**Министерство науки и высшего образования Российской Федерации**

Федеральное государственное автономное образовательное

учреждение высшего образования

**Национальный исследовательский**

**Томский политехнический университет**

Инженерная школа информационных технологий и робототехники

Отделение информационных технологий

Отчет по лабораторной работе №18 по дисциплине

**«Язык Kotlin и основы разработки»**

Многопоточность

****

Выполнил:

Студент группы 1А22 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ О.К. Кравцов

Проверил:

Ст. преп. ОИТ ИШИТР \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ В.А. Дорофеев

Томск 2025

# Задание

Напишите приложение, которое скачивает с сайта ЦБ РФ и отображает в списке актуальные курсы валют. Приложение должно выглядеть примерно следующим образом:



Изначально экран пустой, только в нижней части присутствует кнопка «Получить курсы валют». При нажатии этой кнопки в центре экрана появляется крутящийся элемент ProgressBar, и в фоне (в потоке или корутине, по желанию) начинается загрузка файла по адресу https://www.cbr.ru/scripts/XML\_daily.asp – это XML-файл.

# Ход работы

1. Создан проект Lab18 на основе Empty Views Activity.\
2. В файл манифеста добавлено разрешение на доступ к сети Интернет:

<?xml version="1.0" encoding="utf-8"?>  
<manifest xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:tools="http://schemas.android.com/tools">  
  
 <uses-permission android:name="android.permission.INTERNET"/>  
  
 <application  
 android:allowBackup="true"  
 android:dataExtractionRules="@xml/data\_extraction\_rules"  
 android:fullBackupContent="@xml/backup\_rules"  
 android:icon="@mipmap/ic\_launcher"  
 android:label="@string/app\_name"  
 android:roundIcon="@mipmap/ic\_launcher\_round"  
 android:supportsRtl="true"  
 android:theme="@style/Theme.Lab18">  
 <activity  
 android:name=".MainActivity"  
 android:exported="true">  
 <intent-filter>  
 <action android:name="android.intent.action.MAIN" />  
  
 <category android:name="android.intent.category.LAUNCHER" />  
 </intent-filter>  
 </activity>  
 </application>  
  
</manifest>

1. Разработана модель данных для хранения информации о валютах. Создан класс Currency, содержащий поля для названия, значения, номинала и буквенного кода валюты, а также методы для форматированного отображения:

package ru.olegkravtsov.lab18  
  
data class Currency(  
 val name: String,  
 val value: String,  
 val nominal: Int,  
 val charCode: String  
) {  
 // Форматированное название с номиналом  
 fun getDisplayName(): String {  
 return if (nominal == 1) {  
 name  
 } else {  
 "$nominal $name"  
 }  
 }  
  
 // Полное отображение с кодом валюты  
 fun getFullDisplayName(): String {  
 return if (nominal == 1) {  
 "$name ($charCode)"  
 } else {  
 "$nominal $name ($charCode)"  
 }  
 }  
}

1. Настроены ресурсы приложения:

* strings.xml

<?xml version="1.0" encoding="utf-8"?>  
<resources>  
 <string name="app\_name">Lab18</string>  
 <string name="courses\_title">Курсы валют ЦБ РФ</string>  
 <string name="load\_currencies\_button">Получить курсы валют</string>  
 <string name="loading\_success">Загружено %d валют</string>  
 <string name="loading\_error">Ошибка загрузки: %s</string>  
 <string name="url\_currency\_rates">https://www.cbr.ru/scripts/XML\_daily.asp</string>  
 <string name="encoding\_windows1251">Windows-1251</string>  
</resources>

* dimens.xml

<?xml version="1.0" encoding="utf-8"?>  
<resources>  
 <!-- Margins -->  
 <dimen name="default\_margin">16dp</dimen>  
 <dimen name="small\_margin">8dp</dimen>  
  
 <!-- Paddings -->  
 <dimen name="default\_padding">12dp</dimen>  
 <dimen name="medium\_padding">8dp</dimen>  
 <dimen name="small\_padding">2dp</dimen>  
  
 <!-- Text sizes -->  
 <dimen name="text\_large">18sp</dimen>  
 <dimen name="text\_medium">16sp</dimen>  
 <dimen name="text\_small">12sp</dimen>  
</resources>

1. Разработан пользовательский интерфейс. Основной макет activity\_main содержит RecyclerView для отображения списка валют, ProgressBar для индикации загрузки и кнопку для инициирования процесса обновления данных:

<?xml version="1.0" encoding="utf-8"?>  
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:id="@+id/main"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".MainActivity">  
  
 <TextView  
 android:id="@+id/dateTextView"  
 android:layout\_width="0dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_margin="@dimen/default\_margin"  
 android:gravity="center"  
 android:text="@string/courses\_title"  
 android:textSize="@dimen/text\_large"  
 android:textStyle="bold"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent" />  
  
 <androidx.recyclerview.widget.RecyclerView  
 android:id="@+id/currencyRecyclerView"  
 android:layout\_width="0dp"  
 android:layout\_height="0dp"  
 android:padding="@dimen/medium\_padding"  
 app:layoutManager="androidx.recyclerview.widget.LinearLayoutManager"  
 app:layout\_constraintBottom\_toTopOf="@+id/loadButton"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/dateTextView" />  
  
 <ProgressBar  
 android:id="@+id/progressBar"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:visibility="gone"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent" />  
  
 <Button  
 android:id="@+id/loadButton"  
 android:layout\_width="0dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_margin="@dimen/default\_margin"  
 android:text="@string/load\_currencies\_button"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent" />  
  
</androidx.constraintlayout.widget.ConstraintLayout>

1. Создан макет элемента списка item\_currency, обеспечивающий отображение названия валюты с номиналом, буквенного кода и значения курса:

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="horizontal"  
 android:padding="@dimen/default\_padding"  
 android:background="?android:attr/selectableItemBackground">  
  
 <LinearLayout  
 android:layout\_width="0dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_weight="1"  
 android:orientation="vertical">  
  
 <TextView  
 android:id="@+id/currencyName"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:textSize="@dimen/text\_medium"  
 android:textStyle="bold" />  
  
 <TextView  
 android:id="@+id/currencyCode"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:textSize="@dimen/text\_small"  
 android:textColor="@android:color/darker\_gray"  
 android:layout\_marginTop="@dimen/small\_padding" />  
  
 </LinearLayout>  
  
 <TextView  
 android:id="@+id/currencyValue"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:textSize="@dimen/text\_medium"  
 android:textStyle="bold"  
 android:textColor="@android:color/holo\_green\_dark"  
 android:gravity="end" />  
  
</LinearLayout>

1. Реализован адаптер CurrencyAdapter для управления отображением данных в RecyclerView. Адаптер связывает данные о валютах с элементами списка и обеспечивает обновление интерфейса при изменении данных:

package ru.olegkravtsov.lab18  
  
import android.view.LayoutInflater  
import android.view.View  
import android.view.ViewGroup  
import android.widget.TextView  
import androidx.recyclerview.widget.RecyclerView  
  
class CurrencyAdapter(private var currencies: List<Currency>) :  
 RecyclerView.Adapter<CurrencyAdapter.CurrencyViewHolder>() {  
  
 class CurrencyViewHolder(itemView: View) : RecyclerView.ViewHolder(itemView) {  
 val nameTextView: TextView = itemView.findViewById(R.id.currencyName)  
 val codeTextView: TextView = itemView.findViewById(R.id.currencyCode)  
 val valueTextView: TextView = itemView.findViewById(R.id.currencyValue)  
 }  
  
 override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): CurrencyViewHolder {  
 val view = LayoutInflater.from(parent.context)  
 .inflate(R.layout.item\_currency, parent, false)  
 return CurrencyViewHolder(view)  
 }  
  
 override fun onBindViewHolder(holder: CurrencyViewHolder, position: Int) {  
 val currency = currencies[position]  
 holder.nameTextView.text = currency.getDisplayName()  
 holder.codeTextView.text = currency.charCode  
 holder.valueTextView.text = currency.value  
 }  
  
 override fun getItemCount(): Int = currencies.size  
  
 fun updateData(newCurrencies: List<Currency>) {  
 currencies = newCurrencies  
 notifyDataSetChanged()  
 }  
}

1. В основной активности MainActivity реализована логика работы приложения. Для выполнения сетевых операций и работы с XML использованы корутины с диспетчерами IO и Main, что обеспечивает неблокирующее выполнение длительных операций и корректное обновление пользовательского интерфейса:

package ru.olegkravtsov.lab18  
  
import android.os.Bundle  
import android.util.Xml  
import androidx.activity.enableEdgeToEdge  
import androidx.appcompat.app.AppCompatActivity  
import androidx.core.view.ViewCompat  
import androidx.core.view.WindowInsetsCompat  
import androidx.recyclerview.widget.LinearLayoutManager  
import kotlinx.coroutines.CoroutineScope  
import kotlinx.coroutines.Dispatchers  
import kotlinx.coroutines.launch  
import kotlinx.coroutines.withContext  
import org.xmlpull.v1.XmlPullParser  
import java.net.URL  
import java.nio.charset.Charset  
  
class MainActivity : AppCompatActivity() {  
  
 private lateinit var currencyAdapter: CurrencyAdapter  
  
 override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
 enableEdgeToEdge()  
 setContentView(R.layout.activity\_main)  
  
 ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main)) { v, insets ->  
 val systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars())  
 v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom)  
 insets  
 }  
  
 setupRecyclerView()  
 setupButton()  
 }  
  
 private fun setupRecyclerView() {  
 val recyclerView = findViewById<androidx.recyclerview.widget.RecyclerView>(R.id.currencyRecyclerView)  
 currencyAdapter = CurrencyAdapter(emptyList())  
 recyclerView.apply {  
 layoutManager = LinearLayoutManager(this@MainActivity)  
 adapter = currencyAdapter  
 }  
 }  
  
 private fun setupButton() {  
 val loadButton = findViewById<android.widget.Button>(R.id.loadButton)  
 val progressBar = findViewById<android.widget.ProgressBar>(R.id.progressBar)  
  
 loadButton.setOnClickListener {  
 progressBar.visibility = android.view.View.VISIBLE  
 loadButton.isEnabled = false  
  
 CoroutineScope(Dispatchers.IO).launch {  
 try {  
 val currencies = loadCurrencies()  
 withContext(Dispatchers.Main) {  
 currencyAdapter.updateData(currencies)  
 progressBar.visibility = android.view.View.GONE  
 loadButton.isEnabled = true  
  
 // Используем ресурсы для строки  
 val successMessage = getString(R.string.loading\_success, currencies.size)  
 android.widget.Toast.makeText(  
 this@MainActivity,  
 successMessage,  
 android.widget.Toast.LENGTH\_SHORT  
 ).show()  
 }  
 } catch (e: Exception) {  
 withContext(Dispatchers.Main) {  
 progressBar.visibility = android.view.View.GONE  
 loadButton.isEnabled = true  
  
 // Используем ресурсы для строки ошибки  
 val errorMessage = getString(R.string.loading\_error, e.message ?: "Неизвестная ошибка")  
 android.widget.Toast.makeText(  
 this@MainActivity,  
 errorMessage,  
 android.widget.Toast.LENGTH\_LONG  
 ).show()  
 }  
 }  
 }  
 }  
 }  
  
 private fun loadCurrencies(): List<Currency> {  
 val currencies = mutableListOf<Currency>()  
  
 // Используем ресурсы для URL и кодировки  
 val url = getString(R.string.url\_currency\_rates)  
 val encoding = getString(R.string.encoding\_windows1251)  
  
 val xml = URL(url)  
 .readText(Charset.forName(encoding))  
  
 val parser = Xml.newPullParser()  
 parser.setInput(xml.reader())  
  
 var eventType = parser.eventType  
 var currentName = ""  
 var currentValue = ""  
 var currentNominal = 1  
 var currentCharCode = ""  
  
 var inValute = false  
 var inName = false  
 var inValue = false  
 var inNominal = false  
 var inCharCode = false  
  
 while (eventType != XmlPullParser.END\_DOCUMENT) {  
 when (eventType) {  
 XmlPullParser.START\_TAG -> {  
 when (parser.name) {  
 "Valute" -> {  
 inValute = true  
 currentName = ""  
 currentValue = ""  
 currentNominal = 1  
 currentCharCode = ""  
 }  
 "Name" -> if (inValute) inName = true  
 "Value" -> if (inValute) inValue = true  
 "Nominal" -> if (inValute) inNominal = true  
 "CharCode" -> if (inValute) inCharCode = true  
 }  
 }  
  
 XmlPullParser.TEXT -> {  
 when {  
 inName -> currentName = parser.text  
 inValue -> currentValue = parser.text  
 inNominal -> currentNominal = parser.text.toIntOrNull() ?: 1  
 inCharCode -> currentCharCode = parser.text  
 }  
 }  
  
 XmlPullParser.END\_TAG -> {  
 when (parser.name) {  
 "Valute" -> {  
 inValute = false  
 if (currentName.isNotEmpty() && currentValue.isNotEmpty() && currentCharCode.isNotEmpty()) {  
 currencies.add(  
 Currency(  
 name = currentName,  
 value = currentValue,  
 nominal = currentNominal,  
 charCode = currentCharCode  
 )  
 )  
 }  
 }  
 "Name" -> inName = false  
 "Value" -> inValue = false  
 "Nominal" -> inNominal = false  
 "CharCode" -> inCharCode = false  
 }  
 }  
 }  
 eventType = parser.next()  
 }  
  
 // Сортируем валюты по алфавиту для удобства  
 return currencies.sortedBy { it.name }  
 }  
}

**Результат работы**

Приложение успешно запускается и отображает интерфейс с кнопкой "Получить курсы валют" (рис. 1). При нажатии на кнопку активируется индикатор загрузки и в фоновом режиме начинается загрузка данных с сайта ЦБ РФ (рис. 2).



Рисунок 1 – Интерфейс с кнопкой "Получить курсы валют"

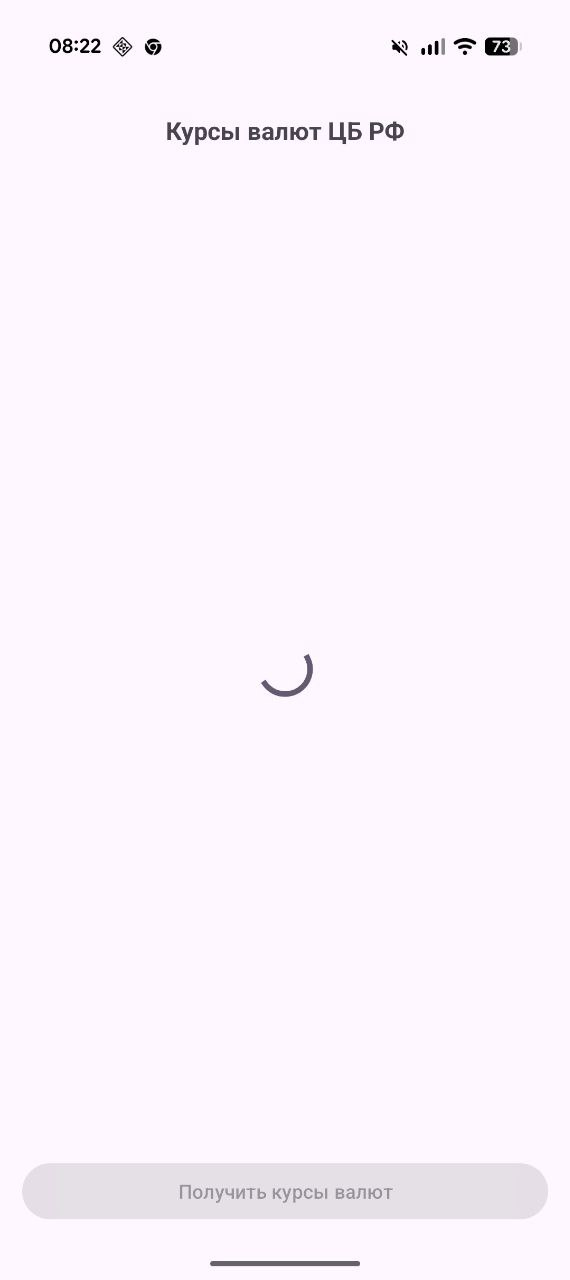


Рисунок 2 – Интерфейс во время загрузки

После завершения загрузки и обработки данных на экране отображается отсортированный по алфавиту список валют (рис. 3). Для каждой валюты указывается название, номинал, буквенный код и значение курса в рублях.

Рисунок 3 – Интерфейс с курсами валют

В случае ошибки сети или обработки данных пользователю отображается соответствующее сообщение (рис. 4).



Рисунок 4 – Ошибка загрузки

# Выводы

В ходе выполнения лабораторной работы было разработано приложение для отображения курсов валют, использующее современные подходы к организации многопоточности в Android. Применение корутин позволило эффективно разделить выполнение сетевых операций и обновление пользовательского интерфейса, обеспечивая отзывчивость приложения во время загрузки данных. Реализация парсера XML на основе XmlPullParser продемонстрировала работу с структурированными данными из внешних источников.