

Если вам не удалось выполнить задание, сравните получившийся у вас код файла pom.xml с эталонным:

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/xsd/maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>

    <groupId>org.example</groupId>
    <artifactId>JSONHandler</artifactId>
    <version>1.0-SNAPSHOT</version>

    <properties>
        <maven.compiler.source>17</maven.compiler.source>
        <maven.compiler.target>17</maven.compiler.target>
    </properties>

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

</project>
```

Также сравните основной код приложения с эталонным:

```
import com.fasterxml.jackson.databind.JsonNode;
import com.fasterxml.jackson.databind.ObjectMapper;
import com.fasterxml.jackson.databind.node.ObjectNode;

import java.io.File;
import java.nio.file.Files;
import java.nio.file.Paths;

public class Main {
    public static void main(String[] args) throws Exception {
        ObjectMapper objectMapper = new ObjectMapper();
        String jsonFile =
Files.readString(Paths.get("/Users/sortedmap/Desktop/map.json
"));
        JsonNode jsonData = objectMapper.readTree(jsonFile);
        JsonNode stations = jsonData.get("stations");
        JsonNode lines = jsonData.get("lines");
```

```

        for(JsonNode line : lines) {
            ObjectNode lineNode = (ObjectNode) line;
            lineNode.remove("color");
            String lineNumber = line.get("number").asText();
            JsonNode stationsList = stations.get(lineNumber);
            int stationsCount = stationsList.size();
            lineNode.put("stationsCount", stationsCount);
        }

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

```

Внесите в свой код изменения таким образом, чтобы он точно соответствовал эталонному коду выше. Запустите и проверьте, как работает программа. Убедитесь, что итоговый JSON-файл выглядит следующим образом (показано в отформатированном виде):

```

[
  {
    "number": 1,
    "name": "Кировско-Выборгская",
    "stationsCount": 19
  },
  {
    "number": 2,
    "name": "Московско-Петроградская",
    "stationsCount": 18
  },
  {
    "number": 3,
    "name": "Невско-Василеостровская",
    "stationsCount": 12
  },
  {
    "number": 4,
    "name": "Правобережная",
    "stationsCount": 8
  },
  {
    "number": 5,
    "name": "Фрунзенско-Приморская",
    "stationsCount": 12
  }
]

```

**Примечание.** Чтобы отформатировать JSON-файл в среде разработки IntelliJ IDEA, в меню сверху выберите пункт “Code” и нажмите “Reformat Code”.

Поздравляем! Теперь вы знаете, как парсить и записывать файлы формата JSON!