|  |
| --- |
|  |
| 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 Баринов.И.В.

Принял Степанов П.В.

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

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

Отметка о выполнении

**Москва – 2021 г.**

## **Задание**

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

## **Ход Работы**

В ходе выполнения работы были получены следующие исходные коды:

package Practise25\_26;  
  
import java.io.EOFException;  
  
public class Main {  
 public static void main(String[] args) throws EOFException {  
 Parser parser = new Parser();  
 }  
}

---

package Practise25\_26;  
  
import com.google.gson.\*;  
import org.jsoup.Jsoup;  
import org.jsoup.nodes.Document;  
import org.jsoup.nodes.Element;  
import org.jsoup.select.Elements;  
  
import java.io.FileReader;  
import java.io.FileWriter;  
import java.io.IOException;  
  
public class Parser {  
 public Parser() {  
 Document document = null;  
 try {  
 document = Jsoup.*connect*("https://www.moscowmap.ru/metro.html#lines")  
 .userAgent("Chrome/4.0.249.0")  
 .referrer("http://www.google.com")  
 .get();  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
 JsonObject result = new JsonObject(), station = new JsonObject();  
 JsonArray lines = new JsonArray();  
 result.add("stations", station);  
 result.add("lines", lines);  
 Elements stations = document.select("div.js-metro-stations");  
 for (Element stat : stations) {  
 JsonArray st = new JsonArray();  
 for(Element el : stat.select("span.name")) st.add(el.text());  
 station.add(stat.attr("data-line"), st);  
 }  
 Elements lnes = document.select("span.js-metro-line");  
 for (Element line : lnes) {  
 JsonObject el = new JsonObject();  
 el.addProperty("number", line.attr("data-line"));  
 el.addProperty("name", line.text());  
 lines.add(el);  
 }  
 saveJSON(result.toString());  
 openJSON();  
 }  
  
 public void saveJSON(String json) {  
 try(FileWriter writer = new FileWriter("src/Practise25\_26/map.json", false)) {  
 writer.write(json);  
 writer.flush();  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
 }  
  
 public void openJSON() { //src/Practise25\_26/map.json  
 String json = "";  
 try(FileReader reader = new FileReader("src/Practise25\_26/map.json")) {  
 // читаем посимвольно  
 int c;  
 while((c=reader.read())!=-1){  
 json += (char)c;  
 }  
 }  
 catch(IOException ex){  
 ex.printStackTrace();  
 }  
 JsonObject jsonObject = (new JsonParser()).parse(json).getAsJsonObject();  
 JsonObject stations = (JsonObject) jsonObject.get("stations");  
 System.*out*.println("Количество станций на каждой линии: ");  
 for(String key : stations.keySet()) System.*out*.format("%s: %s\n",key, ((JsonArray)stations.get(key)).size());  
 }  
}

## **Вывод**

В ходе выполнения работы я научился получать данные из HTML страницы, а также создавать и считывать JSON файлы

GitHub: https://github.com/BarinovIvan/RuPractises/tree/master/src/Practise25\_26