# Приложение В.Текст программы

# АННОТАЦИЯ

В данном программном документе, в разделе «Текст программы» указан текст программы и его описание.

СОДЕРЖАНИЕ

[Приложение В.Текст программы 1](#_Toc73289910)

[АННОТАЦИЯ 1](#_Toc73289911)

[1. ТЕКСТ ПРОГРАММЫ 3](#_Toc73289912)

[1.1. Наименование программы 3](#_Toc73289913)

[1.2. Область применения программы 3](#_Toc73289914)

[1.3. Модули 3](#_Toc73289915)

[1.4. Код программы 4](#_Toc73289916)

# ТЕКСТ ПРОГРАММЫ

## Наименование программы

Наименование - «Мобильное приложения по продаже товаров и услуг DP Stuff Provider».

## Область применения программы

Данная программа предназначена для продажи товаров и услуг.

## Модули

Таблица 1 - Модули программы

| № | Модуль | Описание | Кол-во строк | Размер в КБ |
| --- | --- | --- | --- | --- |
| 1 | CartAdapter.kt | Адаптер для заполнения корзины | 156 | 6.61 |
| 2 | CategoryAdapter.kt | Адаптер для заполнения категорий каталога | 131 | 6.02 |
| 3 | OrderAdapter.kt | Адаптер для заполнения заказов клиента | 49 | 2.18 |
| 4 | ProductAdapter.kt | Адаптер для заполнения информации о товарах | 101 | 4.36 |
| 5 | AccountFragment.kt | Фрагмент аккаунта залогиненного пользователя | 44 | 1.51 |
| 6 | AccountLoginFragment.kt | Фрагмент формы авторизации пользователя | 53 | 2.08 |
| 7 | AccountNotLoginFragment.kt | Фрагмент не залогиненного пользователя | 32 | 0.953 |
| 8 | CartEmptyFragment.kt | Фрагмент пустой корзины | 34 | 0.993 |
| 9 | CartFragment.kt | Фрагмент корзины, корзина заполняется по списку товаров cartList из MainActivity | 56 | 2.09 |
| 10 | CatalogFragment.kt | Фрагмент каталога с поиском товаров по наименованию и категориям | 75 | 2.65 |
| 11 | CheckoutFragment.kt | Фрагмент оформления нового заказа | 118 | 4.86 |
| 12 | ErrorFragment.kt | Фрагмент заглушки на случай ошибок (не отвечает api) | 27 | 0.786 |
| 13 | HomeFragment.kt | Фрагмент домашней страницы со списком популярных товаров | 56 | 1.91 |
| 14 | OrdersFragment.kt | Фрагмент со списком заказов пользователя | 44 | 1.7 |
| 15 | ProductsFragment.kt | Фрагмент со списков товаров (по наименованию или категории) | 74 | 2.7 |
| 16 | RegistrationFragment.kt | Фрагмент регистрации нового пользователя | 87 | 3.43 |
| 17 | ClientApiService.kt | Класс, реализующий все методы Api-интерфейсов | 280 | 10.7 |
| 18 | MainActivity.kt | Основное активити, включающая в себя меню навигации и все фрагменты | 83 | 3.04 |

## Код программы

* + 1. CartAdapter.kt

|  |
| --- |
| package com.example.dpstuffproviderstore.adapter |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| import android.content.Context |
|  |

|  |
| --- |
| import android.view.LayoutInflater |
|  |

|  |
| --- |
| import android.view.View |
|  |

|  |
| --- |
| import android.view.ViewGroup |
|  |

|  |
| --- |
| import android.widget.Button |
|  |

|  |
| --- |
| import android.widget.ImageView |
|  |

|  |
| --- |
| import android.widget.TextView |
|  |

|  |
| --- |
| import android.widget.Toast |
|  |

|  |
| --- |
| import androidx.annotation.NonNull |
|  |

|  |
| --- |
| import androidx.cardview.widget.CardView |
|  |

|  |
| --- |
| import androidx.fragment.app.Fragment |
|  |

|  |
| --- |
| import androidx.recyclerview.widget.RecyclerView |
|  |

|  |
| --- |
| import com.example.dpstuffproviderstore.MainActivity |
|  |

|  |
| --- |
| import com.example.dpstuffproviderstore.R |
|  |

|  |
| --- |
| import com.example.dpstuffproviderstore.`interface`.ICategory |
|  |

|  |
| --- |
| import com.example.dpstuffproviderstore.`interface`.IProduct |
|  |

|  |
| --- |
| import com.example.dpstuffproviderstore.fragment.CartEmptyFragment |
|  |

|  |
| --- |
| import com.example.dpstuffproviderstore.fragment.CartFragment |
|  |

|  |
| --- |
| import com.example.dpstuffproviderstore.fragment.ErrorFragment |
|  |

|  |
| --- |
| import com.example.dpstuffproviderstore.fragment.ProductsFragment |
|  |

|  |
| --- |
| import com.example.dpstuffproviderstore.models.CategoryData |
|  |

|  |
| --- |
| import com.example.dpstuffproviderstore.models.ProductData |
|  |

|  |
| --- |
| import com.example.dpstuffproviderstore.models.ProductImagesData |
|  |

|  |
| --- |
| import com.example.dpstuffproviderstore.other.ClientApiService |
|  |

|  |
| --- |
| import com.google.gson.Gson |
|  |

|  |
| --- |
| import com.squareup.picasso.Picasso |
|  |

|  |
| --- |
| import kotlinx.android.synthetic.main.fragment\_cart.view.\* |
|  |

|  |
| --- |
| import kotlinx.android.synthetic.main.fragment\_catalog.view.\* |
|  |

|  |
| --- |
| import org.w3c.dom.Text |
|  |

|  |
| --- |
| import retrofit2.Call |
|  |

|  |
| --- |
| import retrofit2.Callback |
|  |

|  |
| --- |
| import retrofit2.Response |
|  |

|  |
| --- |
| import retrofit2.Retrofit |
|  |

|  |
| --- |
| import retrofit2.converter.gson.GsonConverterFactory |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| /\*\* |
|  |

|  |
| --- |
| \* Адаптер для заполнения корзины |
|  |

|  |
| --- |
| \*/ |
|  |

|  |
| --- |
| internal class CartAdapter(private var productsList: List<ProductData>, private var fragment: CartFragment) : RecyclerView.Adapter<CartAdapter.MyViewHolder>() { |
|  |

|  |
| --- |
| internal class MyViewHolder(view: View) : RecyclerView.ViewHolder(view){ |
|  |

|  |
| --- |
| val productImage: ImageView = view.findViewById(R.id.imageProductCart) |
|  |

|  |
| --- |
| val productPrice: TextView = view.findViewById(R.id.tvPriceCart) |
|  |

|  |
| --- |
| val productTitle: TextView = view.findViewById(R.id.tvTitleProductCart) |
|  |

|  |
| --- |
| val productStore: TextView = view.findViewById(R.id.tvStoreProductCart) |
|  |

|  |
| --- |
| val productRating: TextView = view.findViewById(R.id.tvRatingProductCart) |
|  |

|  |
| --- |
| val productQuantity: TextView = view.findViewById(R.id.tvQuantityCart) |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| val btnPlus: Button = view.findViewById(R.id.btnPlusCart) |
|  |

|  |
| --- |
| val btnMinus: Button = view.findViewById(R.id.btnMinusCart) |
|  |

|  |
| --- |
| val btnTrash: Button = view.findViewById(R.id.btnDeleteCart) |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| @NonNull |
|  |

|  |
| --- |
| override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): MyViewHolder { |
|  |

|  |
| --- |
| val itemView = LayoutInflater.from(parent.context) |
|  |

|  |
| --- |
| .inflate(R.layout.item\_cart, parent, false) |
|  |

|  |
| --- |
| return MyViewHolder(itemView) |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| override fun onBindViewHolder(holder: CartAdapter.MyViewHolder, position: Int) { |
|  |

|  |
| --- |
| val mainActivity = fragment.activity as MainActivity |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| holder.productPrice.text = "${productsList[position].price} ₽" |
|  |

|  |
| --- |
| holder.productTitle.text = productsList[position].name |
|  |

|  |
| --- |
| holder.productStore.text = "Магазин: ${productsList[position].store}" |
|  |

|  |
| --- |
| holder.productRating.text = "Рейтинг: ${productsList[position].rating}/5" |
|  |

|  |
| --- |
| holder.productQuantity.text = "${productsList[position].quantity ?: 1} Шт" |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| ClientApiService().getImages(productsList[position].id) { |
|  |

|  |
| --- |
| if(it != null) { |
|  |

|  |
| --- |
| Picasso.with(holder.itemView.context) |
|  |

|  |
| --- |
| .load(it!![0].imageUrl) |
|  |

|  |
| --- |
| .placeholder(R.drawable.img\_placeholder) |
|  |

|  |
| --- |
| .error(R.drawable.img\_placeholder) |
|  |

|  |
| --- |
| .into(holder.productImage) |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
| else{ |
|  |

|  |
| --- |
| val fragment = ErrorFragment() |
|  |

|  |
| --- |
| fragment.statusCode = "404" |
|  |

|  |
| --- |
| mainActivity.makeCurrentFragment(fragment) |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| holder.btnPlus.setOnClickListener { |
|  |

|  |
| --- |
| mainActivity.cartList[position].quantity = (productsList[position].quantity ?: 1) + 1 |
|  |

|  |
| --- |
| val editor = mainActivity.getSharedPreferences("sp", Context.MODE\_PRIVATE).edit() |
|  |

|  |
| --- |
| editor.putString("cartList", Gson().toJson(mainActivity.cartList)) |
|  |

|  |
| --- |
| editor.apply() |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| holder.productQuantity.text = "${mainActivity.cartList[position].quantity ?: 1} Шт" |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| var summ : Double = 0.0 |
|  |

|  |
| --- |
| for(i in mainActivity.cartList){ |
|  |

|  |
| --- |
| summ += i.price!! \* (i.quantity ?: 1) |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
| fragment.inflate!!.tvCheckout.text = "Итого: ${summ} ₽" |
|  |

|  |
| --- |
| fragment.summ = summ |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| fun removeProduct(product : ProductData){ |
|  |

|  |
| --- |
| mainActivity.cartList.remove(product) |
|  |

|  |
| --- |
| val editor = mainActivity.getSharedPreferences("sp", Context.MODE\_PRIVATE).edit() |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| if(mainActivity.cartList.size == 0){ |
|  |

|  |
| --- |
| mainActivity.makeCurrentFragment(CartEmptyFragment()) |
|  |

|  |
| --- |
| mainActivity.cartFragment = CartEmptyFragment() |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| editor.putString("cartList", "").apply() |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
| else{ |
|  |

|  |
| --- |
| editor.putString("cartList", Gson().toJson(mainActivity.cartList)).apply() |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| fragment.inflate!!.recyclerCart.adapter = CartAdapter(mainActivity.cartList, fragment) |
|  |

|  |
| --- |
| fragment.inflate!!.tvTitle.text = "${mainActivity.cartList.size} Товара:" |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| var summ : Double = 0.0 |
|  |

|  |
| --- |
| for(i in mainActivity.cartList){ |
|  |

|  |
| --- |
| summ += i.price!! \* (i.quantity ?: 1) |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
| fragment.inflate!!.tvCheckout.text = "Итого: ${summ} ₽" |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| holder.btnMinus.setOnClickListener { |
|  |

|  |
| --- |
| mainActivity.cartList[position].quantity = (productsList[position].quantity ?: 1) - 1 |
|  |

|  |
| --- |
| if (mainActivity.cartList[position].quantity!! <= 0){ |
|  |

|  |
| --- |
| removeProduct(productsList[position]) |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
| else{ |
|  |

|  |
| --- |
| val editor = mainActivity.getSharedPreferences("sp", Context.MODE\_PRIVATE).edit() |
|  |

|  |
| --- |
| editor.putString("cartList", Gson().toJson(mainActivity.cartList)) |
|  |

|  |
| --- |
| editor.apply() |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| holder.productQuantity.text = "${mainActivity.cartList[position].quantity ?: 1} Шт" |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| var summ : Double = 0.0 |
|  |

|  |
| --- |
| for(i in mainActivity.cartList){ |
|  |

|  |
| --- |
| summ += i.price!! \* (i.quantity ?: 1) |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
| fragment.inflate!!.tvCheckout.text = "Итого: ${summ} ₽" |
|  |

|  |
| --- |
| fragment.summ = summ |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| holder.btnTrash.setOnClickListener { |
|  |

|  |
| --- |
| removeProduct(productsList[position]) |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| override fun getItemCount(): Int { |
|  |

|  |
| --- |
| return productsList.size |
|  |

|  |
| --- |
| } |
|  |

}

* + 1. CategoryAdapter.kt

|  |
| --- |
| package com.example.dpstuffproviderstore.adapter |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| import android.app.Activity |
|  |

|  |
| --- |
| import android.content.Context |
|  |

|  |
| --- |
| import android.content.Intent |
|  |

|  |
| --- |
| import android.os.Bundle |
|  |

|  |
| --- |
| import android.text.TextUtils.replace |
|  |

|  |
| --- |
| import android.util.Log |
|  |

|  |
| --- |
| import android.view.LayoutInflater |
|  |

|  |
| --- |
| import android.view.View |
|  |

|  |
| --- |
| import android.view.ViewGroup |
|  |

|  |
| --- |
| import android.widget.Button |
|  |

|  |
| --- |
| import android.widget.TextView |
|  |

|  |
| --- |
| import android.widget.Toast |
|  |

|  |
| --- |
| import androidx.annotation.NonNull |
|  |

|  |
| --- |
| import androidx.cardview.widget.CardView |
|  |

|  |
| --- |
| import androidx.core.content.ContextCompat |
|  |

|  |
| --- |
| import androidx.fragment.app.Fragment |
|  |

|  |
| --- |
| import androidx.fragment.app.FragmentActivity |
|  |

|  |
| --- |
| import androidx.fragment.app.FragmentManager |
|  |

|  |
| --- |
| import androidx.recyclerview.widget.RecyclerView |
|  |

|  |
| --- |
| import com.example.dpstuffproviderstore.MainActivity |
|  |

|  |
| --- |
| import com.example.dpstuffproviderstore.R |
|  |

|  |
| --- |
| import com.example.dpstuffproviderstore.`interface`.ICategory |
|  |

|  |
| --- |
| import com.example.dpstuffproviderstore.fragment.CatalogFragment |
|  |

|  |
| --- |
| import com.example.dpstuffproviderstore.fragment.ErrorFragment |
|  |

|  |
| --- |
| import com.example.dpstuffproviderstore.fragment.HomeFragment |
|  |

|  |
| --- |
| import com.example.dpstuffproviderstore.fragment.ProductsFragment |
|  |

|  |
| --- |
| import com.example.dpstuffproviderstore.models.CategoryData |
|  |

|  |
| --- |
| import com.example.dpstuffproviderstore.other.ClientApiService |
|  |

|  |
| --- |
| import com.google.android.material.internal.ContextUtils.getActivity |
|  |

|  |
| --- |
| import com.squareup.picasso.Picasso |
|  |

|  |
| --- |
| import kotlinx.android.synthetic.main.activity\_main.\* |
|  |

|  |
| --- |
| import kotlinx.android.synthetic.main.fragment\_catalog.\* |
|  |

|  |
| --- |
| import kotlinx.android.synthetic.main.fragment\_catalog.view.\* |
|  |

|  |
| --- |
| import retrofit2.Call |
|  |

|  |
| --- |
| import retrofit2.Callback |
|  |

|  |
| --- |
| import retrofit2.Response |
|  |

|  |
| --- |
| import retrofit2.Retrofit |
|  |

|  |
| --- |
| import retrofit2.converter.gson.GsonConverterFactory |
|  |

|  |
| --- |
| import kotlinx.android.synthetic.main.activity\_main.\* |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| /\*\* |
|  |

|  |
| --- |
| \* Адаптер для заполнения категорий каталога |
|  |

|  |
| --- |
| \*/ |
|  |

|  |
| --- |
| internal class CategoryAdapter(private var categoryList: List<CategoryData>, private var fragment: CatalogFragment, private var parentCategoryList: List<CategoryData>?) : RecyclerView.Adapter<CategoryAdapter.MyViewHolder>(){ |
|  |

|  |
| --- |
| internal class MyViewHolder(view: View) : RecyclerView.ViewHolder(view){ |
|  |

|  |
| --- |
| val title : TextView = view.findViewById(R.id.tvTitleCategory) |
|  |

|  |
| --- |
| val cardCategory : CardView = view.findViewById(R.id.cardCategory) |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| @NonNull |
|  |

|  |
| --- |
| override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): MyViewHolder { |
|  |

|  |
| --- |
| val itemView = LayoutInflater.from(parent.context) |
|  |

|  |
| --- |
| .inflate(R.layout.item\_category, parent, false) |
|  |

|  |
| --- |
| return MyViewHolder(itemView) |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| override fun onBindViewHolder(holder: MyViewHolder, position: Int) { |
|  |

|  |
| --- |
| holder.title.text = categoryList[position].name |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| holder.cardCategory.setOnClickListener { |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| ClientApiService().getChildCategories(categoryList[position].name) { |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| if(it != null){ |
|  |

|  |
| --- |
| fragment.inflate!!.recyclerCategory.adapter = CategoryAdapter(it, fragment, categoryList) |
|  |

|  |
| --- |
| fragment.inflate!!.titleCategory.visibility = View.VISIBLE |
|  |

|  |
| --- |
| fragment.inflate!!.titleCategory.text = it[0].parentName |
|  |

|  |
| --- |
| fragment.inflate!!.linkToCategory.visibility = View.VISIBLE |
|  |

|  |
| --- |
| fragment.inflate!!.categoryImg.visibility = View.VISIBLE |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| if(!categoryList[position].imageUrl.isNullOrEmpty()){ |
|  |

|  |
| --- |
| Picasso.with(holder.itemView.context) |
|  |

|  |
| --- |
| .load(categoryList[position].imageUrl) |
|  |

|  |
| --- |
| .placeholder(R.drawable.img\_placeholder) |
|  |

|  |
| --- |
| .error(R.drawable.img\_placeholder) |
|  |

|  |
| --- |
| .into(fragment.inflate!!.categoryImg) |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
| else{ |
|  |

|  |
| --- |
| fragment.inflate!!.categoryImg.visibility = View.GONE |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| fragment.inflate!!.btnUndo.visibility = View.VISIBLE |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| fragment.inflate!!.btnUndo.setOnClickListener { |
|  |

|  |
| --- |
| ClientApiService().getMainCategories() { |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| if(it != null){ |
|  |

|  |
| --- |
| fragment.inflate!!.recyclerCategory.adapter = CategoryAdapter(it, fragment, null) |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| fragment.inflate!!.titleCategory.visibility = View.GONE |
|  |

|  |
| --- |
| fragment.inflate!!.linkToCategory.visibility = View.GONE |
|  |

|  |
| --- |
| fragment.inflate!!.categoryImg.visibility = View.GONE |
|  |

|  |
| --- |
| fragment.inflate!!.btnUndo.visibility = View.GONE |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
| else{ |
|  |

|  |
| --- |
| val mainActivity = fragment.activity as MainActivity |
|  |

|  |
| --- |
| val fragment = ErrorFragment() |
|  |

|  |
| --- |
| fragment.statusCode = "404" |
|  |

|  |
| --- |
| mainActivity.makeCurrentFragment(fragment) |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| fragment.inflate!!.linkToCategory.setOnClickListener { |
|  |

|  |
| --- |
| val mainActivity = fragment.activity as MainActivity |
|  |

|  |
| --- |
| mainActivity.modeSearch = MainActivity.EnumModeSearch.CATEGORY |
|  |

|  |
| --- |
| mainActivity.productCategory = fragment.inflate!!.titleCategory.text.toString() |
|  |

|  |
| --- |
| mainActivity.makeCurrentFragment(ProductsFragment()) |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
| else{ |
|  |

|  |
| --- |
| val mainActivity = fragment.activity as MainActivity |
|  |

|  |
| --- |
| mainActivity.modeSearch = MainActivity.EnumModeSearch.CATEGORY |
|  |

|  |
| --- |
| mainActivity.productCategory = categoryList[position].name |
|  |

|  |
| --- |
| mainActivity.makeCurrentFragment(ProductsFragment()) |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
| /\*override fun onFailure(call: Call<List<CategoryData>>, t: Throwable){ |
|  |

|  |
| --- |
| val mainActivity = fragment.activity as MainActivity |
|  |

|  |
| --- |
| val fragment = ErrorFragment() |
|  |

|  |
| --- |
| mainActivity.makeCurrentFragment(fragment) |
|  |

|  |
| --- |
| }\*/ |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| override fun getItemCount(): Int { |
|  |

|  |
| --- |
| return categoryList.size |
|  |

|  |
| --- |
| } |
|  |

}

* + 1. OrderAdapter.kt

|  |
| --- |
| package com.example.dpstuffproviderstore.adapter |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| import android.view.LayoutInflater |
|  |

|  |
| --- |
| import android.view.View |
|  |

|  |
| --- |
| import android.view.ViewGroup |
|  |

|  |
| --- |
| import android.widget.Button |
|  |

|  |
| --- |
| import android.widget.ImageView |
|  |

|  |
| --- |
| import android.widget.TextView |
|  |

|  |
| --- |
| import androidx.annotation.NonNull |
|  |

|  |
| --- |
| import androidx.recyclerview.widget.RecyclerView |
|  |

|  |
| --- |
| import com.example.dpstuffproviderstore.R |
|  |

|  |
| --- |
| import com.example.dpstuffproviderstore.fragment.CartFragment |
|  |

|  |
| --- |
| import com.example.dpstuffproviderstore.fragment.OrdersFragment |
|  |

|  |
| --- |
| import com.example.dpstuffproviderstore.models.ClientData |
|  |

|  |
| --- |
| import com.example.dpstuffproviderstore.models.OrderData |
|  |

|  |
| --- |
| import com.example.dpstuffproviderstore.models.ProductData |
|  |

|  |
| --- |
| import com.example.dpstuffproviderstore.other.ClientApiService |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| /\*\* |
|  |

|  |
| --- |
| \* Адаптер для заполнения заказов клиента |
|  |

|  |
| --- |
| \*/ |
|  |

|  |
| --- |
| internal class OrderAdapter(private var ordersList: List<OrderData>, private var fragment: OrdersFragment) : RecyclerView.Adapter<OrderAdapter.MyViewHolder>() { |
|  |

|  |
| --- |
| internal class MyViewHolder(view: View) : RecyclerView.ViewHolder(view){ |
|  |

|  |
| --- |
| val orderInfo: TextView = view.findViewById(R.id.tvOrderInfo) |
|  |

|  |
| --- |
| val orderDate: TextView = view.findViewById(R.id.tvOrderDate) |
|  |

|  |
| --- |
| val orderStatus: TextView = view.findViewById(R.id.tvOrderStatus) |
|  |

|  |
| --- |
| val orderNum: TextView = view.findViewById(R.id.tvOrderNum) |
|  |

|  |
| --- |
| val orderCode: TextView = view.findViewById(R.id.tvOrderCode) |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| @NonNull |
|  |

|  |
| --- |
| override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): OrderAdapter.MyViewHolder { |
|  |

|  |
| --- |
| val itemView = LayoutInflater.from(parent.context) |
|  |

|  |
| --- |
| .inflate(R.layout.item\_order, parent, false) |
|  |

|  |
| --- |
| return OrderAdapter.MyViewHolder(itemView) |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| override fun onBindViewHolder(holder: OrderAdapter.MyViewHolder, position: Int) { |
|  |

|  |
| --- |
| holder.orderNum.text = "Заказ №${ordersList[position].id}" |
|  |

|  |
| --- |
| holder.orderDate.text = "${ordersList[position].deliveryDate}" |
|  |

|  |
| --- |
| holder.orderStatus.text = "${ordersList[position].status}" |
|  |

|  |
| --- |
| holder.orderInfo.text = "Итого: ${ordersList[position].summ} ₽" |
|  |

|  |
| --- |
| holder.orderCode.text = "Код: ${ordersList[position].codeToFinish}" |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| override fun getItemCount(): Int { |
|  |

|  |
| --- |
| return ordersList.size |
|  |

|  |
| --- |
| } |
|  |

}

* + 1. ProductAdapter.kt

|  |
| --- |
| package com.example.dpstuffproviderstore.adapter |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| import android.content.Context |
|  |

|  |
| --- |
| import android.util.Log |
|  |

|  |
| --- |
| import android.view.LayoutInflater |
|  |

|  |
| --- |
| import android.view.View |
|  |

|  |
| --- |
| import android.view.ViewGroup |
|  |

|  |
| --- |
| import android.widget.Button |
|  |

|  |
| --- |
| import android.widget.ImageView |
|  |

|  |
| --- |
| import android.widget.TextView |
|  |

|  |
| --- |
| import android.widget.Toast |
|  |

|  |
| --- |
| import androidx.annotation.NonNull |
|  |

|  |
| --- |
| import androidx.cardview.widget.CardView |
|  |

|  |
| --- |
| import androidx.fragment.app.Fragment |
|  |

|  |
| --- |
| import androidx.recyclerview.widget.RecyclerView |
|  |

|  |
| --- |
| import com.example.dpstuffproviderstore.MainActivity |
|  |

|  |
| --- |
| import com.example.dpstuffproviderstore.R |
|  |

|  |
| --- |
| import com.example.dpstuffproviderstore.`interface`.ICategory |
|  |

|  |
| --- |
| import com.example.dpstuffproviderstore.`interface`.IProduct |
|  |

|  |
| --- |
| import com.example.dpstuffproviderstore.fragment.CartFragment |
|  |

|  |
| --- |
| import com.example.dpstuffproviderstore.fragment.ErrorFragment |
|  |

|  |
| --- |
| import com.example.dpstuffproviderstore.fragment.ProductsFragment |
|  |

|  |
| --- |
| import com.example.dpstuffproviderstore.models.CategoryData |
|  |

|  |
| --- |
| import com.example.dpstuffproviderstore.models.ProductData |
|  |

|  |
| --- |
| import com.example.dpstuffproviderstore.models.ProductImagesData |
|  |

|  |
| --- |
| import com.example.dpstuffproviderstore.other.ClientApiService |
|  |

|  |
| --- |
| import com.google.gson.Gson |
|  |

|  |
| --- |
| import com.squareup.picasso.Picasso |
|  |

|  |
| --- |
| import kotlinx.android.synthetic.main.fragment\_catalog.view.\* |
|  |

|  |
| --- |
| import retrofit2.Call |
|  |

|  |
| --- |
| import retrofit2.Callback |
|  |

|  |
| --- |
| import retrofit2.Response |
|  |

|  |
| --- |
| import retrofit2.Retrofit |
|  |

|  |
| --- |
| import retrofit2.converter.gson.GsonConverterFactory |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| /\*\* |
|  |

|  |
| --- |
| \* Адаптер для заполнения информации о товарах |
|  |

|  |
| --- |
| \*/ |
|  |

|  |
| --- |
| internal class ProductAdapter(private var productsList: List<ProductData>, private var fragment: Fragment) : RecyclerView.Adapter<ProductAdapter.MyViewHolder>() { |
|  |

|  |
| --- |
| internal class MyViewHolder(view: View) : RecyclerView.ViewHolder(view){ |
|  |

|  |
| --- |
| val productImage: ImageView = view.findViewById(R.id.imageProduct) |
|  |

|  |
| --- |
| val productPrice: TextView = view.findViewById(R.id.tvPrice) |
|  |

|  |
| --- |
| val productTitle: TextView = view.findViewById(R.id.tvTitleProduct) |
|  |

|  |
| --- |
| val productStore: TextView = view.findViewById(R.id.tvStoreProduct) |
|  |

|  |
| --- |
| val productRating: TextView = view.findViewById(R.id.tvRatingProduct) |
|  |

|  |
| --- |
| val btnCart: Button = view.findViewById(R.id.btnAddCart) |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| @NonNull |
|  |

|  |
| --- |
| override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): MyViewHolder { |
|  |

|  |
| --- |
| val itemView = LayoutInflater.from(parent.context) |
|  |

|  |
| --- |
| .inflate(R.layout.item\_product, parent, false) |
|  |

|  |
| --- |
| return MyViewHolder(itemView) |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| override fun onBindViewHolder(holder: ProductAdapter.MyViewHolder, position: Int) { |
|  |

|  |
| --- |
| val mainActivity = fragment.activity as MainActivity |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| holder.productPrice.text = "${productsList[position].price} ₽" |
|  |

|  |
| --- |
| holder.productTitle.text = productsList[position].name |
|  |

|  |
| --- |
| holder.productStore.text = "Магазин: ${productsList[position].store}" |
|  |

|  |
| --- |
| holder.productRating.text = "Рейтинг: ${productsList[position].rating}/5" |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| holder.btnCart.setOnClickListener { |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| if(mainActivity.cartList.contains(productsList[position])){ |
|  |

|  |
| --- |
| Toast.makeText(holder.itemView.context, "Товар уже добавлен в корзину", Toast.LENGTH\_LONG).show() |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
| else{ |
|  |

|  |
| --- |
| Toast.makeText(holder.itemView.context, "${productsList[position].name} Добавлен в корзину", Toast.LENGTH\_LONG).show() |
|  |

|  |
| --- |
| mainActivity.cartList.add(productsList[position]) |
|  |

|  |
| --- |
| mainActivity.cartFragment = CartFragment() |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| val editor = mainActivity.getSharedPreferences("sp", Context.MODE\_PRIVATE).edit() |
|  |

|  |
| --- |
| editor.putString("cartList", Gson().toJson(mainActivity.cartList)) |
|  |

|  |
| --- |
| editor.apply() |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
| Log.i("myLog", "add to cart: ${Gson().toJson(mainActivity.cartList)}") |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| ClientApiService().getImages(productsList[position].id) { |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| if(it != null){ |
|  |

|  |
| --- |
| Picasso.with(holder.itemView.context) |
|  |

|  |
| --- |
| .load(it[0].imageUrl) |
|  |

|  |
| --- |
| .placeholder(R.drawable.img\_placeholder) |
|  |

|  |
| --- |
| .error(R.drawable.img\_placeholder) |
|  |

|  |
| --- |
| .into(holder.productImage) |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
| else{ |
|  |

|  |
| --- |
| val fragment = ErrorFragment() |
|  |

|  |
| --- |
| fragment.statusCode = "404" |
|  |

|  |
| --- |
| mainActivity.makeCurrentFragment(fragment) |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| override fun getItemCount(): Int { |
|  |

|  |
| --- |
| return productsList.size |
|  |

|  |
| --- |
| } |
|  |

}

* + 1. AccountFragment.kt

|  |
| --- |
| package com.example.dpstuffproviderstore.fragment |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| import android.content.Context |
|  |

|  |
| --- |
| import android.os.Bundle |
|  |

|  |
| --- |
| import androidx.fragment.app.Fragment |
|  |

|  |
| --- |
| import android.view.LayoutInflater |
|  |

|  |
| --- |
| import android.view.View |
|  |

|  |
| --- |
| import android.view.ViewGroup |
|  |

|  |
| --- |
| import com.example.dpstuffproviderstore.MainActivity |
|  |

|  |
| --- |
| import com.example.dpstuffproviderstore.R |
|  |

|  |
| --- |
| import kotlinx.android.synthetic.main.fragment\_account.view.\* |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| /\*\* |
|  |

|  |
| --- |
| \* Фрагмент аккаунта залогиненного пользователя |
|  |

|  |
| --- |
| \*/ |
|  |

|  |
| --- |
| class AccountFragment : Fragment() { |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| var testText: String? = null |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| override fun onCreateView(inflater: LayoutInflater, container: ViewGroup?, |
|  |

|  |
| --- |
| savedInstanceState: Bundle?): View? { |
|  |

|  |
| --- |
| val mainActivity = activity as MainActivity |
|  |

|  |
| --- |
| val inflate = inflater.inflate(R.layout.fragment\_account, container, false) |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| inflate.tvTestName.text = "${mainActivity.client!!.lastName} ${mainActivity.client!!.firstName} ${mainActivity.client!!.patronymic}" |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| inflate.btnOut.setOnClickListener { |
|  |

|  |
| --- |
| val editor = mainActivity.getSharedPreferences("sp", Context.MODE\_PRIVATE).edit() |
|  |

|  |
| --- |
| editor.putBoolean("isLogin", false) |
|  |

|  |
| --- |
| editor.apply() |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| mainActivity.accountFragment = AccountNotLoginFragment() |
|  |

|  |
| --- |
| mainActivity.client = null |
|  |

|  |
| --- |
| mainActivity.makeCurrentFragment(AccountNotLoginFragment()) |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| inflate.cardOrders.setOnClickListener { |
|  |

|  |
| --- |
| mainActivity.makeCurrentFragment(OrdersFragment()) |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| return inflate |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
|  |
|  |

}

* + 1. AccountLoginFragment.kt

|  |
| --- |
| package com.example.dpstuffproviderstore.fragment |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| import android.content.Context |
|  |

|  |
| --- |
| import android.os.Bundle |
|  |

|  |
| --- |
| import android.util.Log |
|  |

|  |
| --- |
| import androidx.fragment.app.Fragment |
|  |

|  |
| --- |
| import android.view.LayoutInflater |
|  |

|  |
| --- |
| import android.view.View |
|  |

|  |
| --- |
| import android.view.ViewGroup |
|  |

|  |
| --- |
| import android.widget.Toast |
|  |

|  |
| --- |
| import com.example.dpstuffproviderstore.MainActivity |
|  |

|  |
| --- |
| import com.example.dpstuffproviderstore.R |
|  |

|  |
| --- |
| import com.example.dpstuffproviderstore.other.ClientApiService |
|  |

|  |
| --- |
| import kotlinx.android.synthetic.main.fragment\_account\_login.\* |
|  |

|  |
| --- |
| import kotlinx.android.synthetic.main.fragment\_account\_login.view.\* |
|  |

|  |
| --- |
| import kotlin.math.log |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| /\*\* |
|  |

|  |
| --- |
| \* Фрагмент формы авторизации пользователя |
|  |

|  |
| --- |
| \*/ |
|  |

|  |
| --- |
| class AccountLoginFragment : Fragment() { |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| override fun onCreateView(inflater: LayoutInflater, container: ViewGroup?, |
|  |

|  |
| --- |
| savedInstanceState: Bundle?): View? { |
|  |

|  |
| --- |
| val mainActivity = activity as MainActivity |
|  |

|  |
| --- |
| val inflate = inflater.inflate(R.layout.fragment\_account\_login, container, false) |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| inflate.btnLogin.setOnClickListener { |
|  |

|  |
| --- |
| ClientApiService().getClient(inputTextLogin.text.toString(), inputTextPassword.text.toString()){ |
|  |

|  |
| --- |
| if(it != null){ |
|  |

|  |
| --- |
| mainActivity.client = it |
|  |

|  |
| --- |
| val editor = mainActivity.getSharedPreferences("sp", Context.MODE\_PRIVATE).edit() |
|  |

|  |
| --- |
| editor.putBoolean("isLogin", true) |
|  |

|  |
| --- |
| editor.putString("login", inputTextLogin.text.toString()) |
|  |

|  |
| --- |
| editor.putString("password", inputTextPassword.text.toString()) |
|  |

|  |
| --- |
| editor.apply() |
|  |

|  |
| --- |
| mainActivity.accountFragment = AccountFragment() |
|  |

|  |
| --- |
| mainActivity.makeCurrentFragment(AccountFragment()) |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
| else{ |
|  |

|  |
| --- |
| Toast.makeText(requireContext(), "Неправильный логин и/или пароль", Toast.LENGTH\_LONG).show() |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| inflate.btnReg.setOnClickListener { |
|  |

|  |
| --- |
| mainActivity.makeCurrentFragment(RegistrationFragment()) |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| return inflate |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
|  |
|  |

}

* + 1. AccountNotLoginFragment.kt

|  |
| --- |
| package com.example.dpstuffproviderstore.fragment |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| import android.os.Bundle |
|  |

|  |
| --- |
| import androidx.fragment.app.Fragment |
|  |

|  |
| --- |
| import android.view.LayoutInflater |
|  |

|  |
| --- |
| import android.view.View |
|  |

|  |
| --- |
| import android.view.ViewGroup |
|  |

|  |
| --- |
| import android.widget.Toast |
|  |

|  |
| --- |
| import com.example.dpstuffproviderstore.MainActivity |
|  |

|  |
| --- |
| import com.example.dpstuffproviderstore.R |
|  |

|  |
| --- |
| import kotlinx.android.synthetic.main.fragment\_account\_notlogin.view.\* |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| /\*\* |
|  |

|  |
| --- |
| \* Фрагмент не залогиненного пользователя |
|  |

|  |
| --- |
| \*/ |
|  |

|  |
| --- |
| class AccountNotLoginFragment : Fragment() { |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| override fun onCreateView( |
|  |

|  |
| --- |
| inflater: LayoutInflater, container: ViewGroup?, |
|  |

|  |
| --- |
| savedInstanceState: Bundle? |
|  |

|  |
| --- |
| ): View? { |
|  |

|  |
| --- |
| var inflate = inflater.inflate(R.layout.fragment\_account\_notlogin, container, false) |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| inflate.btnLogin.setOnClickListener { |
|  |

|  |
| --- |
| val mainActivity = activity as MainActivity |
|  |

|  |
| --- |
| mainActivity.makeCurrentFragment(AccountLoginFragment()) |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| return inflate |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
|  |
|  |

}

* + 1. CartEmptyFragment.kt

|  |
| --- |
| package com.example.dpstuffproviderstore.fragment |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| import android.os.Bundle |
|  |

|  |
| --- |
| import androidx.fragment.app.Fragment |
|  |

|  |
| --- |
| import android.view.LayoutInflater |
|  |

|  |
| --- |
| import android.view.View |
|  |

|  |
| --- |
| import android.view.ViewGroup |
|  |

|  |
| --- |
| import com.example.dpstuffproviderstore.MainActivity |
|  |

|  |
| --- |
| import com.example.dpstuffproviderstore.R |
|  |

|  |
| --- |
| import kotlinx.android.synthetic.main.activity\_main.\* |
|  |

|  |
| --- |
| import kotlinx.android.synthetic.main.fragment\_cart.view.\* |
|  |

|  |
| --- |
| import kotlinx.android.synthetic.main.fragment\_cart\_empty.view.\* |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| /\*\* |
|  |

|  |
| --- |
| \* Фрагмент пустой корзины |
|  |

|  |
| --- |
| \*/ |
|  |

|  |
| --- |
| class CartEmptyFragment : Fragment() { |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| override fun onCreateView( |
|  |

|  |
| --- |
| inflater: LayoutInflater, container: ViewGroup?, |
|  |

|  |
| --- |
| savedInstanceState: Bundle? |
|  |

|  |
| --- |
| ): View? { |
|  |

|  |
| --- |
| val inflate = inflater.inflate(R.layout.fragment\_cart\_empty, container, false) |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| val mainActivity = activity as MainActivity |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| inflate.btnGoToCatalog.setOnClickListener { |
|  |

|  |
| --- |
| mainActivity.bottomNavMenu.selectedItemId = R.id.catalog |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| return inflate |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
|  |
|  |

}

* + 1. CartFragment.kt

|  |
| --- |
| package com.example.dpstuffproviderstore.fragment |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| import android.content.Context |
|  |

|  |
| --- |
| import android.os.Bundle |
|  |

|  |
| --- |
| import android.util.Log |
|  |

|  |
| --- |
| import androidx.fragment.app.Fragment |
|  |

|  |
| --- |
| import android.view.LayoutInflater |
|  |

|  |
| --- |
| import android.view.View |
|  |

|  |
| --- |
| import android.view.ViewGroup |
|  |

|  |
| --- |
| import android.widget.Toast |
|  |

|  |
| --- |
| import androidx.recyclerview.widget.LinearLayoutManager |
|  |

|  |
| --- |
| import com.example.dpstuffproviderstore.MainActivity |
|  |

|  |
| --- |
| import com.example.dpstuffproviderstore.R |
|  |

|  |
| --- |
| import com.example.dpstuffproviderstore.adapter.CartAdapter |
|  |

|  |
| --- |
| import com.example.dpstuffproviderstore.adapter.ProductAdapter |
|  |

|  |
| --- |
| import kotlinx.android.synthetic.main.fragment\_cart.view.\* |
|  |

|  |
| --- |
| import kotlinx.android.synthetic.main.fragment\_products.view.\* |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| /\*\* |
|  |

|  |
| --- |
| \* Фрагмент корзины, корзина заполняется по списку товаров cartList из MainActivity |
|  |

|  |
| --- |
| \*/ |
|  |

|  |
| --- |
| class CartFragment : Fragment() { |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| var inflate : View? = null |
|  |

|  |
| --- |
| var summ : Double = 0.0 |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| override fun onCreateView(inflater: LayoutInflater, container: ViewGroup?, |
|  |

|  |
| --- |
| savedInstanceState: Bundle?): View? { |
|  |

|  |
| --- |
| inflate = inflater.inflate(R.layout.fragment\_cart, container, false) |
|  |

|  |
| --- |
| val mainActivity = activity as MainActivity |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| inflate!!.recyclerCart.layoutManager = LinearLayoutManager(context!!) |
|  |

|  |
| --- |
| inflate!!.recyclerCart.adapter = CartAdapter(mainActivity.cartList, this@CartFragment) |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| inflate!!.tvTitle.text = "${mainActivity.cartList.size} Товара:" |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| summ = 0.0 |
|  |

|  |
| --- |
| for(i in mainActivity.cartList){ |
|  |

|  |
| --- |
| summ += i.price!! \* (i.quantity ?: 1) |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
| inflate!!.tvCheckout.text = "Итого: ${summ} ₽" |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| inflate!!.btnCheckout.setOnClickListener { |
|  |

|  |
| --- |
| if(mainActivity.getSharedPreferences("sp", Context.MODE\_PRIVATE).getBoolean("isLogin", false)){ |
|  |

|  |
| --- |
| val fragment = CheckoutFragment() |
|  |

|  |
| --- |
| fragment.checkSumm = summ |
|  |

|  |
| --- |
| mainActivity.makeCurrentFragment(fragment) |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
| else{ |
|  |

|  |
| --- |
| Toast.makeText(requireContext(), "Вы не авторизованы", Toast.LENGTH\_LONG).show() |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| return inflate |
|  |

|  |
| --- |
| } |
|  |

}

* + 1. CatalogFragment.kt

|  |
| --- |
| package com.example.dpstuffproviderstore.fragment |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| import android.opengl.Visibility |
|  |

|  |
| --- |
| import android.os.Bundle |
|  |

|  |
| --- |
| import android.util.Log |
|  |

|  |
| --- |
| import androidx.fragment.app.Fragment |
|  |

|  |
| --- |
| import android.view.LayoutInflater |
|  |

|  |
| --- |
| import android.view.View |
|  |

|  |
| --- |
| import android.view.ViewGroup |
|  |

|  |
| --- |
| import android.view.inputmethod.EditorInfo |
|  |

|  |
| --- |
| import android.widget.Toast |
|  |

|  |
| --- |
| import androidx.recyclerview.widget.LinearLayoutManager |
|  |

|  |
| --- |
| import com.example.dpstuffproviderstore.MainActivity |
|  |

|  |
| --- |
| import com.example.dpstuffproviderstore.R |
|  |

|  |
| --- |
| import com.example.dpstuffproviderstore.`interface`.ICategory |
|  |

|  |
| --- |
| import com.example.dpstuffproviderstore.adapter.CategoryAdapter |
|  |

|  |
| --- |
| import com.example.dpstuffproviderstore.models.CategoryData |
|  |

|  |
| --- |
| import com.example.dpstuffproviderstore.other.ClientApiService |
|  |

|  |
| --- |
| import kotlinx.android.synthetic.main.fragment\_catalog.\* |
|  |

|  |
| --- |
| import kotlinx.android.synthetic.main.fragment\_catalog.view.\* |
|  |

|  |
| --- |
| import kotlinx.android.synthetic.main.fragment\_error.view.\* |
|  |

|  |
| --- |
| import kotlinx.coroutines.GlobalScope |
|  |

|  |
| --- |
| import kotlinx.coroutines.launch |
|  |

|  |
| --- |
| import retrofit2.Call |
|  |

|  |
| --- |
| import retrofit2.Callback |
|  |

|  |
| --- |
| import retrofit2.Response |
|  |

|  |
| --- |
| import retrofit2.Retrofit |
|  |

|  |
| --- |
| import retrofit2.converter.gson.GsonConverterFactory |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| /\*\* |
|  |

|  |
| --- |
| \* Фрагмент каталога с поиском товаров по наименованию и категориям |
|  |

|  |
| --- |
| \*/ |
|  |

|  |
| --- |
| class CatalogFragment() : Fragment() { |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| var inflate : View? = null |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| override fun onCreateView( |
|  |

|  |
| --- |
| inflater: LayoutInflater, container: ViewGroup?, |
|  |

|  |
| --- |
| savedInstanceState: Bundle? |
|  |

|  |
| --- |
| ): View? { |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| inflate = inflater.inflate(R.layout.fragment\_catalog, container, false) |
|  |

|  |
| --- |
| val mainActivity = activity as MainActivity |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| GlobalScope.launch { |
|  |

|  |
| --- |
| inflate!!.recyclerCategory.layoutManager = LinearLayoutManager(context!!) |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| ClientApiService().getMainCategories() { |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| if(it != null){ |
|  |

|  |
| --- |
| inflate!!.recyclerCategory.adapter = CategoryAdapter(it, this@CatalogFragment, null) |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
| else{ |
|  |

|  |
| --- |
| val fragment = ErrorFragment() |
|  |

|  |
| --- |
| fragment.statusCode = "404" |
|  |

|  |
| --- |
| mainActivity.makeCurrentFragment(fragment) |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| inflate!!.inputTextSearch.setOnEditorActionListener { v, actionId, event -> |
|  |

|  |
| --- |
| if(actionId == EditorInfo.IME\_ACTION\_SEARCH){ |
|  |

|  |
| --- |
| Log.i("myLog", "Пошёл поиск...") |
|  |

|  |
| --- |
| mainActivity.modeSearch = MainActivity.EnumModeSearch.NAME |
|  |

|  |
| --- |
| mainActivity.productName = inflate!!.inputTextSearch.text.toString() |
|  |

|  |
| --- |
| mainActivity.makeCurrentFragment(ProductsFragment()) |
|  |

|  |
| --- |
| true |
|  |

|  |
| --- |
| } else { |
|  |

|  |
| --- |
| false |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| return inflate |
|  |

|  |
| --- |
| } |
|  |

}

* + 1. CheckoutFragment.kt

|  |
| --- |
| package com.example.dpstuffproviderstore.fragment |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| import android.content.Context |
|  |

|  |
| --- |
| import android.os.Bundle |
|  |

|  |
| --- |
| import android.util.Log |
|  |

|  |
| --- |
| import androidx.fragment.app.Fragment |
|  |

|  |
| --- |
| import android.view.LayoutInflater |
|  |

|  |
| --- |
| import android.view.View |
|  |

|  |
| --- |
| import android.view.ViewGroup |
|  |

|  |
| --- |
| import android.widget.Toast |
|  |

|  |
| --- |
| import com.example.dpstuffproviderstore.MainActivity |
|  |

|  |
| --- |
| import com.example.dpstuffproviderstore.R |
|  |

|  |
| --- |
| import com.example.dpstuffproviderstore.models.OrderData |
|  |

|  |
| --- |
| import com.example.dpstuffproviderstore.models.ProductData |
|  |

|  |
| --- |
| import com.example.dpstuffproviderstore.other.ClientApiService |
|  |

|  |
| --- |
| import com.google.gson.Gson |
|  |

|  |
| --- |
| import kotlinx.android.synthetic.main.fragment\_checkout.\* |
|  |

|  |
| --- |
| import kotlinx.android.synthetic.main.fragment\_checkout.view.\* |
|  |

|  |
| --- |
| import java.util.\* |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| /\*\* |
|  |

|  |
| --- |
| \* Фрагмент оформления нового заказа |
|  |

|  |
| --- |
| \*/ |
|  |

|  |
| --- |
| class CheckoutFragment : Fragment() { |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| var inflate : View? = null |
|  |

|  |
| --- |
| var checkSumm : Double = 0.0 |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| override fun onCreateView( |
|  |

|  |
| --- |
| inflater: LayoutInflater, container: ViewGroup?, |
|  |

|  |
| --- |
| savedInstanceState: Bundle? |
|  |

|  |
| --- |
| ): View? { |
|  |

|  |
| --- |
| inflate = inflater.inflate(R.layout.fragment\_checkout, container, false) |
|  |

|  |
| --- |
| val mainActivity = activity as MainActivity |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| //inflate!!.dateDelivery.minDate = java.util.Calendar.getInstance() |
|  |

|  |
| --- |
| inflate!!.tvCheckoutConfirm.text = "Итого: ${checkSumm} ₽" |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| var c = Calendar.getInstance() |
|  |

|  |
| --- |
| c.add(Calendar.DAY\_OF\_MONTH, 1) |
|  |

|  |
| --- |
| inflate!!.dateDelivery.minDate = c.timeInMillis |
|  |

|  |
| --- |
| c.add(Calendar.DAY\_OF\_MONTH, 14) |
|  |

|  |
| --- |
| inflate!!.dateDelivery.maxDate = c.timeInMillis |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| var frontDoor : Int? = null |
|  |

|  |
| --- |
| var apartNum : Int? = null |
|  |

|  |
| --- |
| var floorNum : Int? = null |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| if(!inflate!!.inputTextFrontDoorCheckout.text.isNullOrEmpty()){ |
|  |

|  |
| --- |
| frontDoor = inflate!!.inputTextFrontDoorCheckout.text.toString().toInt() |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| if(!inflate!!.inputTextApartCheckout.text.isNullOrEmpty()){ |
|  |

|  |
| --- |
| apartNum = inflate!!.inputTextApartCheckout.text.toString().toInt() |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| if(!inflate!!.inputTextFloorCheckout.text.isNullOrEmpty()){ |
|  |

|  |
| --- |
| floorNum = inflate!!.inputTextFloorCheckout.text.toString().toInt() |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| inflate!!.btnCheckoutConfirm.setOnClickListener { |
|  |

|  |
| --- |
| if(!validForm()){ |
|  |

|  |
| --- |
| return@setOnClickListener |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| val newOrder = OrderData( |
|  |

|  |
| --- |
| id = null, |
|  |

|  |
| --- |
| address = inflate!!.inputTextAddressCheckout.text.toString(), |
|  |

|  |
| --- |
| lastName = mainActivity.client!!.lastName, |
|  |

|  |
| --- |
| firstName = mainActivity.client!!.firstName, |
|  |

|  |
| --- |
| phone = mainActivity.client!!.phone, |
|  |

|  |
| --- |
| frontDoor = frontDoor, |
|  |

|  |
| --- |
| apartNum = apartNum, |
|  |

|  |
| --- |
| floorNum = floorNum, |
|  |

|  |
| --- |
| intercom = inflate!!.inputTextIntercomCheckout.text.toString(), |
|  |

|  |
| --- |
| deliveryDate = "${inflate!!.dateDelivery.dayOfMonth}.${inflate!!.dateDelivery.month + 1}.${inflate!!.dateDelivery.year}", |
|  |

|  |
| --- |
| timeFrom = inflate!!.spinnerTimeFrom.selectedItem.toString(), |
|  |

|  |
| --- |
| timeTo = inflate!!.spinnerTimeTo.selectedItem.toString(), |
|  |

|  |
| --- |
| commentary = inflate!!.inputTextCommentaryCheckout.text.toString(), |
|  |

|  |
| --- |
| summ = checkSumm, |
|  |

|  |
| --- |
| status = null, |
|  |

|  |
| --- |
| idCourier = null, |
|  |

|  |
| --- |
| priority = 5, |
|  |

|  |
| --- |
| codeToFinish = ((1000..9999).random().toString()), |
|  |

|  |
| --- |
| listProducts = mainActivity.cartList |
|  |

|  |
| --- |
| ) |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| Log.i("myLog", "checkout: ${Gson().toJson(newOrder.listProducts)}") |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| ClientApiService().addOrder(newOrder){ |
|  |

|  |
| --- |
| if(it == null){ |
|  |

|  |
| --- |
| Toast.makeText(requireContext(), "Ошибка, повторите позже", Toast.LENGTH\_LONG).show() |
|  |

|  |
| --- |
| mainActivity.makeCurrentFragment(CartFragment()) |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
| else{ |
|  |

|  |
| --- |
| Toast.makeText(requireContext(), "Заказ успешно оформлен", Toast.LENGTH\_LONG).show() |
|  |

|  |
| --- |
| mainActivity.cartList = arrayListOf() |
|  |

|  |
| --- |
| mainActivity.cartFragment = CartEmptyFragment() |
|  |

|  |
| --- |
| mainActivity.getSharedPreferences("sp", Context.MODE\_PRIVATE).edit().putString("cartList", Gson().toJson(mainActivity.cartList)).apply() |
|  |

|  |
| --- |
| mainActivity.makeCurrentFragment(CartEmptyFragment()) |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| return inflate |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| fun validForm(): Boolean{ |
|  |

|  |
| --- |
| if(inflate!!.inputTextAddressCheckout.text.isNullOrEmpty()){ |
|  |

|  |
| --- |
| Toast.makeText(requireContext(), "Поле адреса обязательно для заполнения", Toast.LENGTH\_LONG).show() |
|  |

|  |
| --- |
| return false |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| if(inflate!!.spinnerTimeFrom.selectedItemId >= inflate!!.spinnerTimeTo.selectedItemId){ |
|  |

|  |
| --- |
| Toast.makeText(requireContext(), "Некорректное время доставки", Toast.LENGTH\_LONG).show() |
|  |

|  |
| --- |
| return false |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| return true |
|  |

|  |
| --- |
| } |
|  |

}

* + 1. ErrorFragment.kt

|  |
| --- |
| package com.example.dpstuffproviderstore.fragment |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| import android.os.Bundle |
|  |

|  |
| --- |
| import androidx.fragment.app.Fragment |
|  |

|  |
| --- |
| import android.view.LayoutInflater |
|  |

|  |
| --- |
| import android.view.View |
|  |

|  |
| --- |
| import android.view.ViewGroup |
|  |

|  |
| --- |
| import com.example.dpstuffproviderstore.R |
|  |

|  |
| --- |
| import kotlinx.android.synthetic.main.fragment\_error.view.\* |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| /\*\* |
|  |

|  |
| --- |
| \* Фрагмент заглушки на случай ошибок (не отвечает api) |
|  |

|  |
| --- |
| \*/ |
|  |

|  |
| --- |
| class ErrorFragment : Fragment() { |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| var statusCode : String = "404" |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| override fun onCreateView(inflater: LayoutInflater, container: ViewGroup?, |
|  |

|  |
| --- |
| savedInstanceState: Bundle?): View? { |
|  |

|  |
| --- |
| val inflate = inflater.inflate(R.layout.fragment\_error, container, false) |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| inflate.tvStatusCode.text = statusCode |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| return inflate |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
|  |
|  |

}

* + 1. HomeFragment.kt

|  |
| --- |
| package com.example.dpstuffproviderstore.fragment |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| import android.os.Bundle |
|  |

|  |
| --- |
| import android.util.Log |
|  |

|  |
| --- |
| import androidx.fragment.app.Fragment |
|  |

|  |
| --- |
| import android.view.LayoutInflater |
|  |

|  |
| --- |
| import android.view.View |
|  |

|  |
| --- |
| import android.view.ViewGroup |
|  |

|  |
| --- |
| import android.widget.Toast |
|  |

|  |
| --- |
| import androidx.recyclerview.widget.LinearLayoutManager |
|  |

|  |
| --- |
| import com.example.dpstuffproviderstore.MainActivity |
|  |

|  |
| --- |
| import com.example.dpstuffproviderstore.R |
|  |

|  |
| --- |
| import com.example.dpstuffproviderstore.R.string |
|  |

|  |
| --- |
| import com.example.dpstuffproviderstore.`interface`.IProduct |
|  |

|  |
| --- |
| import com.example.dpstuffproviderstore.adapter.ProductAdapter |
|  |

|  |
| --- |
| import com.example.dpstuffproviderstore.models.ProductData |
|  |

|  |
| --- |
| import com.example.dpstuffproviderstore.other.ClientApiService |
|  |

|  |
| --- |
| import kotlinx.android.synthetic.main.fragment\_home.view.\* |
|  |

|  |
| --- |
| import kotlinx.android.synthetic.main.fragment\_products.view.\* |
|  |

|  |
| --- |
| import retrofit2.Call |
|  |

|  |
| --- |
| import retrofit2.Callback |
|  |

|  |
| --- |
| import retrofit2.Response |
|  |

|  |
| --- |
| import retrofit2.Retrofit |
|  |

|  |
| --- |
| import retrofit2.converter.gson.GsonConverterFactory |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| /\*\* |
|  |

|  |
| --- |
| \* Фрагмент домашней страницы со списком популярных товаров (пока просто выводит все товары) |
|  |

|  |
| --- |
| \*/ |
|  |

|  |
| --- |
| class HomeFragment : Fragment() { |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| override fun onCreateView( |
|  |

|  |
| --- |
| inflater: LayoutInflater, container: ViewGroup?, |
|  |

|  |
| --- |
| savedInstanceState: Bundle? |
|  |

|  |
| --- |
| ): View? { |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| val inflate = inflater.inflate(R.layout.fragment\_home, container, false) |
|  |

|  |
| --- |
| val mainActivity = activity as MainActivity |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| inflate.recyclerBestProducts.layoutManager = LinearLayoutManager(context!!) |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| ClientApiService().getAllProducts() { |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| if(it != null){ |
|  |

|  |
| --- |
| inflate.recyclerBestProducts.adapter = ProductAdapter(it, this@HomeFragment) |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
| else{ |
|  |

|  |
| --- |
| val fragment = ErrorFragment() |
|  |

|  |
| --- |
| fragment.statusCode = "404" |
|  |

|  |
| --- |
| mainActivity.makeCurrentFragment(fragment) |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| return inflate |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
|  |
|  |

}

* + 1. OrdersFragment.kt

|  |
| --- |
| package com.example.dpstuffproviderstore.fragment |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| import android.content.Context |
|  |

|  |
| --- |
| import android.opengl.Visibility |
|  |

|  |
| --- |
| import android.os.Bundle |
|  |

|  |
| --- |
| import android.util.Log |
|  |

|  |
| --- |
| import androidx.fragment.app.Fragment |
|  |

|  |
| --- |
| import android.view.LayoutInflater |
|  |

|  |
| --- |
| import android.view.View |
|  |

|  |
| --- |
| import android.view.ViewGroup |
|  |

|  |
| --- |
| import androidx.recyclerview.widget.LinearLayoutManager |
|  |

|  |
| --- |
| import com.example.dpstuffproviderstore.MainActivity |
|  |

|  |
| --- |
| import com.example.dpstuffproviderstore.R |
|  |

|  |
| --- |
| import com.example.dpstuffproviderstore.adapter.CartAdapter |
|  |

|  |
| --- |
| import com.example.dpstuffproviderstore.adapter.OrderAdapter |
|  |

|  |
| --- |
| import com.example.dpstuffproviderstore.other.ClientApiService |
|  |

|  |
| --- |
| import com.example.dpstuffproviderstore.other.ServiceBuilder |
|  |

|  |
| --- |
| import kotlinx.android.synthetic.main.fragment\_cart.view.\* |
|  |

|  |
| --- |
| import kotlinx.android.synthetic.main.fragment\_orders.view.\* |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| /\*\* |
|  |

|  |
| --- |
| \* Фрагмент со списком заказов пользователя |
|  |

|  |
| --- |
| \*/ |
|  |

|  |
| --- |
| class OrdersFragment : Fragment() { |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| override fun onCreateView(inflater: LayoutInflater, container: ViewGroup?, |
|  |

|  |
| --- |
| savedInstanceState: Bundle?): View? { |
|  |

|  |
| --- |
| val inflate = inflater.inflate(R.layout.fragment\_orders, container, false) |
|  |

|  |
| --- |
| val mainActivity = activity as MainActivity |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| inflate!!.recyclerOrders.layoutManager = LinearLayoutManager(context!!) |
|  |

|  |
| --- |
| ClientApiService().getOrderByClient(mainActivity.client!!.login!!, mainActivity.getSharedPreferences("sp", Context.MODE\_PRIVATE).getString("password", "")!!){ |
|  |

|  |
| --- |
| if(it.isNullOrEmpty()){ |
|  |

|  |
| --- |
| inflate!!.hintEmpty.visibility = View.VISIBLE |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
| else{ |
|  |

|  |
| --- |
| inflate!!.recyclerOrders.adapter = OrderAdapter(it, this@OrdersFragment) |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| return inflate |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
|  |
|  |

}

* + 1. ProductsFragment.kt

|  |
| --- |
| package com.example.dpstuffproviderstore.fragment |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| import android.os.Bundle |
|  |

|  |
| --- |
| import androidx.fragment.app.Fragment |
|  |

|  |
| --- |
| import android.view.LayoutInflater |
|  |

|  |
| --- |
| import android.view.View |
|  |

|  |
| --- |
| import android.view.ViewGroup |
|  |

|  |
| --- |
| import android.widget.Toast |
|  |

|  |
| --- |
| import androidx.recyclerview.widget.LinearLayoutManager |
|  |

|  |
| --- |
| import com.example.dpstuffproviderstore.MainActivity |
|  |

|  |
| --- |
| import com.example.dpstuffproviderstore.R |
|  |

|  |
| --- |
| import com.example.dpstuffproviderstore.`interface`.ICategory |
|  |

|  |
| --- |
| import com.example.dpstuffproviderstore.`interface`.IProduct |
|  |

|  |
| --- |
| import com.example.dpstuffproviderstore.adapter.CategoryAdapter |
|  |

|  |
| --- |
| import com.example.dpstuffproviderstore.adapter.ProductAdapter |
|  |

|  |
| --- |
| import com.example.dpstuffproviderstore.models.CategoryData |
|  |

|  |
| --- |
| import com.example.dpstuffproviderstore.models.ProductData |
|  |

|  |
| --- |
| import com.example.dpstuffproviderstore.other.ClientApiService |
|  |

|  |
| --- |
| import kotlinx.android.synthetic.main.fragment\_catalog.view.\* |
|  |

|  |
| --- |
| import kotlinx.android.synthetic.main.fragment\_products.view.\* |
|  |

|  |
| --- |
| import retrofit2.Call |
|  |

|  |
| --- |
| import retrofit2.Callback |
|  |

|  |
| --- |
| import retrofit2.Response |
|  |

|  |
| --- |
| import retrofit2.Retrofit |
|  |

|  |
| --- |
| import retrofit2.converter.gson.GsonConverterFactory |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| /\*\* |
|  |

|  |
| --- |
| \* Фрагмент со списков товаров (по наименованию или категории) |
|  |

|  |
| --- |
| \*/ |
|  |

|  |
| --- |
| class ProductsFragment : Fragment() { |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| var inflate : View? = null |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| override fun onCreateView(inflater: LayoutInflater, container: ViewGroup?, |
|  |

|  |
| --- |
| savedInstanceState: Bundle?): View? { |
|  |

|  |
| --- |
| inflate = inflater.inflate(R.layout.fragment\_products, container, false) |
|  |

|  |
| --- |
| val mainActivity = activity as MainActivity |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| inflate!!.recyclerProducts.layoutManager = LinearLayoutManager(context!!) |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| if(mainActivity.modeSearch == MainActivity.EnumModeSearch.CATEGORY) |
|  |

|  |
| --- |
| { |
|  |

|  |
| --- |
| ClientApiService().getProductsByCategory(mainActivity.productCategory) { |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| if(it != null){ |
|  |

|  |
| --- |
| inflate!!.recyclerProducts.adapter = ProductAdapter(it, this@ProductsFragment) |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
| else{ |
|  |

|  |
| --- |
| val fragment = ErrorFragment() |
|  |

|  |
| --- |
| fragment.statusCode = "404" |
|  |

|  |
| --- |
| mainActivity.makeCurrentFragment(fragment) |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| if(mainActivity.modeSearch == MainActivity.EnumModeSearch.NAME) |
|  |

|  |
| --- |
| { |
|  |

|  |
| --- |
| ClientApiService().getProductsByName(mainActivity.productName) { |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| if(it != null){ |
|  |

|  |
| --- |
| inflate!!.recyclerProducts.adapter = ProductAdapter(it, this@ProductsFragment) |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
| else{ |
|  |

|  |
| --- |
| val fragment = ErrorFragment() |
|  |

|  |
| --- |
| fragment.statusCode = "404" |
|  |

|  |
| --- |
| mainActivity.makeCurrentFragment(fragment) |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| return inflate |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
|  |
|  |

}

* + 1. RegistrationFragment.kt

|  |
| --- |
| package com.example.dpstuffproviderstore.fragment |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| import android.os.Bundle |
|  |

|  |
| --- |
| import android.util.Log |
|  |

|  |
| --- |
| import androidx.fragment.app.Fragment |
|  |

|  |
| --- |
| import android.view.LayoutInflater |
|  |

|  |
| --- |
| import android.view.View |
|  |

|  |
| --- |
| import android.view.ViewGroup |
|  |

|  |
| --- |
| import android.widget.Toast |
|  |

|  |
| --- |
| import com.example.dpstuffproviderstore.MainActivity |
|  |

|  |
| --- |
| import com.example.dpstuffproviderstore.R |
|  |

|  |
| --- |
| import com.example.dpstuffproviderstore.models.ClientData |
|  |

|  |
| --- |
| import com.example.dpstuffproviderstore.other.ClientApiService |
|  |

|  |
| --- |
| import kotlinx.android.synthetic.main.fragment\_registration.\* |
|  |

|  |
| --- |
| import kotlinx.android.synthetic.main.fragment\_registration.view.\* |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| /\*\* |
|  |

|  |
| --- |
| \* Фрагмент регистрации нового пользователя |
|  |

|  |
| --- |
| \*/ |
|  |

|  |
| --- |
| class RegistrationFragment : Fragment() { |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| override fun onCreateView( |
|  |

|  |
| --- |
| inflater: LayoutInflater, container: ViewGroup?, |
|  |

|  |
| --- |
| savedInstanceState: Bundle? |
|  |

|  |
| --- |
| ): View? { |
|  |

|  |
| --- |
| val inflate = inflater.inflate(R.layout.fragment\_registration, container, false) |
|  |

|  |
| --- |
| val mainActivity = activity as MainActivity |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| inflate.btnReg.setOnClickListener { |
|  |

|  |
| --- |
| if(!validForm()){ |
|  |

|  |
| --- |
| return@setOnClickListener Unit |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
| val client = ClientData( |
|  |

|  |
| --- |
| id = null, |
|  |

|  |
| --- |
| lastName = inputTextLastName.text.toString(), |
|  |

|  |
| --- |
| firstName = inputTextFirstName.text.toString(), |
|  |

|  |
| --- |
| patronymic = inputTextPatronymic.text.toString(), |
|  |

|  |
| --- |
| email = inputTextEmail.text.toString(), |
|  |

|  |
| --- |
| phone = inputTextPhone.text.toString(), |
|  |

|  |
| --- |
| login = inputTextLogin.text.toString(), |
|  |

|  |
| --- |
| password = null |
|  |

|  |
| --- |
| ) |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| ClientApiService().addClient(client, inputTextPassword.text.toString()){ |
|  |

|  |
| --- |
| if(it != null){ |
|  |

|  |
| --- |
| Toast.makeText(context!!, "Регистрация успешна", Toast.LENGTH\_LONG).show() |
|  |

|  |
| --- |
| mainActivity.makeCurrentFragment(AccountNotLoginFragment()) |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
| else{ |
|  |

|  |
| --- |
| Toast.makeText(context!!, "Что-то пошло не по плану, повторите позже", Toast.LENGTH\_LONG).show() |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| return inflate |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| fun validForm() : Boolean{ |
|  |

|  |
| --- |
| if(inputTextFirstName.text.isNullOrEmpty() || |
|  |

|  |
| --- |
| inputTextLastName.text.isNullOrEmpty() || |
|  |

|  |
| --- |
| inputTextPatronymic.text.isNullOrEmpty() || |
|  |

|  |
| --- |
| inputTextEmail.text.isNullOrEmpty() || |
|  |

|  |
| --- |
| inputTextPhone.text.isNullOrEmpty() || |
|  |

|  |
| --- |
| inputTextLogin.text.isNullOrEmpty() || |
|  |

|  |
| --- |
| inputTextPassword.text.isNullOrEmpty()){ |
|  |

|  |
| --- |
| Toast.makeText(context!!, "Заполните все поля", Toast.LENGTH\_LONG).show() |
|  |

|  |
| --- |
| return false |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| if(!inputTextEmail.text!!.contains('@') || !inputTextEmail.text!!.contains('.')){ |
|  |

|  |
| --- |
| Toast.makeText(context!!, "Некорректный адрес почты", Toast.LENGTH\_LONG).show() |
|  |

|  |
| --- |
| return false |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| if(inputTextPassword.text!!.length < 8){ |
|  |

|  |
| --- |
| Toast.makeText(context!!, "Пароль должени быть не короче 8-ми символов", Toast.LENGTH\_LONG).show() |
|  |

|  |
| --- |
| return false |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| if(inputTextPassword.text.toString() != inputTextRepeatPassword.text.toString()){ |
|  |

|  |
| --- |
| Toast.makeText(context!!, "Пароли не совпадают", Toast.LENGTH\_LONG).show() |
|  |

|  |
| --- |
| return false |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| return true |
|  |

|  |
| --- |
| } |
|  |

}

* + 1. ClientApiService.kt

|  |
| --- |
| package com.example.dpstuffproviderstore.other |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| import android.util.Log |
|  |

|  |
| --- |
| import com.example.dpstuffproviderstore.MainActivity |
|  |

|  |
| --- |
| import com.example.dpstuffproviderstore.`interface`.ICategory |
|  |

|  |
| --- |
| import com.example.dpstuffproviderstore.`interface`.IClient |
|  |

|  |
| --- |
| import com.example.dpstuffproviderstore.`interface`.IOrder |
|  |

|  |
| --- |
| import com.example.dpstuffproviderstore.`interface`.IProduct |
|  |

|  |
| --- |
| import com.example.dpstuffproviderstore.adapter.ProductAdapter |
|  |

|  |
| --- |
| import com.example.dpstuffproviderstore.fragment.ErrorFragment |
|  |

|  |
| --- |
| import com.example.dpstuffproviderstore.models.\* |
|  |

|  |
| --- |
| import kotlinx.android.synthetic.main.fragment\_home.view.\* |
|  |

|  |
| --- |
| import retrofit2.Call |
|  |

|  |
| --- |
| import retrofit2.Callback |
|  |

|  |
| --- |
| import retrofit2.Response |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| /\*\* |
|  |

|  |
| --- |
| \* Класс, реализующий все методы Api-интерфейсов |
|  |

|  |
| --- |
| \*/ |
|  |

|  |
| --- |
| class ClientApiService { |
|  |

|  |
| --- |
| /\*\* |
|  |

|  |
| --- |
| \* Получаем объект клиента по логину и паролю (авторизация) |
|  |

|  |
| --- |
| \*/ |
|  |

|  |
| --- |
| fun getClient(login: String, password: String, onResult: (ClientData?) -> Unit){ |
|  |

|  |
| --- |
| val api = ServiceBuilder.buildService(IClient::class.java) |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| api.GetClient(login, password).enqueue(object : Callback<List<ClientData>> { |
|  |

|  |
| --- |
| override fun onResponse(call: Call<List<ClientData>>, |
|  |

|  |
| --- |
| response: Response<List<ClientData>> |
|  |

|  |
| --- |
| ) { |
|  |

|  |
| --- |
| if(response.code() == 200){ |
|  |

|  |
| --- |
| onResult(response.body()!![0]) |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
| else{ |
|  |

|  |
| --- |
| Log.i("myLog", "${response.code()}; ${response.errorBody()}; ${response.message()};") |
|  |

|  |
| --- |
| onResult(null) |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| override fun onFailure(call: Call<List<ClientData>>, t: Throwable){ |
|  |

|  |
| --- |
| Log.i("myLog", "Api Failure") |
|  |

|  |
| --- |
| onResult(null) |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
| }) |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| /\*\* |
|  |

|  |
| --- |
| \* Регистрация нового пользователя |
|  |

|  |
| --- |
| \*/ |
|  |

|  |
| --- |
| fun addClient(clientData: ClientData, noHashPassword: String, onResult: (ClientData?) -> Unit){ |
|  |

|  |
| --- |
| val api = ServiceBuilder.buildService(IClient::class.java) |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| api.addUser(clientData, noHashPassword).enqueue(object : Callback<List<ClientData>> { |
|  |

|  |
| --- |
| override fun onResponse(call: Call<List<ClientData>>, |
|  |

|  |
| --- |
| response: Response<List<ClientData>> |
|  |

|  |
| --- |
| ) { |
|  |

|  |
| --- |
| if(response.code() == 200){ |
|  |

|  |
| --- |
| onResult(response.body()!![0]) |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
| else{ |
|  |

|  |
| --- |
| Log.i("myLog", "${response.code()}; ${response.errorBody()}; ${response.message()};") |
|  |

|  |
| --- |
| onResult(null) |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| override fun onFailure(call: Call<List<ClientData>>, t: Throwable){ |
|  |

|  |
| --- |
| Log.i("myLog", "Api Failure") |
|  |

|  |
| --- |
| onResult(null) |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
| }) |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| /\*\* |
|  |

|  |
| --- |
| \* Регистрация нового заказа |
|  |

|  |
| --- |
| \*/ |
|  |

|  |
| --- |
| fun addOrder(orderData: OrderData, onResult: (OrderData?) -> Unit){ |
|  |

|  |
| --- |
| val api = ServiceBuilder.buildService(IOrder::class.java) |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| api.addOrder(orderData).enqueue(object : Callback<OrderData> { |
|  |

|  |
| --- |
| override fun onResponse(call: Call<OrderData>, |
|  |

|  |
| --- |
| response: Response<OrderData> |
|  |

|  |
| --- |
| ) { |
|  |

|  |
| --- |
| if(response.code() == 200){ |
|  |

|  |
| --- |
| onResult(response.body()!!) |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
| else{ |
|  |

|  |
| --- |
| Log.i("myLog", "${response.code()}; ${response.errorBody()}; ${response.message()};") |
|  |

|  |
| --- |
| onResult(null) |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| override fun onFailure(call: Call<OrderData>, t: Throwable){ |
|  |

|  |
| --- |
| Log.i("myLog", "Api Failure") |
|  |

|  |
| --- |
| onResult(null) |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
| }) |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| /\*\* |
|  |

|  |
| --- |
| \* Получаем список заказов конкретного клиента (по логину и паролю) |
|  |

|  |
| --- |
| \*/ |
|  |

|  |
| --- |
| fun getOrderByClient(login: String, password: String, onResult: (List<OrderData>?) -> Unit){ |
|  |

|  |
| --- |
| val api = ServiceBuilder.buildService(IOrder::class.java) |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| api.getOrdersByClient(login, password).enqueue(object : Callback<List<OrderData>> { |
|  |

|  |
| --- |
| override fun onResponse(call: Call<List<OrderData>>, |
|  |

|  |
| --- |
| response: Response<List<OrderData>> |
|  |

|  |
| --- |
| ) { |
|  |

|  |
| --- |
| if(response.code() == 200){ |
|  |

|  |
| --- |
| onResult(response.body()!!) |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
| else{ |
|  |

|  |
| --- |
| Log.i("myLog", "${response.code()}; ${response.errorBody()}; ${response.message()};") |
|  |

|  |
| --- |
| onResult(null) |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| override fun onFailure(call: Call<List<OrderData>>, t: Throwable){ |
|  |

|  |
| --- |
| Log.i("myLog", "Api Failure") |
|  |

|  |
| --- |
| onResult(null) |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
| }) |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| /\*\* |
|  |

|  |
| --- |
| \* Получаем первый объект изображения, относящийся к товару (по id товара) |
|  |

|  |
| --- |
| \*/ |
|  |

|  |
| --- |
| fun getImages(productId: Int, onResult: (List<ProductImagesData>?) -> Unit){ |
|  |

|  |
| --- |
| val api = ServiceBuilder.buildService(IProduct::class.java) |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| api.GetImages(productId).enqueue(object : Callback<List<ProductImagesData>> { |
|  |

|  |
| --- |
| override fun onResponse(call: Call<List<ProductImagesData>>, |
|  |

|  |
| --- |
| response: Response<List<ProductImagesData>> |
|  |

|  |
| --- |
| ) { |
|  |

|  |
| --- |
| if(response.code() == 200){ |
|  |

|  |
| --- |
| onResult(response.body()!!) |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
| else{ |
|  |

|  |
| --- |
| Log.i("myLog", "${response.code()}; ${response.errorBody()}; ${response.message()};") |
|  |

|  |
| --- |
| onResult(null) |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| override fun onFailure(call: Call<List<ProductImagesData>>, t: Throwable){ |
|  |

|  |
| --- |
| Log.i("myLog", "Api Failure") |
|  |

|  |
| --- |
| onResult(null) |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
| }) |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| /\*\* |
|  |

|  |
| --- |
| \* Получаем список основных категорий (у которыйх нет родительской категории) |
|  |

|  |
| --- |
| \*/ |
|  |

|  |
| --- |
| fun getMainCategories(onResult: (List<CategoryData>?) -> Unit){ |
|  |

|  |
| --- |
| val api = ServiceBuilder.buildService(ICategory::class.java) |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| api.GetMainCategories().enqueue(object : Callback<List<CategoryData>> { |
|  |

|  |
| --- |
| override fun onResponse(call: Call<List<CategoryData>>, |
|  |

|  |
| --- |
| response: Response<List<CategoryData>> |
|  |

|  |
| --- |
| ) { |
|  |

|  |
| --- |
| if(response.code() == 200){ |
|  |

|  |
| --- |
| onResult(response.body()!!) |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
| else{ |
|  |

|  |
| --- |
| Log.i("myLog", "${response.code()}; ${response.errorBody()}; ${response.message()};") |
|  |

|  |
| --- |
| onResult(null) |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| override fun onFailure(call: Call<List<CategoryData>>, t: Throwable){ |
|  |

|  |
| --- |
| Log.i("myLog", "Api Failure") |
|  |

|  |
| --- |
| onResult(null) |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
| }) |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| /\*\* |
|  |

|  |
| --- |
| \* Получаем список дочерних категорий (по имени родительской категории) |
|  |

|  |
| --- |
| \*/ |
|  |

|  |
| --- |
| fun getChildCategories(categoryName: String, onResult: (List<CategoryData>?) -> Unit){ |
|  |

|  |
| --- |
| val api = ServiceBuilder.buildService(ICategory::class.java) |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| api.GetChildCategories(categoryName).enqueue(object : Callback<List<CategoryData>> { |
|  |

|  |
| --- |
| override fun onResponse(call: Call<List<CategoryData>>, |
|  |

|  |
| --- |
| response: Response<List<CategoryData>> |
|  |

|  |
| --- |
| ) { |
|  |

|  |
| --- |
| if(response.code() == 200){ |
|  |

|  |
| --- |
| onResult(response.body()!!) |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
| else{ |
|  |

|  |
| --- |
| Log.i("myLog", "${response.code()}; ${response.errorBody()}; ${response.message()};") |
|  |

|  |
| --- |
| onResult(null) |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| override fun onFailure(call: Call<List<CategoryData>>, t: Throwable){ |
|  |

|  |
| --- |
| Log.i("myLog", "Api Failure") |
|  |

|  |
| --- |
| onResult(null) |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
| }) |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| /\*\* |
|  |

|  |
| --- |
| \* Получаем список всех товаров в БД |
|  |

|  |
| --- |
| \*/ |
|  |

|  |
| --- |
| fun getAllProducts(onResult: (List<ProductData>?) -> Unit){ |
|  |

|  |
| --- |
| val api = ServiceBuilder.buildService(IProduct::class.java) |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| api.GetAllProducts().enqueue(object : Callback<List<ProductData>> { |
|  |

|  |
| --- |
| override fun onResponse(call: Call<List<ProductData>>, |
|  |

|  |
| --- |
| response: Response<List<ProductData>> |
|  |

|  |
| --- |
| ) { |
|  |

|  |
| --- |
| if(response.code() == 200){ |
|  |

|  |
| --- |
| onResult(response.body()!!) |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
| else{ |
|  |

|  |
| --- |
| Log.i("myLog", "${response.code()}; ${response.errorBody()}; ${response.message()};") |
|  |

|  |
| --- |
| onResult(null) |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| override fun onFailure(call: Call<List<ProductData>>, t: Throwable){ |
|  |

|  |
| --- |
| Log.i("myLog", "Api Failure") |
|  |

|  |
| --- |
| onResult(null) |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
| }) |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| /\*\* |
|  |

|  |
| --- |
| \* Получаем список товаров (по наименованию) |
|  |

|  |
| --- |
| \*/ |
|  |

|  |
| --- |
| fun getProductsByName(productName: String, onResult: (List<ProductData>?) -> Unit){ |
|  |

|  |
| --- |
| val api = ServiceBuilder.buildService(IProduct::class.java) |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| api.GetProductsByName(productName).enqueue(object : Callback<List<ProductData>> { |
|  |

|  |
| --- |
| override fun onResponse(call: Call<List<ProductData>>, |
|  |

|  |
| --- |
| response: Response<List<ProductData>> |
|  |

|  |
| --- |
| ) { |
|  |

|  |
| --- |
| if(response.code() == 200){ |
|  |

|  |
| --- |
| onResult(response.body()!!) |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
| else{ |
|  |

|  |
| --- |
| Log.i("myLog", "${response.code()}; ${response.errorBody()}; ${response.message()};") |
|  |

|  |
| --- |
| onResult(null) |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| override fun onFailure(call: Call<List<ProductData>>, t: Throwable){ |
|  |

|  |
| --- |
| Log.i("myLog", "Api Failure") |
|  |

|  |
| --- |
| onResult(null) |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
| }) |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| /\*\* |
|  |

|  |
| --- |
| \* Получаем список товаров, относящихся к категории и ко всем дочерним категориям (по имени категории) |
|  |

|  |
| --- |
| \*/ |
|  |

|  |
| --- |
| fun getProductsByCategory(categoryName: String, onResult: (List<ProductData>?) -> Unit){ |
|  |

|  |
| --- |
| val api = ServiceBuilder.buildService(IProduct::class.java) |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| api.GetProductsByCategory(categoryName).enqueue(object : Callback<List<ProductData>> { |
|  |

|  |
| --- |
| override fun onResponse(call: Call<List<ProductData>>, |
|  |

|  |
| --- |
| response: Response<List<ProductData>> |
|  |

|  |
| --- |
| ) { |
|  |

|  |
| --- |
| if(response.code() == 200){ |
|  |

|  |
| --- |
| onResult(response.body()!!) |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
| else{ |
|  |

|  |
| --- |
| Log.i("myLog", "${response.code()}; ${response.errorBody()}; ${response.message()};") |
|  |

|  |
| --- |
| onResult(null) |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| override fun onFailure(call: Call<List<ProductData>>, t: Throwable){ |
|  |

|  |
| --- |
| Log.i("myLog", "Api Failure") |
|  |

|  |
| --- |
| onResult(null) |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
| }) |
|  |

|  |
| --- |
| } |
|  |

}

* + 1. MainActivity.kt

|  |
| --- |
| package com.example.dpstuffproviderstore |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| import android.content.Context |
|  |

|  |
| --- |
| import androidx.appcompat.app.AppCompatActivity |
|  |

|  |
| --- |
| import android.os.Bundle |
|  |

|  |
| --- |
| import android.util.Log |
|  |

|  |
| --- |
| import androidx.fragment.app.Fragment |
|  |

|  |
| --- |
| import com.example.dpstuffproviderstore.fragment.\* |
|  |

|  |
| --- |
| import com.example.dpstuffproviderstore.models.ClientData |
|  |

|  |
| --- |
| import com.example.dpstuffproviderstore.models.ProductData |
|  |

|  |
| --- |
| import com.example.dpstuffproviderstore.other.ClientApiService |
|  |

|  |
| --- |
| import com.google.gson.Gson |
|  |

|  |
| --- |
| import com.google.gson.reflect.TypeToken |
|  |

|  |
| --- |
| import kotlinx.android.synthetic.main.activity\_main.\* |
|  |

|  |
| --- |
| import java.lang.reflect.Type |
|  |

|  |
| --- |
| import kotlin.reflect.typeOf |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| class MainActivity : AppCompatActivity() { |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| enum class EnumModeSearch{ CATEGORY, NAME } |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| //val listCategory : List<CategoryData> = listOf(CategoryData(1, "1cat", ""), CategoryData(2, "2cat", ""), CategoryData(3, "3cat", "")) |
|  |

|  |
| --- |
| var productCategory : String = "" |
|  |

|  |
| --- |
| var productName : String = "" |
|  |

|  |
| --- |
| var modeSearch = EnumModeSearch.CATEGORY |
|  |

|  |
| --- |
| var client: ClientData? = null |
|  |

|  |
| --- |
| var cartList: ArrayList<ProductData> = arrayListOf() |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| var accountFragment: Fragment = AccountNotLoginFragment() |
|  |

|  |
| --- |
| var homeFragment: Fragment = HomeFragment() |
|  |

|  |
| --- |
| var catalogFragment: Fragment = CatalogFragment() |
|  |

|  |
| --- |
| var cartFragment: Fragment = CartEmptyFragment() |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| override fun onCreate(savedInstanceState: Bundle?) { |
|  |

|  |
| --- |
| super.onCreate(savedInstanceState) |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| val sharedPreferences = getSharedPreferences("sp", Context.MODE\_PRIVATE) |
|  |

|  |
| --- |
| if(sharedPreferences.getBoolean("isLogin", false)){ |
|  |

|  |
| --- |
| ClientApiService().getClient(sharedPreferences.getString("login", "")!!, sharedPreferences.getString("password", "")!!){ |
|  |

|  |
| --- |
| if (it != null){ |
|  |

|  |
| --- |
| client = it |
|  |

|  |
| --- |
| accountFragment = AccountFragment() |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
| else{ |
|  |

|  |
| --- |
| accountFragment = AccountNotLoginFragment() |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| //sharedPreferences.edit().clear().apply() |
|  |

|  |
| --- |
| var jsonString: String? = sharedPreferences.getString("cartList", "") |
|  |

|  |
| --- |
| if(!jsonString.isNullOrEmpty() && jsonString != "[]"){ |
|  |

|  |
| --- |
| Log.i("myLog", "жсон строка не пустая") |
|  |

|  |
| --- |
| val itemType = object : TypeToken<ArrayList<ProductData>>() {}.type |
|  |

|  |
| --- |
| cartList = Gson().fromJson<ArrayList<ProductData>>(jsonString, itemType) |
|  |

|  |
| --- |
| cartFragment = CartFragment() |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| setContentView(R.layout.activity\_main) |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| makeCurrentFragment(homeFragment) |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| bottomNavMenu.setOnNavigationItemSelectedListener { |
|  |

|  |
| --- |
| when(it.itemId){ |
|  |

|  |
| --- |
| R.id.home -> makeCurrentFragment(homeFragment) |
|  |

|  |
| --- |
| R.id.catalog -> makeCurrentFragment(catalogFragment) |
|  |

|  |
| --- |
| R.id.cart -> makeCurrentFragment(cartFragment) |
|  |

|  |
| --- |
| R.id.account -> makeCurrentFragment(accountFragment!!) |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
| true |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| public fun makeCurrentFragment(fragment: Fragment) = |
|  |

|  |
| --- |
| supportFragmentManager.beginTransaction().apply { |
|  |

|  |
| --- |
| replace(R.id.frameWrapper, fragment, "activeFragment") |
|  |

|  |
| --- |
| commit() |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |
|  |

}