|  |
| --- |
|  |
| 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 №25, №26**

Выполнил студент группы ИНБО-02-20 Самойленко М. А.

Принял Степанов П.В.

Практические работы выполнены «\_\_\_»\_\_\_\_\_\_\_2020г.

«\_\_\_\_\_\_\_\_\_\_\_\_\_\_» «\_\_\_»\_\_\_\_\_\_\_2020г.

Отметка о выполнении

**Москва – 2020 г.**

**Практическое занятие №25-26**

**Задание:**

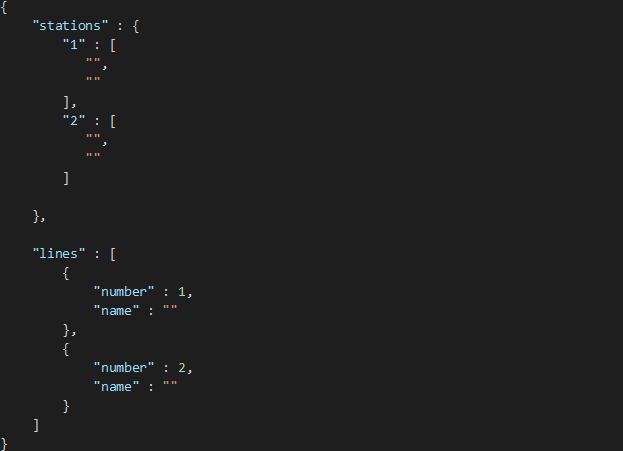
Получение данных из HTML-страницы, создания и чтения JSON-файлы.

Получает HTML-код страницы «Список станций Московского метрополитена» <https://www.moscowmap.ru/metro.html#lines> с помощью библиотеки jsoup.

Парсит полученную страницу и получает из неё:

Линии московского метро (получаете имя линии, номер линии).

Станции московского метро (получаете имя станции, номер линии).

[](https://github.com/artemmad/JavaAllExercices/blob/master/src/ru/mirea/exercies25and26/1.png?raw=true)Создаёт и записывает на диск JSON-файл со списком станций по линиям и списком линий по формату:

Читает файл и выводит в консоль количество станций на каждой линии.

**Код программы:**

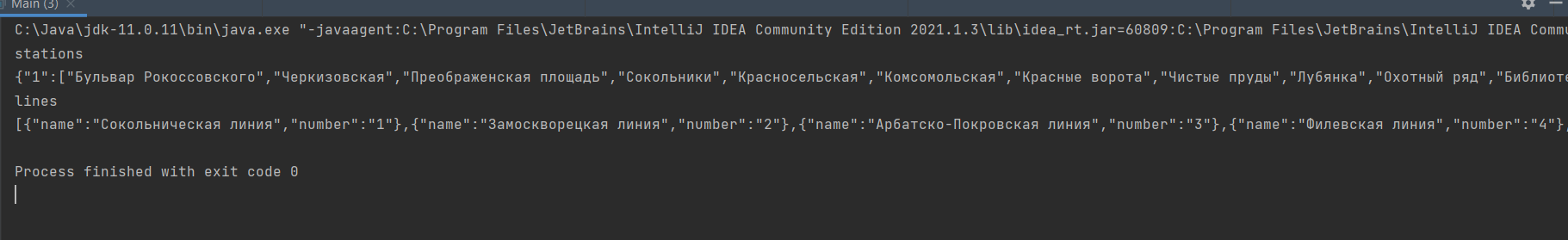
**Класс Main**

package ru.mirea.java.practice25.practice26;  
  
import org.json.JSONException;  
  
import java.io.IOException;  
  
public class Main {  
 public static void main(String[] args) throws IOException, JSONException {  
 ParserMetro parserMetro = new ParserMetro();  
 }  
}

**Класс ParserMetro**

package ru.mirea.java.practice25.practice26;  
  
import com.google.gson.\*;  
  
import org.json.JSONObject;  
import org.json.JSONException;  
  
import org.jsoup.Jsoup;  
import org.jsoup.nodes.Document;  
import org.jsoup.nodes.Element;  
import org.jsoup.select.Elements;  
  
import java.io.\*;  
  
public class ParserMetro {  
 private JsonObject listStatLine = new JsonObject();  
 private JsonObject stations = new JsonObject();  
 private JsonArray metroLines = new JsonArray();  
  
 public ParserMetro() throws IOException, JSONException {  
 Document document = null;  
 try {  
 document = Jsoup.connect("https://www.moscowmap.ru/metro.html#lines").timeout(10000).get();  
 } catch (IOException e) {  
  
 }  
  
 addStations(document);  
  
 addMetroLines(document);  
  
 CreateJsonFile();  
  
 }  
  
 public void CreateJsonFile() {  
  
 listStatLine.add("stations", stations);  
  
 listStatLine.add("lines", metroLines);  
  
 try {  
  
 FileWriter writer = new FileWriter("src/ru/mirea/java/practice25/practice26/metro.json");  
 writer.write(listStatLine.toString());  
 writer.close();  
  
 } catch (IOException e) {  
  
 }  
 for (String key : listStatLine.keySet()) {  
  
 System.out.println(key);  
  
 System.out.println(listStatLine.get(key).toString());  
  
 }  
 }  
  
 public void addStations(Document document) {  
  
 Elements metroStations = document.select("div.js-metro-stations");  
  
 for (Element station : metroStations) {  
 JsonArray stationsName = new JsonArray();  
 for (Element metro\_st : station.select("span.name"))  
 stationsName.add(metro\_st.text());  
  
 stations.add(station.attr("data-line"), stationsName);  
 }  
 }  
  
 public void addMetroLines(Document document) {  
  
 Elements lines = document.select("span.js-metro-line");  
  
 for (Element line : lines) {  
  
 JsonObject lineM = new JsonObject();  
 lineM.addProperty("name", line.text());  
 lineM.addProperty("number", line.attr("data-line"));  
 metroLines.add(lineM);  
 }  
 }  
}

**Тесты**



**Рисунок 2 – тестирование**

**Выводы**

Я изучил и научился парсить HTML-код страницы и находить в нем нужные с помощью селекторов.