

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 1A22



О.К. Кравцов

Проверил:

Ст. преп. ОИТ ИШИТР



В.А. Дорофеев

Задание

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

Австралийский доллар	62,3530
Азербайджанский манат	54,0817
Фунт стерлингов Соединенного королевства	16,3579
Армянских драмов x100	22,6663
Белорусский рубль	28,6986
Болгарский лев	51,6279
Бразильский реал	18,8585
Венгерских форинтов x100	26,4702
Вьетнамских донгов x10000	38,4440
Гонконгский доллар	11,7885
Грузинский лари	34,2009
Датская крона	13,5433
Дирхам ОАЭ	25,0344
Доллар США	91,9389
Евро	101,2863
Египетских фунтов x10	29,7602
Индийских рупий x10	11,0530
Индонезийских рупий x10000	59,1894
Казахстанских тенге x100	20,1184
Канадский доллар	69,0699
Катарский риал	25,2579
Киргизских сомов x10	10,3042
Китайский юань	12,8598
Молдавских леев x10	52,3824
Новозеландский доллар	57,8571
Норвежских крон x10	89,2411
Польский злотый	23,3715
Румынский лей	20,3851
СДР (специальные права заимствования)	123,0933
Сингапурский доллар	69,2885
Таиландский бат x10	82,0111

Получить курсы валют

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>
```

3. 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)"
        }
    }
}
```

4. 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>
```

5. 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>

```

6. 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>

```

7. 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()
    }
}
```


8. 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.currencyRecycler
View)

        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"

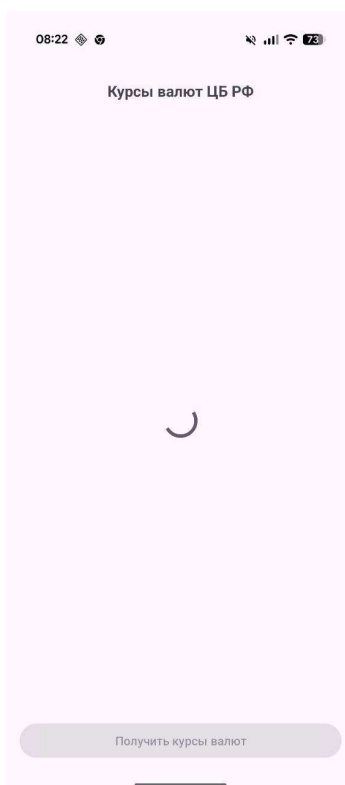


Рисунок 2 – Интерфейс во время загрузки

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.

Курсы валют ЦБ РФ		Курсы валют ЦБ РФ	
Австралийский доллар	51,6008	100 Тенге	14,6776
AUD		KZT	
Азербайджанский манат	46,4616	1000 Тугриков	21,9996
AZN		MNT	
100 Алжирских динаров	60,7213	10 Турецких лир	18,8596
DZD		TRY	
100 Армянских драмов	20,6388	10000 Узбекских сумов	65,1832
AMD		UZS	
10 Батов	24,1706	100 Форинтов	23,5881
THB		HUF	
Бахрейнский динар	210,0206	Фунт стерлингов	105,0577
BHD		GBP	
Белорусский рубль	26,9849	10 Чешских крон	37,6854
BYN		CZK	
Болгарский лев	46,8942	10 Шведских крон	84,1133
BGN		SEK	
Боливиано	11,4305	Швейцарский франк	99,2396
BOB		CHF	
Бразильский реал	14,6828	100 Эфиопских быров	52,1255
BRL		ETB	
1000 Вон	54,9995	Юань	11,0648
KRW		CNY	
Загружено 55 валют		Получить курсы валют	
Получить курсы валют		Получить курсы валют	

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.