

## Задание 1. Напишите программу, которая будет читать JSON-файл в одном формате, а записывать — в другом

- В рамках этого задания вам предстоит написать код, который будет парсить [JSON-файл](#) из проекта SPBMetro и сохранять его в другом формате тоже в виде JSON-файла, содержащего информации о линиях: номер, название и количество станций. Эти данные уже содержатся в исходном JSON-файле:
  - Информация о номере и названии станций содержится в его конце в виде массива объектов вида:

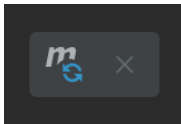
```
{  
  "number": 1,  
  "name": "Кировско-Выборгская",  
  "color": "red"  
}
```

Из данных объектов вам нужно будет удалить ключи “color”.

- Информация о количестве станций на каждой содержится в первой части JSON-файла.
- Создайте новый Maven-проект JSONHandler, создайте также в нём класс Main с методом main, в котором после закрывающей круглой скобки напишите “throws Exception”.
- Подключите к вашему проекту библиотеку Jackson при помощи Maven (<https://mvnrepository.com/artifact/com.fasterxml.jackson.core/jackson-databind>). Перейдите в документации на страницу последней версии библиотеки и скопируйте оттуда код для вставки в файл pom.xml, например:

```
<dependency>  
  <groupId>com.fasterxml.jackson.core</groupId>  
  <artifactId>jackson-databind</artifactId>  
  <version>2.13.4</version>  
</dependency>
```

- Поместите этот код в тэг <dependencies></dependencies> файла pom.xml вашего проекта. Если слова “com.fasterxml.jackson.core”, “jackson-core” и номер версии отображаются в коде XML-файла красным, нажмите на кнопку обновления (“Load Maven Changes”):



- Откройте [JSON-файл](#) из проекта SPBMetro и ознакомьтесь с его структурой. Вам необходимо создать класс, структура которого будет соответствовать структуре в данном JSON-файле.
- В методе “main” напишите код, который будет читать JSON-файл в строку:

```
String jsonFile = Files.readString(  
    Paths.get("/path/to/file/map.json")  
);
```

- Напишите код, который будет парсить JSON-данные в объект класса JsonNode:

```
JsonNode jsonData = objectMapper.readTree(jsonFile);
```

- Получите из этого объекта данные о станциях на каждой линии, а также массив самих линий следующим образом:

```
JsonNode stations = jsonData.get("stations");  
JsonNode lines = jsonData.get("lines");
```

- Напишите код, который обойдёт массив с линиями и получит на основе каждой линии объект, который можно будет изменять:

```
for(JsonNode line : lines) {  
    ObjectNode lineNode = (ObjectNode) line;  
}
```

- В документации к библиотеке Jackson найдите способы удаления и установки значений в объектах класса ObjectNode. Удалите из них ключ “color” и добавьте ключи “stationsCount” со значениями, равными количеству станций на каждой из линий.
- Напишите код, который запишет изменённый объект “lines” в отдельный JSON-файл. Пример:

```
ObjectMapper mapper = new ObjectMapper();  
File output = new File("export.json");  
mapper.writeValue(output, lines);
```

- Запустите получившуюся программа и проверьте, что в результате её выполнения записывается JSON-файл, содержащий массив объектов-станций следующего формата:

```
{  
  "number": 1,  
  "name": "Кировско-Выборгская",  
  "stationsCount": 19  
}
```

- Если что-то не получилось, постарайтесь самостоятельно добиться необходимого результата.
- В случае возникновения трудностей, вы, как и всегда в заданиях для самостоятельного выполнения, можете воспользоваться рекомендациями под видео, под которым была указана ссылка на это задание.
- Если всё получилось, поздравляем! Теперь вы умеете парсить и записывать файлы формата JSON.