|  |  |
| --- | --- |
|  | МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ ТАТАРСТАН **Государственное автономное профессиональное образовательное учреждение** «ЗЕЛЕНОДОЛЬСКИЙ МЕХАНИЧЕСКИЙ КОЛЛЕДЖ»  (ГАПОУ «ЗМК») |

**09.02.07 «Информационные системы и программирование»**

# МДК 04.01. ВНЕДРЕНИЕ И УПРАВЛЕНИЕ КОМПЬЮТЕРНЫХ СИСТЕМ

**Отчет о практических работах**

**Исполнитель**: Архипов Артём Алексеевич

**Группа**: 217

**Преподаватель:** Алемасов Евгений Павлович

**Дата сдачи** 23.01.2025 **Оценка** \_\_\_\_\_\_\_\_\_\_\_\_

**Подпись преподавателя**  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Работа №8. Создание мини интерфейса для импортированных данных**

Данная работа сочетает в себе почти все, что было пройдено до этого. Поэтому всё, что я сделаю - это опишу саму программу: то, как она работает и что для этого применяется (очень кратно, т.к., опять же, всё это было рассмотрено в предыдущих работах). Вот код программы:

package org.example;  
  
import com.fasterxml.jackson.databind.ObjectMapper;  
import org.apache.poi.ss.usermodel.Row;  
import org.apache.poi.xssf.usermodel.XSSFSheet;  
import org.apache.poi.xssf.usermodel.XSSFWorkbook;  
import org.json.JSONObject;  
import java.io.\*;  
import java.net.HttpURLConnection;  
import java.net.URL;  
import java.util.List;  
import java.util.Scanner;  
  
public class Main {  
 public static void main(String[] args) {  
 String BASE\_URL = "https://api.coindesk.com/v1/bpi/currentprice.json";  
 try {  
 URL url = new URL(BASE\_URL);  
 HttpURLConnection con = (HttpURLConnection) url.openConnection();  
 con.setRequestMethod("GET");  
 BufferedReader in = new BufferedReader(new InputStreamReader(con.getInputStream()));  
 String line;  
 StringBuilder responce = new StringBuilder();  
 while ((line = in.readLine()) != null)  
 responce.append(line);  
  
 in.close();  
 con.disconnect();  
  
 ObjectMapper mapper = new ObjectMapper();  
  
 mapper.writeValue(new File("info.json"),responce);  
  
  
 JSONObject first = new JSONObject(responce.toString());  
  
 List<String> values = List.*of*("USD","GBP","EUR","rate","description","rate\_float","time");  
  
 Scanner scanner = new Scanner(System.*in*);  
 JSONObject jsobj;  
 JSONObject cur;  
 JSONObject price;  
 JSONObject desc;  
 XSSFWorkbook book = new XSSFWorkbook();  
 XSSFSheet list = book.createSheet("Temp");  
 Row row;  
 int i = 0;

while(true){  
 System.*out*.println("Что вывести:\n1)Время последнего обновления;\n2)USD;\n3)GBR;\n4)EUR;\n5)Цена;\n6)Описание;\n7)Выход");  
 String userInput = scanner.nextLine();  
 List<String> userInputList = List.*of*(userInput.split(" "));  
 if(userInputList.contains("1")) {  
 jsobj = first.getJSONObject("time");  
 System.*out*.println("Время последнего обновления: " + jsobj.getString("updated"));  
 }  
 if(!(userInputList.contains("2") || userInputList.contains("3") || userInputList.contains("4"))) {  
 System.*out*.println("Пожалуйст, укажите валюту, информацию которой вы хотите узнать");  
 continue;  
 }  
 jsobj = first.getJSONObject("bpi");  
  
 if(userInputList.contains("2")) {  
 row = list.createRow(i);  
 cur = jsobj.getJSONObject("USD");  
 row.createCell(0).setCellValue(cur.getString("code"));  
 System.*out*.println("Валюта: " + cur.getString("code"));  
 if(userInputList.contains("5")){  
 System.*out*.println("Цена: " + cur.getString("rate"));  
 row.createCell(1).setCellValue(cur.getString("rate"));  
 }  
 if(userInputList.contains("6")) {  
 System.*out*.println("Описание: " + cur.getString("description"));  
 row.createCell(2).setCellValue(cur.getString("description"));  
 }  
  
 }  
 if(userInputList.contains("3")) {  
 i ++;  
 row = list.createRow(i);  
 cur = jsobj.getJSONObject("GBP");  
 row.createCell(0).setCellValue(cur.getString("code"));  
 System.*out*.println("Валюта: " + cur.getString("code"));  
 if(userInputList.contains("5")) {  
 System.*out*.println("Цена: " + cur.getString("rate"));  
 row.createCell(1).setCellValue(cur.getString("rate"));  
 }  
 if(userInputList.contains("6")) {  
 System.*out*.println("Описание: " + cur.getString("description"));  
 row.createCell(2).setCellValue(cur.getString("description"));  
 }  
 }  
 if(userInputList.contains("4")) {  
 i ++;  
 row = list.createRow(i);  
 cur = jsobj.getJSONObject("EUR");  
 row.createCell(0).setCellValue(cur.getString("code"));  
 System.*out*.println("Валюта: " + cur.getString("code"));  
 if(userInputList.contains("5")){  
 System.*out*.println("Цена: " + cur.getString("rate"));  
 row.createCell(1).setCellValue(cur.getString("rate"));  
 }  
 if(userInputList.contains("6")) {  
 System.*out*.println("Описание: " + cur.getString("description"));  
 row.createCell(1).setCellValue(cur.getString("rate"));  
 }  
 }  
 if(userInputList.contains("7")){

book.write(new FileOutputStream("E:\\Test.xlsx"));  
 book.close();  
 break;  
 }

book.write(new FileOutputStream("E:\\Test.xlsx"));

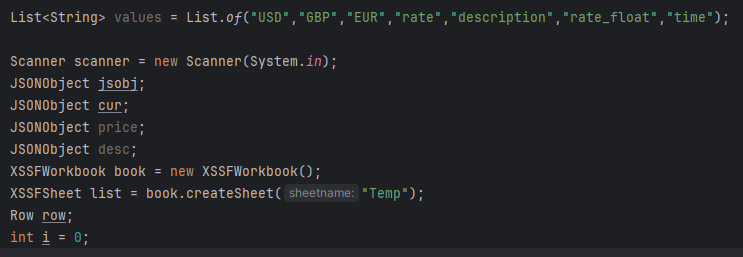
i++;  
  
 }  
  
  
 }  
 catch (Exception e){  
 System.*out*.println(e.getMessage());  
 }  
  
 }  
}

Все зависимости находятся в текстовом файле Pomp, а исходник можно найти в Main.

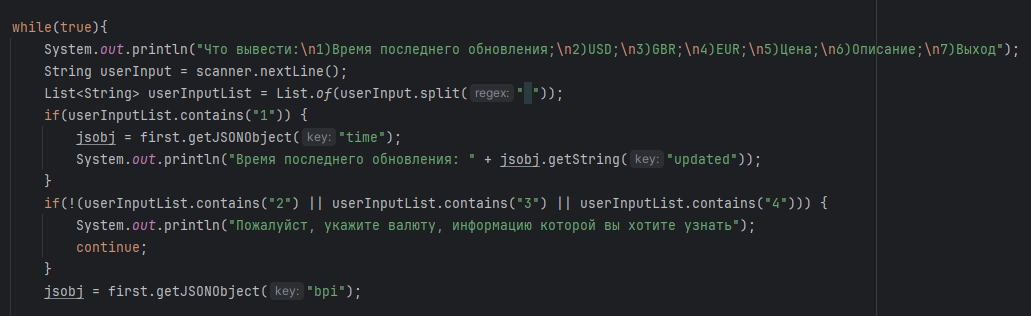
Все начинается с парсинга данных по API с последующей записью в JSON файл



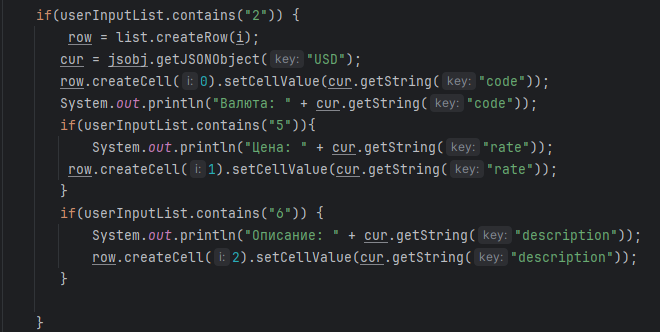
Создаём переменные, которые будут (или уже) хранить данные для последующей работы.



Спрашиваем user-а, что он хочет получить, и параллельно проверяем введённые данные. Если те не содержат в себе выбранной валюты, то выводим предупреждение и пропускаем итерацию



В случае если проверка прошла успешно, то выполняем вывод данных на основе ввода user-а



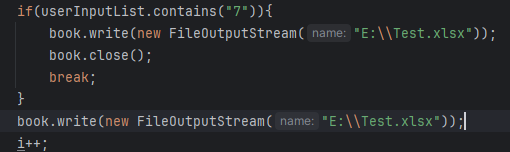
Данный код в программе повторяется три раза с единственным отличием, что название валюты меняется на другое:





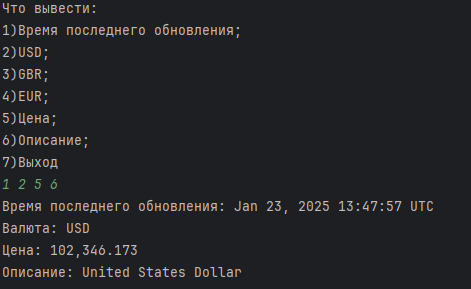


Проверяем, не ввёл ли user-а код для выхода из цикла, если да, то заканчиваем код программы и записываем все в файл.



**Пример работы:**

Запрошенные данные:



Результат:

