|  |
| --- |
|  |
| 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 ПРАКТИЧЕСКИМ ЗАНЯТИЯМ №23, №24**

Выполнил студент группы ИНБО-02-20 Зотов Е. А.

Принял Степанов П.В.

Практические работы выполнены «\_\_\_»\_\_\_\_\_\_\_2021 г.

«\_\_\_\_\_\_\_\_\_\_\_\_\_\_» «\_\_\_»\_\_\_\_\_\_\_2021 г.

Отметка о выполнении

**Москва – 2021 г.**

**Практические занятия №23-24**

**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. Ссылка в рекомендациях.

Примеры работы программы:

Сумма расходов: 398 563.39 руб.

Сумма доходов: 289 890.06 руб.

Суммы расходов по организациям:

RUSMOSKVA56 SHLOVE REPUBLIC 1 081.53 руб.

RUSMOSCOW42 SHCL ETOILE 126.34 руб.

RUSPUSHKINO105ZOOMAGAZIN 4 217.65 руб.

# 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.

**Код программы**

**Класс Main**

package ru.mirea.java.practice23.practice24;  
  
import java.io.IOException;  
import java.nio.file.Files;  
import java.nio.file.Paths;  
import java.util.List;  
public class Main {  
 public static void main(String[] args) throws IOException {  
 ParserHTML ex1 = new ParserHTML();  
 // второе задание  
 String csvFilename = "src/ru/mirea/java/practice23/practice24/movementList.csv";  
 List<String> list = Files.*readAllLines*(Paths.*get*(csvFilename));  
 DateCompany dateCompany = new DateCompany();  
 for(int i=1;i<list.size();i++){ // первая строка с названиями столбцовA  
 dateCompany.newDateCompany(list.get(i));  
 }  
 dateCompany.outResult();  
 }  
}

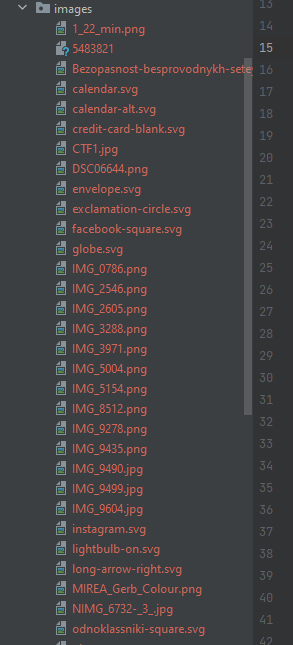
**Класс DateCompany**

package ru.mirea.java.practice23.practice24;  
  
import java.util.HashMap;  
import java.util.Map;  
  
public class DateCompany {  
 private double expenses = 0; // общее количество расходов  
 private double income = 0; // общее количество доходов  
 private Map<String, Double> expensesAllComp = new HashMap<String, Double>(); // y каждой компании свои расходы  
  
 public void newDateCompany(String str) {  
  
 String[] stroka = str.split(",", 8);  
 double dohod = Double.*parseDouble*(stroka[6].replaceAll("\"", "").replace(',', '.'));  
 double rashod = Double.*parseDouble*(stroka[7].replaceAll("\"", "").replace(',', '.')); // некоторые данные имеют вид "129,56", нужно избавиться от ковычек и , превратить в .  
 this.expenses += rashod;  
 this.income += dohod;  
  
 String[] s = stroka[5].trim().split(" {3,}")[1].split("/"); //выделение основной части, разделение елси больше 3 или больше 3 пробелов, так же 1 делит по / и кладет в s  
 String[] strokaCompany = s[s.length - 1].split("\"); // основная часть всегда последняя, с проверкой на \  
 String company = strokaCompany[strokaCompany.length - 1]; //достаем название компании  
  
 if (!expensesAllComp.containsKey(company)) { // если компании до этого не было, добавляем  
 expensesAllComp.put(company, rashod);  
  
 } else {  
 double sum = expensesAllComp.get(company);  
 sum += rashod;  
 expensesAllComp.put(company, sum);  
 }  
  
 }  
  
 public void outResult() {  
 System.*out*.println("Сумма доходов:" + this.income + " rub");  
 System.*out*.println("Сумма расходов " + this.expenses + " rub");  
 System.*out*.println("Расходы по компаниям: ");  
  
 for (String company : this.expensesAllComp.keySet()) {  
  
 System.*out*.println(company + ": " + this.expensesAllComp.get(company) + "руб");  
  
 }  
  
  
 }  
}

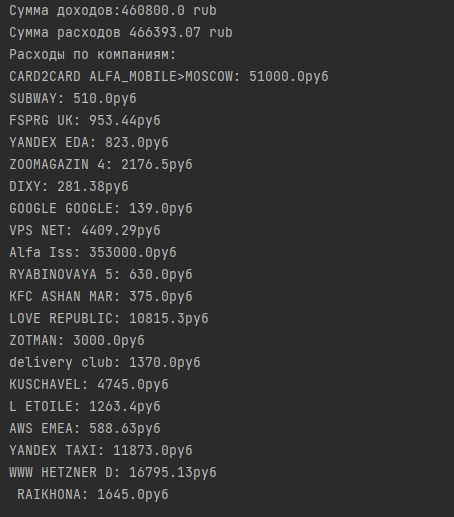
**Класс ParserHTML**

package ru.mirea.java.practice23.practice24;  
  
import org.jsoup.Jsoup;  
import org.jsoup.nodes.Document;  
import org.jsoup.nodes.Element;  
import org.jsoup.select.Elements;  
  
import java.io.\*;  
import java.net.URL;  
  
public class ParserHTML {  
 private final String URL = "http://www.mirea.ru";  
 private final String FOLDER = "src/ru/mirea/java/practice23/practice24/images";  
  
 public ParserHTML() throws IOException{  
  
  
 new File(FOLDER).mkdirs(); // папка, куда будут сохранены изображения  
 Document document = Jsoup.*connect*(URL).get(); // get HTML  
 Elements elements = document.select("img"); // all img in document  
  
  
 String imgURL;  
  
  
 for(Element img : elements){  
  
 imgURL = img.attr("abs:src");  
 if (!imgURL.equals("")){  
 String imgName = imgURL.substring(imgURL.lastIndexOf("/")+1);  
 System.*out*.println("image name: " + imgName);  
 try {  
 URL url = new URL(imgURL);  
  
 InputStream in = url.openStream();  
 byte[] buffer = new byte[4096];  
 int n = -1;  
  
 OutputStream folderOUt = new FileOutputStream(FOLDER + "/" + imgName);  
  
 while ( (n= in.read(buffer)) != -1){ // сохранение в файл  
 folderOUt.write(buffer,0,n);  
 }  
 folderOUt.close();  
 } catch (IOException e){  
 }  
  
 }  
  
 }  
  
 }  
  
}

**Тестирование**



**Рисунок 1 – Тестирование 1**



**Рисунок 2 – Тестирование 2**

**Выводы**

Был изучен парсинг HTML-код страницы и находить в нем нужные абсолютные ссылки на изображения, а также скачивать эти изображения на компьютер. Во второй практике я научился работать с данными в csv файле.

Git: <https://github.com/kwazr/mirea-practice/tree/master/src/ru/mirea>