|  |
| --- |
|  |
| МИНОБРНАУКИ РОССИИ |
| *Федеральное государственное бюджетное образовательное учреждение высшего образования* ***«МИРЭА – Российский технологический университет»***  **РТУ МИРЭА** |

**Институт информационных технологий (ИТ)**

**Кафедра инструментального и прикладного программного обеспечения (ИиППО)**

**Дисциплина «Программирование на языке Джава»**

**ОТЧЕТ**

**ПО ПРАКТИЧЕСКОМУ ЗАНЯТИЮ №23-24**

Выполнил студент группы ИНБО-02-20 Маркарянц Р.Д.

Принял Степанов П.В.

Практические работы выполнены «\_\_\_»\_\_\_\_\_\_\_2021г.

«\_\_\_\_\_\_\_\_\_\_\_\_\_\_» «\_\_\_»\_\_\_\_\_\_\_2021г.

Отметка о выполнении

**Москва – 2021 г.**

## **Задание**

1) Парс CSV файлов

Создать новый проект, который будет читать файл csv банковской выписки movementsList.csv и парсить полученные строки. Путь к файлу выписки храните в константе. Код должен выводить сводную информацию по этой выписке: общий приход, общий расход и разбивку расходов.

Для выполнения работы предлагаю пользоваться библиотекой OpenCSV.

2) Парс HTML кода страницы

Напишите программу, которая:

* получает с помощью библиотеки jsoup HTML-код страницы [https://www.mirea.ru](https://www.mirea.ru/);
* находит в HTML-коде теги img и получает абсолютные ссылки на изображения из атрибута src;
* скачивает изображения в папку images проекта, при этом сохраняя оригинальные названия файлов;
* выводит в консоль список c названиями скачанных файлов.

## **Ход Работы**

В ходе выполнения работы были получены следующие исходные коды:

package ru.mirea.inbo0220.markaryants.pr23.Task1;  
  
import java.util.HashMap;  
import java.util.Map;  
  
public class CSVParser {  
 private double income = 0;  
 private double expense = 0;  
 private final Map<String, Double> expenseList = new HashMap<>();  
  
 public void CalculateData(String input) {  
 String[] columns = input.split(",", 8);  
 double incomeTmp = Double.*parseDouble*(columns[6]);  
 double expenseTmp = Double.*parseDouble*(columns[7].  
 replaceAll("\\\"", "").replace(',', '.'));  
 if (incomeTmp == 0)  
 calculateExpense(columns[5], expenseTmp);  
 else  
 this.income += incomeTmp;  
 }  
  
 private void calculateExpense(String expenseName, double expense) {  
 this.expense += expense;  
 String[] transaction = expenseName.trim().split(" {3,}");  
 String[] temp = transaction[1].split("/");  
 String[] suppliesTemp = temp[temp.length - 1].split("\\\\");  
 String supplies = suppliesTemp[suppliesTemp.length - 1];  
 if (!expenseList.containsKey(supplies))  
 expenseList.put(supplies, expense);  
 else {  
 double sum = expenseList.get(supplies);  
 sum += expense;  
 expenseList.put(supplies.trim(), sum);  
 }  
 }  
  
 public void printCSV() {  
 System.*out*.printf("Сумма расходов: %.2f руб.\n", expense);  
 System.*out*.printf("Сумма доходов: %.2f руб.\n", income);  
 System.*out*.println("Суммы расходов по организациям:");  
  
 for (String supplies : expenseList.keySet()) {  
 System.*out*.printf("%-30s %.2f %-10s\n", supplies, expenseList.get(supplies), "руб.");  
 }  
 }  
}

package ru.mirea.inbo0220.markaryants.pr23.Task1;  
  
import java.io.IOException;  
 import java.nio.file.Files;  
 import java.nio.file.Paths;  
 import java.util.List;  
  
public class Test {  
 public static void main(String[] args) throws IOException {  
 CSVParser csv = new CSVParser();  
 List<String> lines = Files.*readAllLines* (Paths.*get*("src/ru/mirea/inbo0220/markaryants/pr23/Task1/movementList.csv"));  
 for (int i = 1; i < lines.size(); i++) {  
 csv.CalculateData(lines.get(i));  
 }  
 csv.printCSV();  
 }  
}

package ru.mirea.inbo0220.markaryants.pr23.Task2;  
  
import java.io.\*;  
import java.net.URL;  
  
import org.jsoup.Jsoup;  
import org.jsoup.nodes.Document;  
import org.jsoup.nodes.Element;  
import org.jsoup.select.Elements;  
  
public class HTMLParser {  
 private static String *IMAGE\_DESTINATION\_FOLDER* = "src/ru/mirea/inbo0220/markaryants/pr23/Task2/images";  
  
 public static void main(String[] args) throws IOException {  
 new File("src/ru/mirea/inbo0220/markaryants/pr23/Task2/images").mkdirs();  
  
 String strURL = "http://www.mirea.ru";  
 Document document = Jsoup  
 .connect(strURL)  
 .userAgent("Mozilla/5.0")  
 .timeout(10 \* 1000)  
 .get();  
  
 Elements imageElements = document.select("img");  
  
 for(Element imageElement : imageElements){  
 String strImageURL = imageElement.attr("abs:src");  
 *downloadImage*(strImageURL);  
 }  
 }  
  
 private static void downloadImage(String strImageURL){  
 String strImageName =  
 strImageURL.substring( strImageURL.lastIndexOf("/") + 1 );  
  
 System.*out*.println("Saving: " + strImageName + ", from: " + strImageURL);  
 try {  
 URL urlImage = new URL(strImageURL);  
 InputStream in = urlImage.openStream();  
  
 byte[] buffer = new byte[4096];  
 int n = -1;  
  
 OutputStream os =  
 new FileOutputStream( *IMAGE\_DESTINATION\_FOLDER* + "/" + strImageName );  
  
 while ( (n = in.read(buffer)) != -1 ){  
 os.write(buffer, 0, n);  
 }  
  
 os.close();  
 System.*out*.println("Image saved");  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
 }  
}

## **Вывод**

Выполнив практическую работу, я изучил работу с библиотекой OpenCSV и jsoup на языке Java.