Министерство образования и молодежной политики Свердловской области



ГАПОУ СО «Екатеринбургский колледж транспортного строительства»

Отчёт по программе

«**Практическое занятие 24. Навигация по фрагментам. Работа по вариантам**»

Выполнила: Мезенцева Полина Сергеевна

Группа: ПР-31

Преподаватель: Мирошниченко Г.В

2024 г.

**Вариант 2:**

Используя Recycleview и fragment (навигацию по фрагментам),

создать приложение: «Список доходов по датам»

**Входные данные:**

typeED- EditText, Вид заработка/дохода.

moneyED- EditText, Объем заработка/дохода.

bankED- EditText, Банк прибытия заработка/дохода.

dateED- EditText, Дата зачисления заработка/дохода.

sninner- Spinner, Вид валюты заработка/дохода.

**Выходные данные:** historyList – Список RecycleView, куда сохраняются и откуда выводятся сохраняемые доходы.

**Листинг программы:**

**HistoryItem.kt:**

class HistoryItem(  
 val type: String,  
 val money: String,  
 val bank: String,  
 val date: String  
): Parcelable {  
 constructor(parcel: Parcel) : this(  
 parcel.readString() ?: "",  
 parcel.readString() ?: "",  
 parcel.readString() ?: "",  
 parcel.readString() ?: ""  
 )  
  
 override fun writeToParcel(parcel: Parcel, flags: Int) {  
 parcel.writeString(type)  
 parcel.writeString(money)  
 parcel.writeString(bank)  
 parcel.writeString(date)  
 }  
  
 override fun describeContents(): Int {  
 return 0  
 }  
  
 companion object CREATOR : Parcelable.Creator<HistoryItem> {  
 override fun createFromParcel(parcel: Parcel): HistoryItem {  
 return HistoryItem(parcel)  
 }  
   
 override fun newArray(size: Int): Array<HistoryItem?> {  
 return *arrayOfNulls*(size)  
 }  
 }  
}

**HistoryAdapter.kt:**

@Suppress("DEPRECATION")  
class HistoryAdapter(private val historyList: List<HistoryItem>, private val itemClickListener: (Int) -> Unit) : RecyclerView.Adapter<HistoryAdapter.ViewHolder>() {  
  
 inner class ViewHolder(view: View) : RecyclerView.ViewHolder(view) {  
 val incomeTextView: TextView = itemView.findViewById(R.id.*incomeTextView*)  
 val amountTextView: TextView = itemView.findViewById(R.id.*amountTextView*)  
 val bankTextView: TextView = itemView.findViewById(R.id.*bankTextView*)  
 val dateTextView: TextView = itemView.findViewById(R.id.*dateTextView*)  
 init {  
 itemView.setOnClickListener **{** itemClickListener(*adapterPosition*)  
 **}** }  
 }  
  
 override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): ViewHolder {  
 val view = LayoutInflater.from(parent.*context*).inflate(R.layout.*item\_history*, parent, false)  
 return ViewHolder(view)  
 }  
  
 override fun onBindViewHolder(holder: ViewHolder, position: Int) {  
 val historyItem = historyList[position]  
 holder.incomeTextView.*text* = historyItem.type  
 holder.amountTextView.*text* = historyItem.money  
 holder.bankTextView.*text* = historyItem.bank  
 holder.dateTextView.*text* = historyItem.date  
 }  
 override fun getItemCount(): Int = historyList.size  
}

**MainActivity.kt:**

class MainActivity : AppCompatActivity() {  
 override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
 setContentView(R.layout.*activity\_main*)  
 }  
}

**Fragment1.kt:**

class Fragment1 : Fragment() {  
 private lateinit var spinner: Spinner  
 private lateinit var navController: NavController  
 private lateinit var typeEditText: EditText  
 private lateinit var moneyEditText: EditText  
 private var selectedItem: String = "null"  
   
 override fun onCreateView(  
 inflater: LayoutInflater, container: ViewGroup?,  
 savedInstanceState: Bundle?  
 ): View? {  
 val view = inflater.inflate(R.layout.*fragment\_1*, container, false)  
 spinner = view.findViewById(R.id.*my\_spinner*)  
 typeEditText = view.findViewById(R.id.*type\_salary\_ET*)  
 moneyEditText = view.findViewById(R.id.*money\_ET*)  
   
 val options = *arrayOf*("₽", "$", "€")  
 val adapter = ArrayAdapter(requireContext(), android.R.layout.*simple\_spinner\_item*, options)  
 adapter.setDropDownViewResource(android.R.layout.*simple\_spinner\_dropdown\_item*)  
 spinner.*adapter* = adapter  
   
 spinner.*onItemSelectedListener* = object : AdapterView.OnItemSelectedListener {  
 override fun onItemSelected(  
 parent: AdapterView<\*>,  
 view: View,  
 position: Int,  
 id: Long  
 ) {  
 selectedItem = parent.getItemAtPosition(position).toString()  
 }  
 override fun onNothingSelected(parent: AdapterView<\*>) {  
 Toast.makeText(requireContext(), "Не выбран тип валюты", Toast.*LENGTH\_SHORT*).show()  
 }  
 }  
 view.findViewById<Button>(R.id.*button\_f1*).setOnClickListener **{** val bank = typeEditText.*text*.toString()  
 val money = moneyEditText.*text*.toString()  
 if (bank.*isNotEmpty*() && money.*isNotEmpty*()) {  
 saveData(bank, money, selectedItem)  
 navController.navigate(R.id.*fragment2*)  
 } else {  
 Toast.makeText(  
 requireContext(),  
 "Пожалуйста, заполните все поля",  
 Toast.*LENGTH\_SHORT* ).show()  
 }  
 **}** view.findViewById<Button>(R.id.*button\_f1\_2*).setOnClickListener **{** navController.navigate(R.id.*fragment3*)  
 **}** return view  
 }  
   
 override fun onViewCreated(view: View, savedInstanceState: Bundle?) {  
 super.onViewCreated(view, savedInstanceState)  
 navController = *findNavController*()  
 }  
   
 private fun saveData(type: String, money: String, selectedItem: String) {  
 val sharedPreferences = requireActivity().getSharedPreferences("history\_list", Context.*MODE\_PRIVATE*)  
 val editor = sharedPreferences.edit()  
 editor.putString("type", type)  
 editor.putString("money", money)  
 editor.putString("selectedItem", selectedItem)  
 editor.apply()  
 }  
}

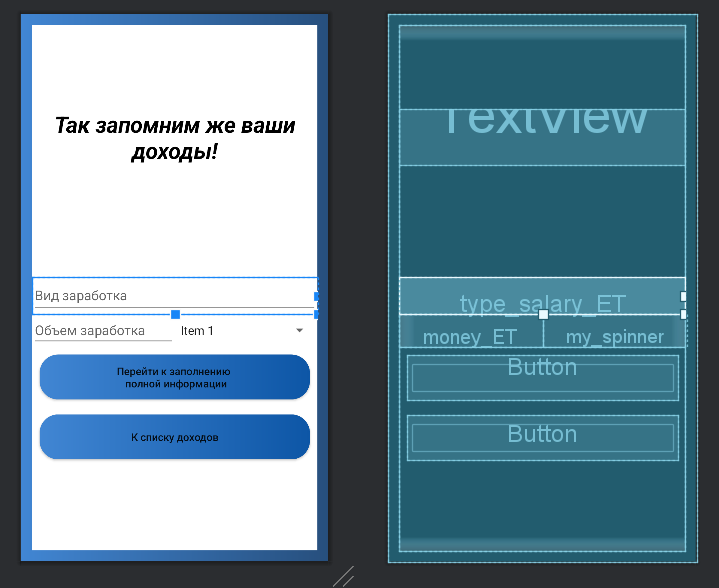
**Fragment2.kt:**

class Fragment2 : Fragment() {  
  
 private lateinit var navController: NavController  
 private lateinit var sharedPreferences: SharedPreferences  
 private lateinit var spinner: Spinner  
 private lateinit var typeEditText: EditText  
 private lateinit var moneyEditText: EditText  
 private lateinit var bankEditText: EditText  
 private lateinit var dateEditText: EditText  
 private var selectedItem: String = "null"  
  
 override fun onCreateView(  
 inflater: LayoutInflater, container: ViewGroup?,  
 savedInstanceState: Bundle?  
 ): View? {  
 val view = inflater.inflate(R.layout.*fragment\_2*, container, false)  
 bankEditText = view.findViewById(R.id.*bankET*)  
 dateEditText = view.findViewById(R.id.*dateET*)  
 typeEditText = view.findViewById(R.id.*type\_salary\_ET*)  
 moneyEditText = view.findViewById(R.id.*money\_ET*)  
 spinner = view.findViewById(R.id.*my\_spinner*)  
   
 val options = *arrayOf*("₽", "$", "€")  
 val adapter = ArrayAdapter(requireContext(), android.R.layout.*simple\_spinner\_item*, options)  
 adapter.setDropDownViewResource(android.R.layout.*simple\_spinner\_dropdown\_item*)  
   
 spinner.*adapter* = adapter  
 spinner.*onItemSelectedListener* = object : AdapterView.OnItemSelectedListener {  
 override fun onItemSelected(  
 parent: AdapterView<\*>,  
 view: View,  
 position: Int,  
 id: Long  
 ) {  
 }  
 override fun onNothingSelected(parent: AdapterView<\*>) {  
 Toast.makeText(requireContext(), "Не выбран тип валюты", Toast.*LENGTH\_SHORT*).show()  
 }  
 }  
 sharedPreferences = requireActivity().getSharedPreferences("history\_list", Context.*MODE\_PRIVATE*)  
 val type = sharedPreferences.getString("type", null)  
 val money = sharedPreferences.getString("money", null)  
   
 selectedItem = sharedPreferences.getString("selectedItem", null).*toString*()  
 selectedItem.*let* **{** val position = options.*indexOf*(**it**)  
 if (position >= 0) {  
 spinner.setSelection(position)  
 }  
 **}** typeEditText.setText(type)  
 moneyEditText.setText(money)  
 return view  
 }  
  
 override fun onViewCreated(view: View, savedInstanceState: Bundle?) {  
 super.onViewCreated(view, savedInstanceState)  
 navController = *findNavController*()  
 view.findViewById<Button>(R.id.*button\_f2*).setOnClickListener **{** val bank = bankEditText.*text*.toString()  
 val date = dateEditText.*text*.toString()  
 if (bank.*isNotEmpty*() && date.*isNotEmpty*()) {  
 if (isValidDate(dateEditText.*text*.toString())){  
 val his = HistoryItem(  
 typeEditText.*text*.toString(),  
 moneyEditText.*text*.toString() + selectedItem,  
 bankEditText.*text*.toString(),  
 dateEditText.*text*.toString()  
 )  
 Fragment3.addHis(requireContext(), his)  
 saveData(bank, date)  
 navController.navigate(R.id.*fragment3*)  
 } else {  
 Toast.makeText(  
 *context*,  
 "Введённая дата не является корректной!",  
 Toast.*LENGTH\_SHORT* ).show()  
 }  
 } else {  
 Toast.makeText(requireContext(), "Пожалуйста, заполните все поля", Toast.*LENGTH\_SHORT*).show()  
 }  
 **}** }  
  
 private fun saveData(bank: String, date: String) {  
 sharedPreferences = requireActivity().getSharedPreferences("history\_list", Context.*MODE\_PRIVATE*)  
 val editor = sharedPreferences.edit()  
 editor.putString("bank", bank)  
 editor.putString("date", date)  
 editor.apply()  
 }  
  
 fun isValidDate(dateString: String, dateFormat: String = "dd.MM.yyyy"): Boolean {  
 val sdf = SimpleDateFormat(dateFormat, Locale.getDefault())  
 sdf.*isLenient* = false  
 return try {  
 sdf.parse(dateString) != null  
 } catch (e: ParseException) {  
 false  
 }  
 }  
}

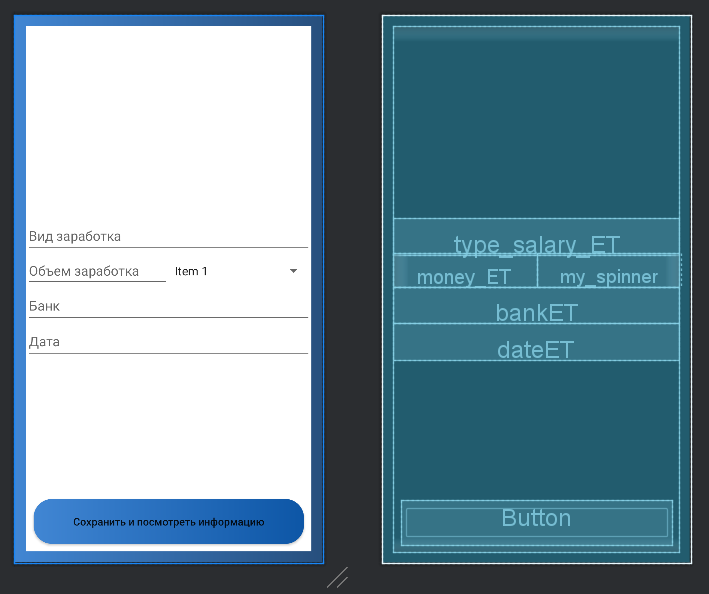
**Fragment3.kt:**

class Fragment3 : Fragment() {  
  
 private lateinit var sharedPreferences: SharedPreferences  
 private var currentPosition: Int = -1  
 private lateinit var navController: NavController  
 private lateinit var recyclerView: RecyclerView  
 private lateinit var adapter: HistoryAdapter  
 private var historyList: MutableList<HistoryItem> = *mutableListOf*()  
 private lateinit var typeEditText2: EditText  
 private lateinit var moneyEditText2: EditText  
 private lateinit var bankEditText2: EditText  
 private lateinit var dateEditText2: EditText  
 private var selectedItem: String = "null"  
 private lateinit var spinner: Spinner  
  
 override fun onCreateView(  
 inflater: LayoutInflater, container: ViewGroup?,  
 savedInstanceState: Bundle?  
 ): View? {  
 val view = inflater.inflate(R.layout.*fragment\_3*, container, false)  
 spinner = view.findViewById(R.id.*my\_spinner*)  
   
 val options = *arrayOf*("₽", "$", "€")  
 val adapter = ArrayAdapter(requireContext(), android.R.layout.*simple\_spinner\_item*, options)  
 adapter.setDropDownViewResource(android.R.layout.*simple\_spinner\_dropdown\_item*)  
  
 spinner.*adapter* = adapter  
 spinner.*onItemSelectedListener* = object : AdapterView.OnItemSelectedListener {  
 override fun onItemSelected(  
 parent: AdapterView<\*>,  
 view: View,  
 position: Int,  
 id: Long  
 ) {  
 selectedItem = parent.getItemAtPosition(position).toString()  
 }  
 override fun onNothingSelected(parent: AdapterView<\*>) {  
 Toast.makeText(requireContext(), "Не выбран тип валюты", Toast.*LENGTH\_SHORT*).show()  
 }  
 }  
 return view  
 }  
  
 override fun onViewCreated(view: View, savedInstanceState: Bundle?) {  
 bankEditText2 = view.findViewById(R.id.*bankET2*)  
 dateEditText2 = view.findViewById(R.id.*dateET2*)  
 typeEditText2 = view.findViewById(R.id.*type\_salary\_ET2*)  
 moneyEditText2 = view.findViewById(R.id.*money\_ET2*)  
 navController = *findNavController*()  
 recyclerView = view.findViewById(R.id.*recyclerView*)  
 recyclerView.*layoutManager* = LinearLayoutManager(requireContext())  
  
 adapter = HistoryAdapter(historyList) **{** position **->** currentPosition = position  
 val selectedItem = historyList[position]  
  
 typeEditText2.setText(selectedItem.type)  
 moneyEditText2.setText(selectedItem.money.*dropLast*(1))  
 bankEditText2.setText(selectedItem.bank)  
 dateEditText2.setText(selectedItem.date)  
 **}** recyclerView.*adapter* = adapter  
 loadHistory()  
   
 view.findViewById<Button>(R.id.*editBut*).setOnClickListener **{** if (currentPosition != -1) {  
 if (isValidDate(dateEditText2.*text*.toString())) {  
 val newItem = HistoryItem(  
 typeEditText2.*text*.toString(),  
 moneyEditText2.*text*.toString() + selectedItem,  
 bankEditText2.*text*.toString(),  
 dateEditText2.*text*.toString()  
 )  
 updateItem(currentPosition, newItem)  
 }  
 else{  
 Toast.makeText(requireContext(),"Неверный формат даты!", Toast.*LENGTH\_SHORT*).show()  
 }  
 }  
 **}** view.findViewById<FloatingActionButton>(R.id.*floatingBut*).setOnClickListener **{** navController.navigate(R.id.*fragment1*)  
 **}** }  
  
  
 @SuppressLint("NotifyDataSetChanged")  
 private fun loadHistory() {  
 sharedPreferences = requireActivity().getSharedPreferences("history\_list", Context.*MODE\_PRIVATE*)  
 val json = sharedPreferences.getString("history\_list", null)  
 if (json != null) {  
 val type = object : TypeToken<List<HistoryItem>>() {}.*type* val savedHis: List<HistoryItem> = Gson().fromJson(json, type)  
 historyList.clear()  
 historyList.addAll(savedHis)  
 adapter.notifyDataSetChanged()  
 }  
 }  
  
 fun updateItem(position: Int, newItem: HistoryItem) {  
 historyList[position] = newItem  
 adapter.notifyItemChanged(position)  
 saveHistory()  
 }  
  
 private fun saveHistory() {  
 val sharedPreferences = requireActivity().getSharedPreferences("history\_list", Context.*MODE\_PRIVATE*)  
 val editor = sharedPreferences.edit()  
 val json = Gson().toJson(historyList)  
 editor.putString("history\_list", json)  
 editor.apply()  
 }  
  
 fun isValidDate(dateString: String, dateFormat: String = "dd.MM.yyyy"): Boolean {  
 val sdf = SimpleDateFormat(dateFormat, Locale.getDefault())  
 sdf.*isLenient* = false  
 return try {  
 sdf.parse(dateString) != null  
 } catch (e: ParseException) {  
 false  
 }  
 }  
  
 companion object {  
 fun addHis(context: Context, task: HistoryItem) {  
 val sharedPreferences =  
 context.getSharedPreferences("history\_list", Context.*MODE\_PRIVATE*)  
 val json = sharedPreferences.getString("history\_list", null)  
 val taskList = if (json != null) {  
 val type = object : TypeToken<MutableList<HistoryItem>>() {}.*type* val savedTasks: MutableList<HistoryItem> = Gson().fromJson(json, type)  
 savedTasks  
 } else {  
 *mutableListOf*()  
 }  
 taskList.add(task)  
 sharedPreferences.edit().putString("history\_list", Gson().toJson(taskList)).apply()  
 }  
 @JvmStatic  
 fun newInstance(param1: String, param2: String) =  
 Fragment3().*apply* **{** *arguments* = Bundle().*apply* **{** putString(*ARG\_PARAM1*, param1)  
 putString(*ARG\_PARAM2*, param2)  
 **}  
 }** }  
}

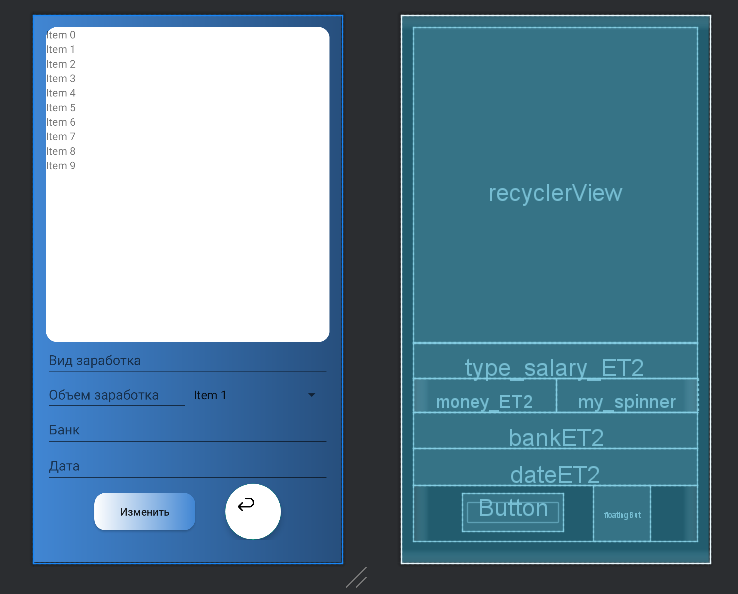
**Макеты страниц:**

**Fragment\_1.xml:** 

**Fragment\_2.xml:**

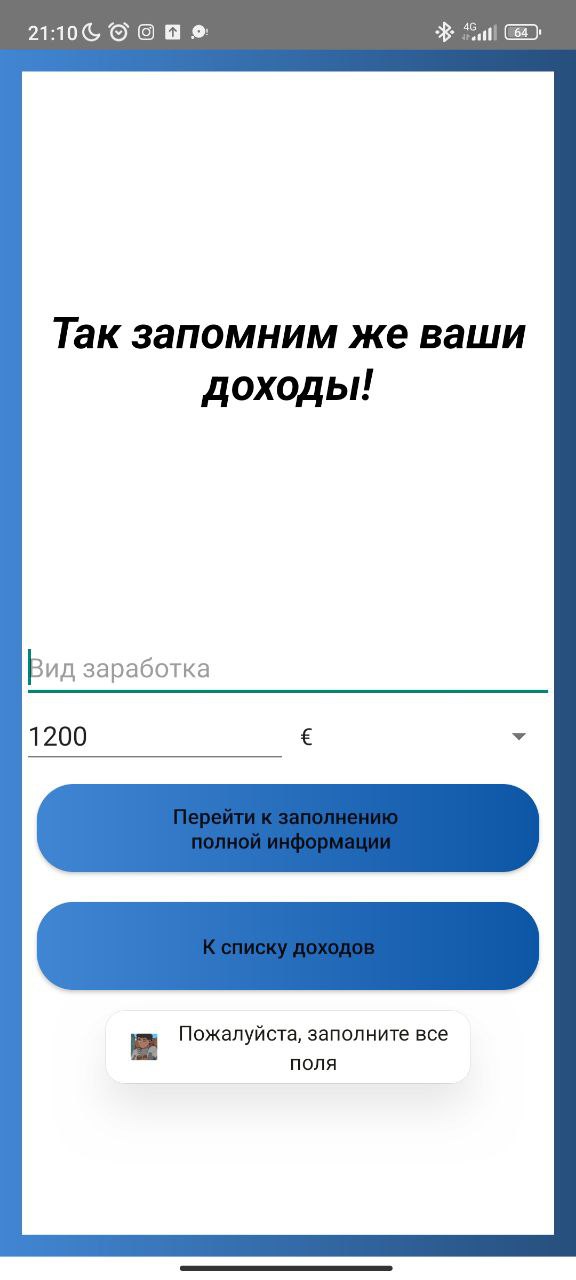


**Fragment\_3:**

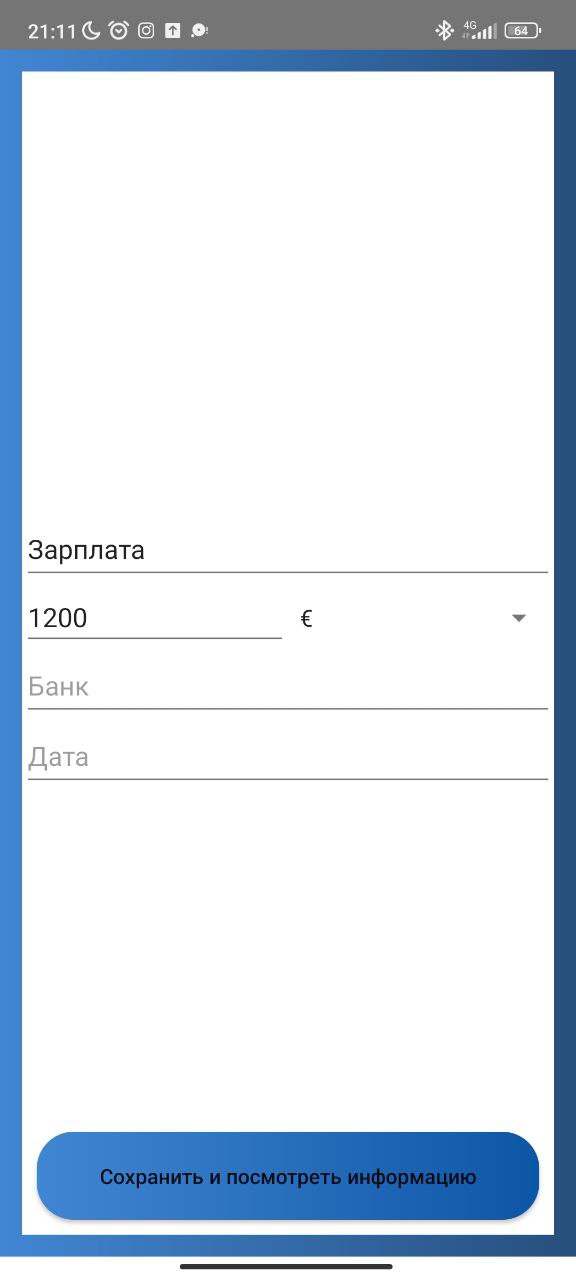
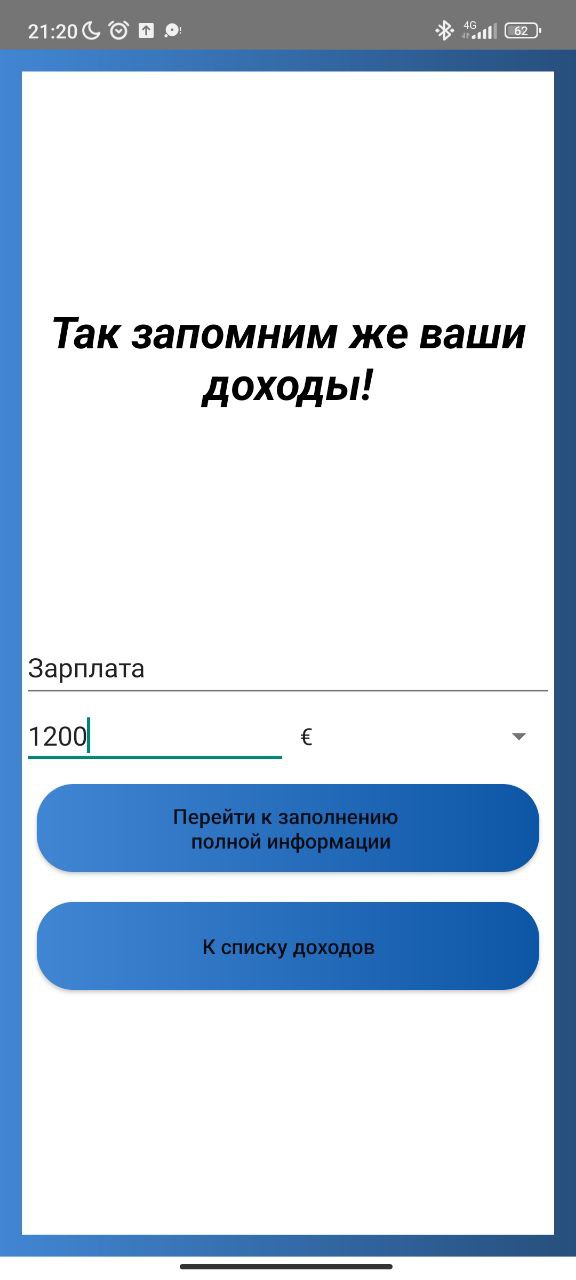


**Тестовые ситуации:**

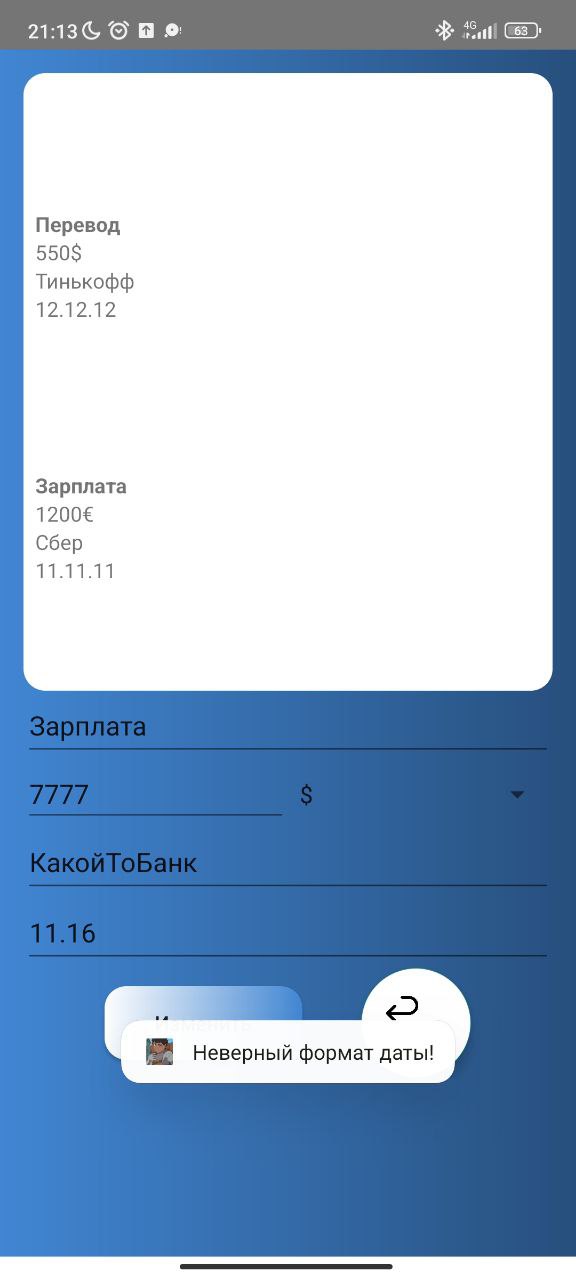
1. Ситуация, когда пользователь не заполнил поля и нажал на кнопку перехода на следующий экран:



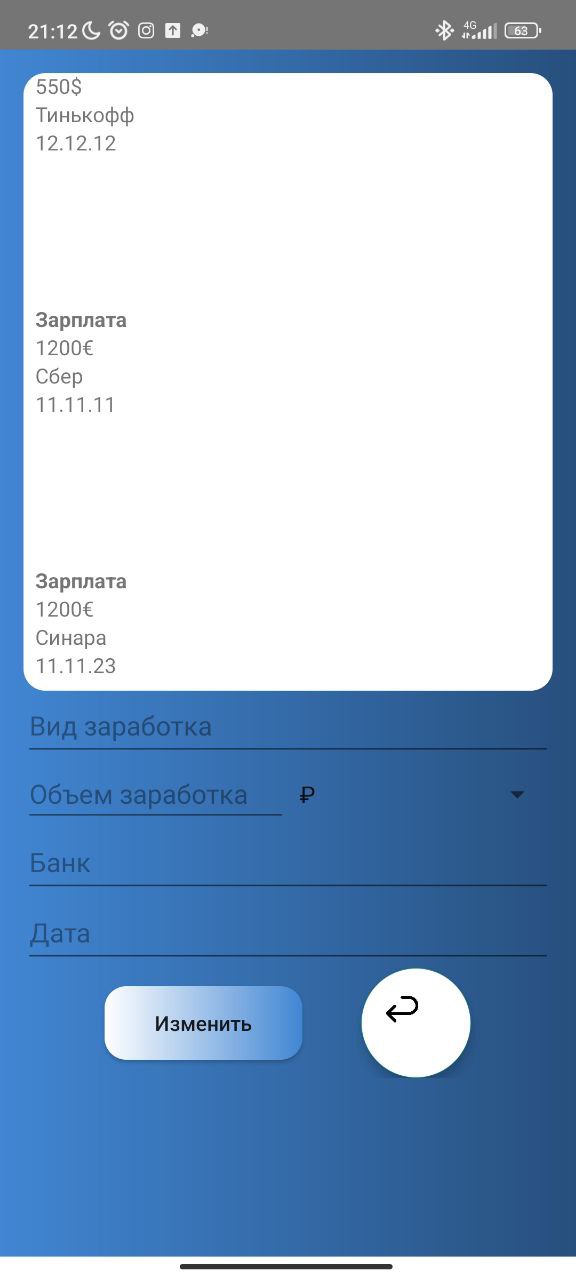
1. Ситуация, когда пользователь правильно ввёл все на первом фрагменте, и перешел на второй фрагемент (информация с первого фрагмента автоматически перенеслась на второй):



1. Ситуация, когда пользователь неправильно ввёл дату на втором / третьем фрагементе:

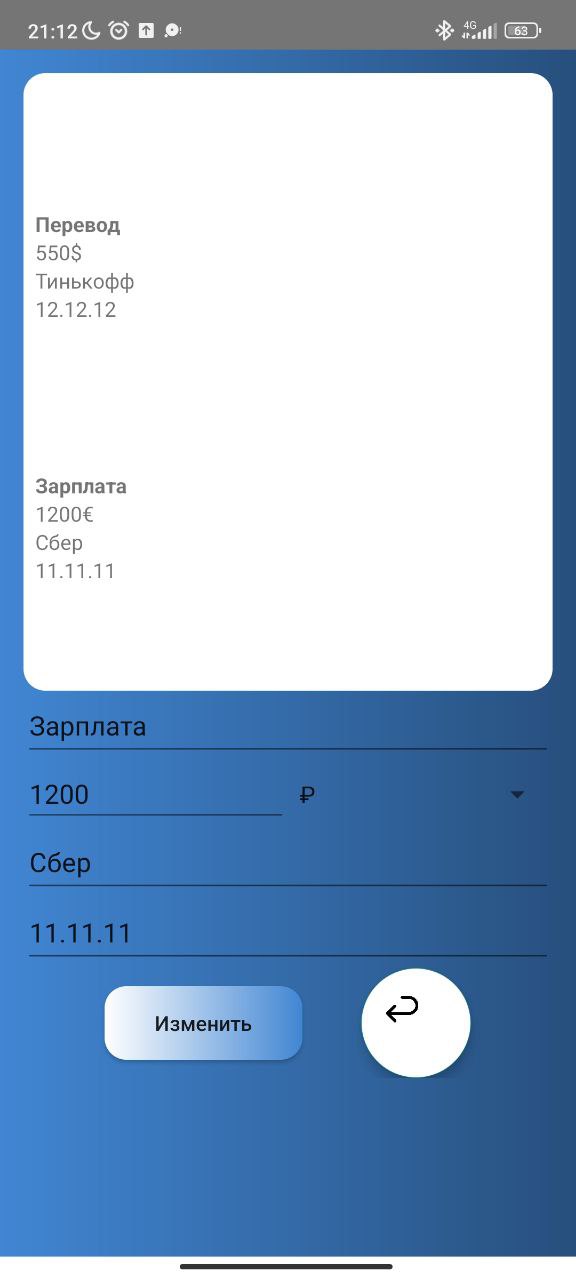
1. Ситуация, когда пользователь сделал все правильно, программа записала доход, и вывела на третий фрагмент:

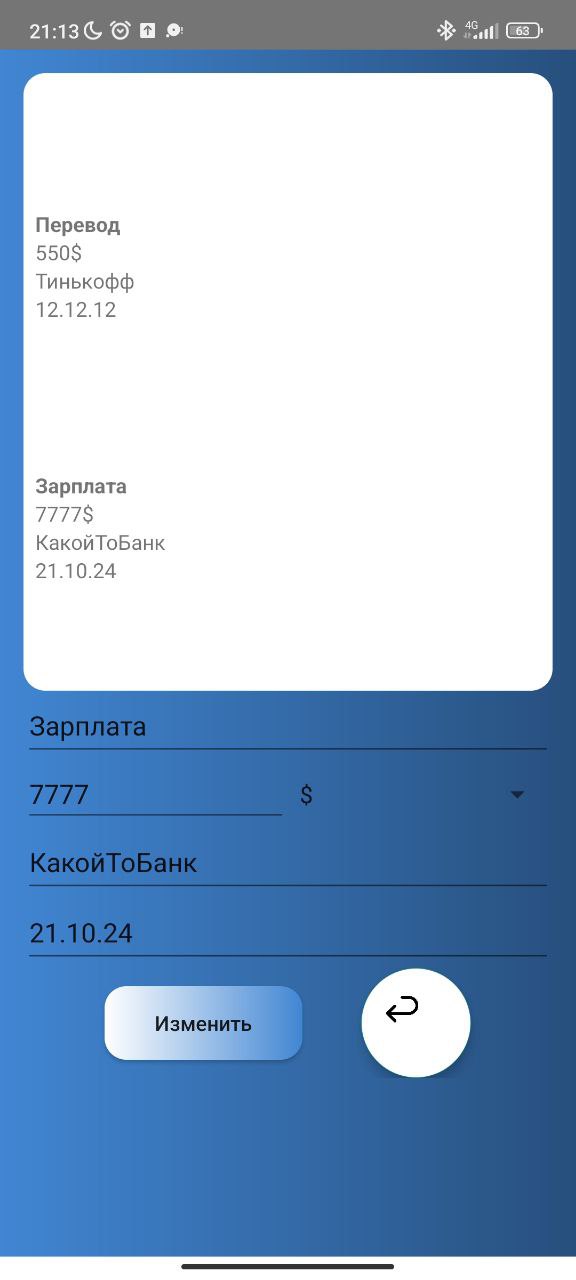


1. Ситуация, когда пользователь решил редактировать выбранный доход, и все сделал правильно:

1) Пользователь нажал на желаемый доход, все нужные поля переместились в поля для редактирования.

2) Пользователь нажал «Изменить» и, так как все было написано правильно, изменения зафиксировались в RecycleView.





**Вывод:** Мы познакомились с использованием Fragment и RecycleView в AndroidStudio.