

# Тестовое задание Java-разработчик

## Входные данные:

1. CSV файл.

*Назначение столбцов:*

Идентификатор ордера, сумма, валюта, комментарий

*Пример записи:*

1,100,USD,оплата заказа

*Примечание:* все столбцы обязательны

2. JSON файл.

*Пример записи:*

```
{"orderId":1,"amount":100,"currency":"USD","comment":"оплата заказа"}
```

*Примечание:* все поля обязательны

## Выходные данные:

```
{"id":1, "amount":100, "comment":"оплата заказа", "filename":"orders.csv", "line":1, "result":"OK" }
```

- id - идентификатор ордера
- amount - сумма ордера
- currency - валюта суммы ордера
- comment - комментарий по ордеру
- filename - имя исходного файла
- line - номер строки исходного файла
- result - результат парсинга записи исходного файла.
  - OK - если запись конвертирована корректно,
  - или описание ошибки если запись не удалось конвертировать.

## Описание задания:

Необходимо разработать приложение парсинга входящих данных и конвертирования результат парсинга в результирующий формат.

1. Приложение должно быть реализовано с использованием фреймворка **Spring**.
2. Исходные код приложения должен быть оформлен в виде **maven** проекта и размещён на **GitHub**. Допускается использовать зависимости **только из публичных репозиториев**.
3. Сборка конечного приложения должна быть выполнена командой:  
**mvn clean install**
4. Приложение должно быть консольным.  
Пример команды запуска: **java -jar orders\_parser.jar orders1.csv orders2.json**  
- где **orders1.csv** и **orders2.json** файлы для парсинга.
5. Результат выполнения должен выводиться в **stdout** поток.  
Примечание: в **stdout** должны попасть только выходные данные, логов там быть не должно.
6. Парсинг и конвертирование должны выполняться параллельно в несколько потоков.
7. Необходимо предусмотреть корректную обработку ошибок в исходных файлах.  
Например, вместо числа в файле может быть строковое значение в поле **amount**.
8. Разрешается использовать инструменты языка не выше Java 8.
9. Необходимо учесть возможность добавления новых форматов входящих данных. Например: XLSX