



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

Деревянкин Н.А.

Принял

Степанов П.В.

Практическая работа выполнена «__» _____ 2021 г.

«_____»

«__» _____ 2021 г.

Отметка о выполнении

Москва – 2021 г.

СОДЕРЖАНИЕ

Цель работы	3
Задание	3
Репозиторий.....	3
Выполнение работы	3
Код выполненной работы.....	3
Тестирование программы.....	6
Вывод.....	7

Цель работы

Цель данной практической работы – Научиться получать данные из HTML-страницы, создание и чтение JSON-файлов.

Задание

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).

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.

Репозиторий

Ссылка:

https://github.com/dronikosha/JavaPractice/tree/master/src/ru/practice_25_to_26

Выполнение работы

Для выполнения данного практического задания первым делом нужно было скачать библиотеку GSON и воспользоваться уже скачанной, JSOUP. Мною был создан класс Parser, в котором я создал конструктор, осуществляющий парсинг самой страницы московского метро.

Также мною были реализованы два метода, один сохраняющий и заполняющий JSON файл, а второй открывающий и читающий с него.

Реализацию перечисленных методов и созданного класса можно посмотреть в главе “Код выполненной программы”.

Код выполненной работы

```

public Parser() {
    Document document = null;
    try {
        document = Jsoup.connect( url: "https://www.moscowmap.ru/metro.html#lines")
            .userAgent("Chrome/4.0.249.0 Safari/532.5")
            .referrer("http://www.google.com")
            .get();
    } catch (IOException e) {
        e.printStackTrace();
    }
    JsonObject result = new JsonObject(), station = new JsonObject();
    JsonArray lines = new JsonArray();
    result.add( property: "stations", station);
    result.add( property: "lines", lines);
    Elements stations = document.select( cssQuery: "div.js-metro-stations");
    for (Element stat : stations) {
        JsonArray st = new JsonArray();
        for(Element el : stat.select( cssQuery: "span.name")) st.add(el.text());
        station.add(stat.attr( attributeKey: "data-line"), st);
    }
    Elements lnes = document.select( cssQuery: "span.js-metro-line");
    for (Element line : lnes) {
        JsonObject el = new JsonObject();
        el.addProperty( property: "number", line.attr( attributeKey: "data-line"));
        el.addProperty( property: "name", line.text());
        lines.add(el);
    }
    saveJSON(result.toString());
    openJSON();
}

```

Рисунок 1 – Класс Parser и его конструктор

```

public void saveJSON(String json) {
    try(FileWriter writer = new FileWriter( fileName: "src/ru/practice_25_to_26/metro.json", append: false)) {
        writer.write(json);
        writer.flush();
    } catch (IOException e) {
        e.printStackTrace();
    }
}

```

Рисунок 2 – Метод saveJSON

```

public void openJSON() {
    String json = "";
    try(FileReader reader = new FileReader( fileName: "src/ru/practice_25_to_26/metro.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());
}

```

Рисунок 3 – Метод openJSON

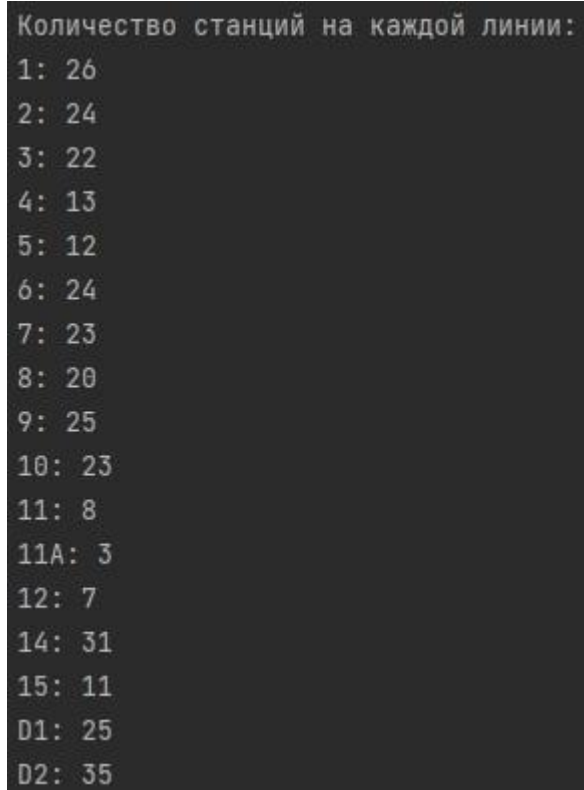
```

public class Main {
    public static void main(String[] args) throws EOFException {
        Parser parser = new Parser();
    }
}

```

Рисунок 4 – Запуск программы

Тестирование программы



```
Количество станций на каждой линии:  
1: 26  
2: 24  
3: 22  
4: 13  
5: 12  
6: 24  
7: 23  
8: 20  
9: 25  
10: 23  
11: 8  
11A: 3  
12: 7  
14: 31  
15: 11  
D1: 25  
D2: 35
```

Рисунок 5 – Результат вывода в консоль

```
1 {"stations":{
2   "1": [
3     "Бульвар Рокоссовского",
4     "Черкизовская",
5     "Преображенская площадь",
6     "Сокольники",
7     "Красносельская",
8     "Комсомольская",
9     "Красные ворота",
10    "Чистые пруды",
11    "Лубянка",
12    "Охотный ряд",
13    "Библиотека имени Ленина",
14    "Кропоткинская",
15    "Парк Культуры",
16    "Фрунзенская",
17    "Спортивная",
18    "Воробьевы горы",
19    "Университет",
20    "Проспект Вернадского",
21    "Юго-Западная",
22    "Тропарёво",
23    "Румянцево",
24    "Саларьево",
25    "Филатов Луг",
26    "Прокшино",
```

Рисунок 6 – Результат записи в JSON файл

Вывод

В результате выполнения данной практической работы я научился получать данные из HTML-страницы, создание и чтение JSON-файлов, и смог создать программу, реализующую это на языке программирования Java.