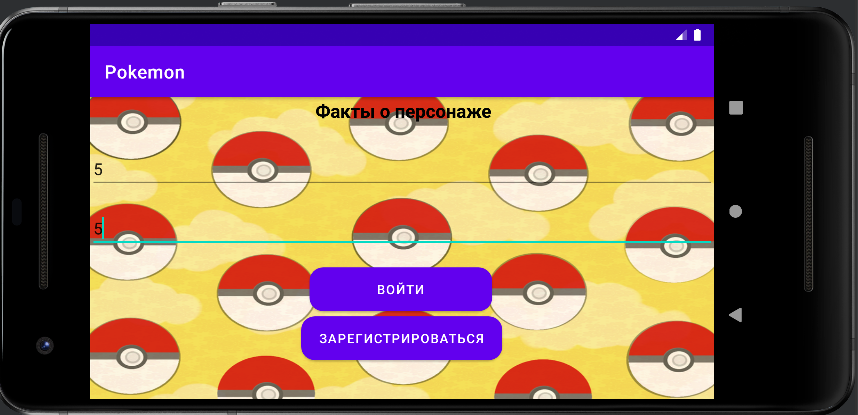
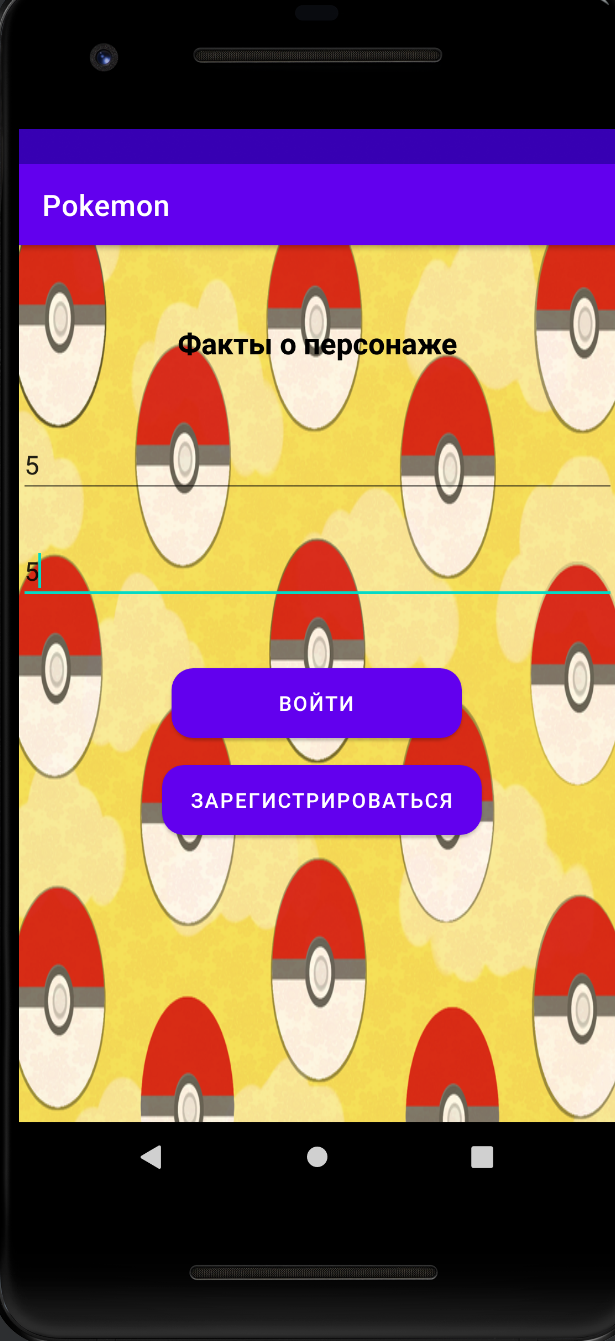
colors.xml

<?xml version="1.0" encoding="utf-8"?>  
<resources>  
 <color name="purple\_200">#FFBB86FC</color>  
 <color name="purple\_500">#FF6200EE</color>  
 <color name="purple\_700">#FF3700B3</color>  
 <color name="teal\_200">#FF03DAC5</color>  
 <color name="teal\_700">#FF018786</color>  
 <color name="black">#FF000000</color>  
 <color name="white">#FFFFFFFF</color>  
 <color name="orang">#FF8C00</color>  
 <color name="yelow">#FFD700</color>  
</resources>

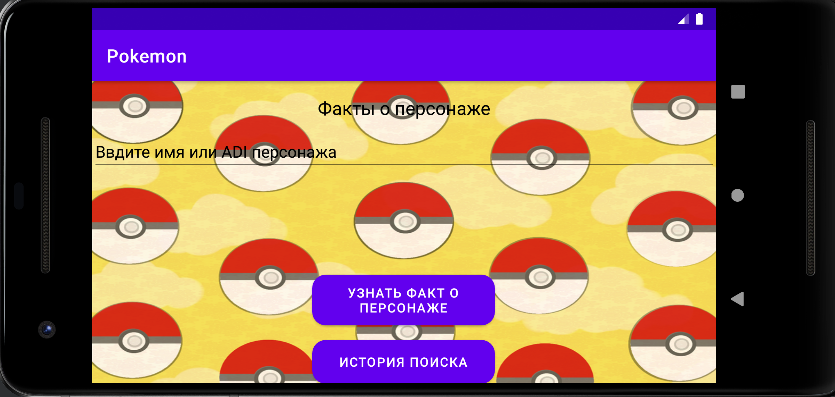
strings.xml

<resources>  
 <string name="app\_name">Pokemon</string>  
 <string name="glav">Факты о персонаже</string>  
 <string name="login">Введите логин</string>  
 <string name="password">Введите пароль</string>  
 <string name="enter">Войти</string>  
 <string name="registration">Зарегистрироваться</string>  
 <string name="namepers">Ввдите имя или ADI персонажа</string>  
 <string name="butinfo">Узнать факт о персонаже</string>  
 <string name="buthistori">История поиска</string>  
</resources>

**Экраны:**



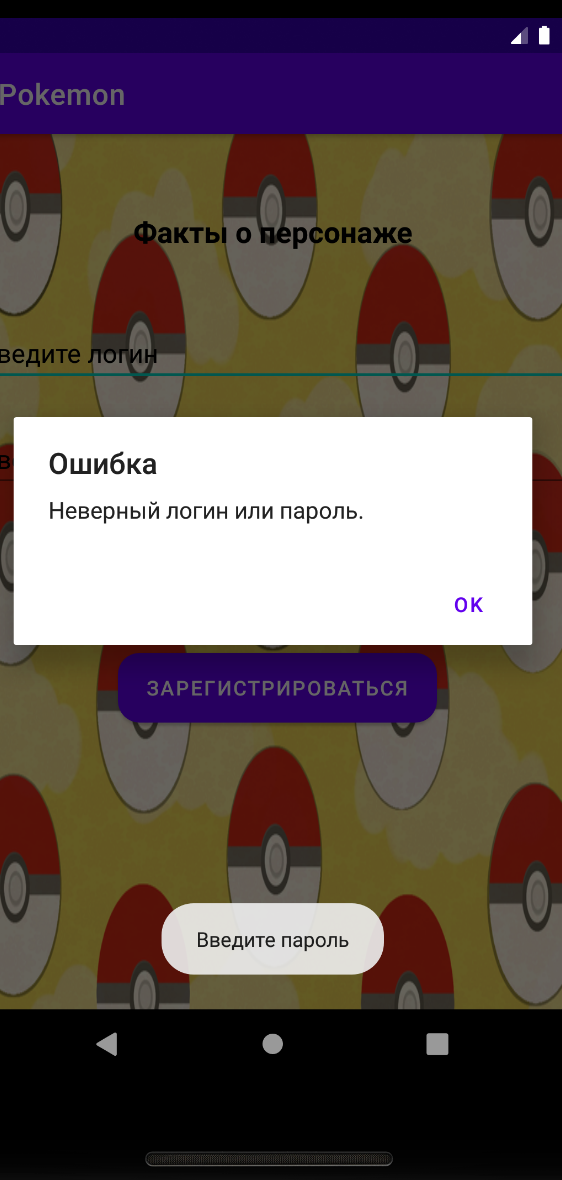




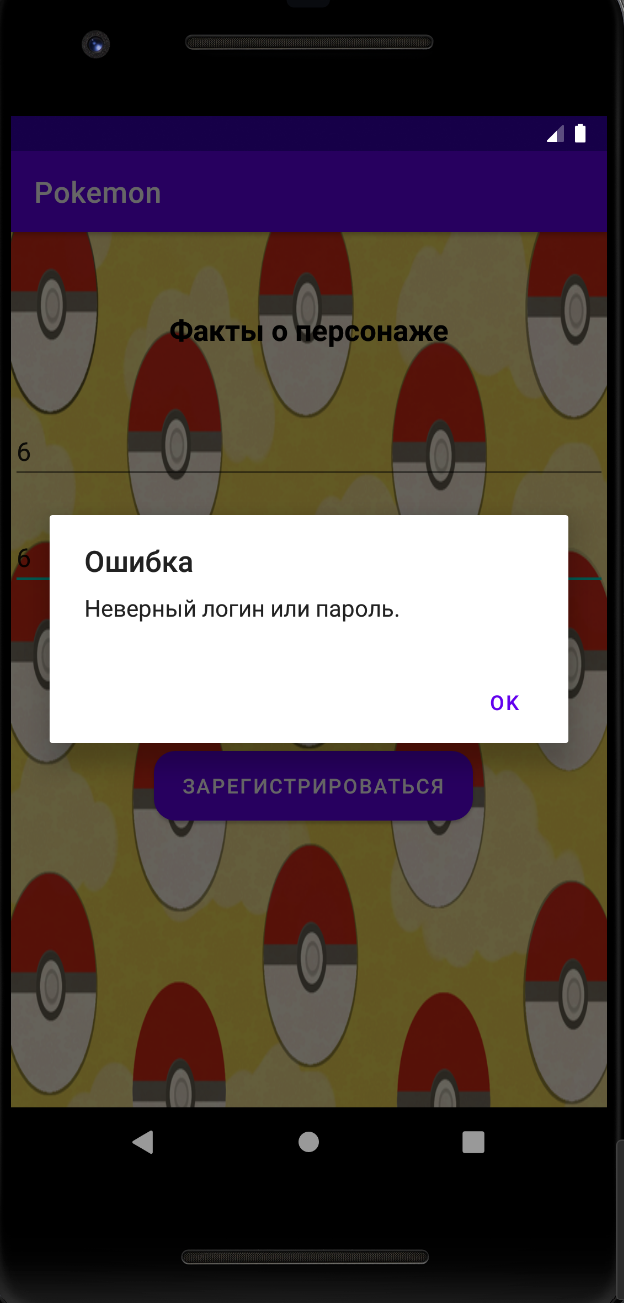


**Тестовые ситуации:**

Если пользователь не будет водить пароль и/или логин, то программа сообщит об ошибке



Если пользователь введет неверный пароль и/или логин то программа сообщит об ошибке



Если пользователь в режиме редактирование не заполнит поля то программа сообщит об этом

