

# Реализация библиотеки для поточковой записи файлов формата XLSX

**Свитков Сергей**

группа 344

научный руководитель: к.т.н, доц., Ю.В. Литвинов

консультант: нач. отд. разр. ПО, "НТЦ Протей",

М.В. Заведеев

СПбГУ

кафедра системного программирования

23 мая 2018 г.

# Введение

- ▶ Веб-приложения в сфере биллинга, телекоммуникаций
- ▶ Различные отчеты, статистика
- ▶ Формат XLSX
- ▶ Потокное формирование файла для экономии используемой RAM

# Постановка задачи

Цель: разработка библиотеки для потоковой записи файлов формата XLSX и сравнение её с существующими реализациями. Для достижения этой цели были поставлены следующие задачи:

- ▶ Сформулировать алгоритм, который будет использоваться для генерации файлов формата XLSX в потоковом режиме
- ▶ Реализовать предложенный алгоритм в виде переиспользуемой библиотеки с открытым исходным кодом, документацией, примерами использования и артефактом в Maven
- ▶ Сравнить полученную реализацию с существующими библиотеками по потреблению RAM (оперативной памяти) и скорости работы при создании документа

# Обзор существующих решений

- ▶ Apache POI
  - ▶ До версии 3.8 — только in-memory
  - ▶ Начиная с 3.8 — Stream-API для записи, Event-API для чтения
  - ▶ Часть операций всё равно только in-memory
  - ▶ Отсутствие полных и подробных примеров работы с Stream-API
  - ▶ Отсутствие поддержки автоматического разбиения документа на страницы
- ▶ SJXLSX
  - ▶ Не поддерживается с 2015 года
  - ▶ Отсутствие документации
  - ▶ Не поддерживается потоковая запись файлов

# Итоги обзора

- ▶ Apache POI поддерживает потоковую генерацию файлов, но всё же имеет ряд недостатков
- ▶ Реализовать свою библиотеку
- ▶ Сравнить её с рассмотренными реализациями

# XLSX

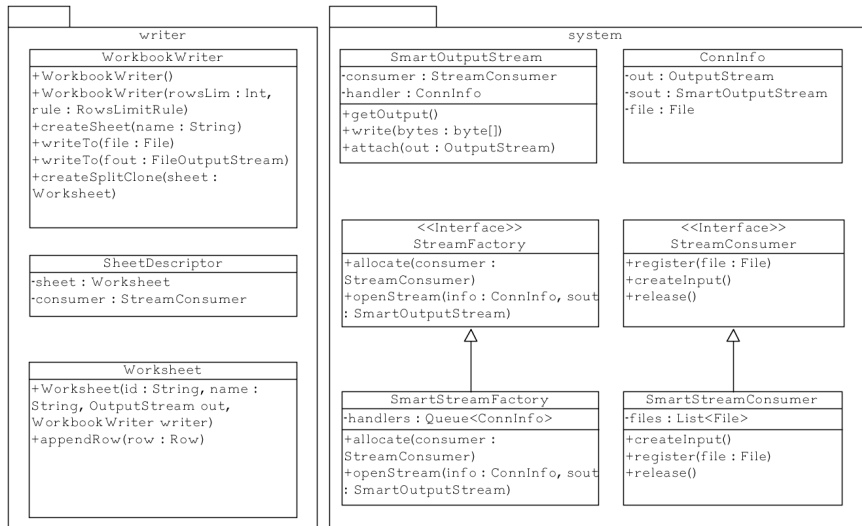
- ▶ zip-архив с xml файлами
- ▶ Структура:
  - ▶ Content\_Types.xml — типы содержимого в архиве и пути к ним
  - ▶ \_rels — зависимости между файлами внутри архива
  - ▶ docProps — метаданные: имя автора, дата создания, ...
  - ▶ xl — директория с основными файлами архива: workbook, страницы, стили, таблицы
    - ▶ Workbook — метаданные, ссылки на страницы с данными
    - ▶ Worksheets — страницы с данными

## Реализация: алгоритм

- ▶ Для каждой страницы создавать временный файл
- ▶ Хранить в RAM только один ряд (во время создания)
- ▶ После создания добавлять ряд во временный файл страницы
- ▶ После завершения формирования документа — записывать данные из временных файлов в основной файл
- ▶ Для экономии дискового пространства сжимать временные файлы

# Реализация: архитектура

## Упрощённая архитектура библиотеки:

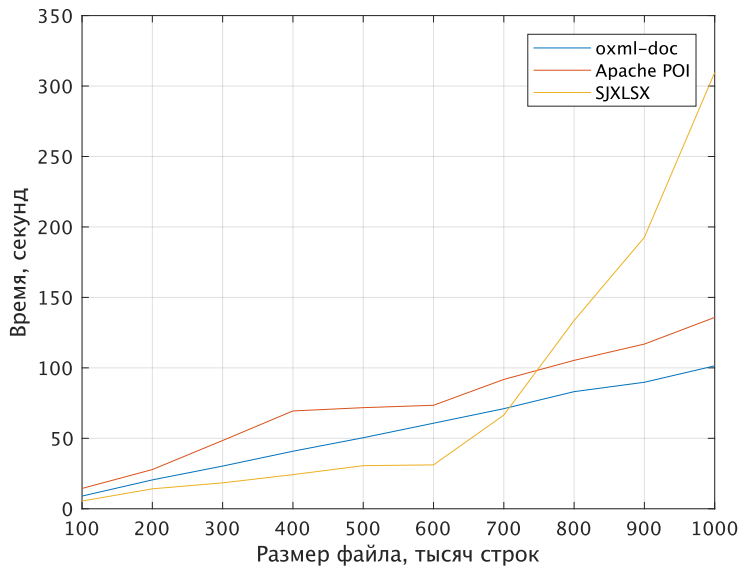




# Эксперименты: RAM



# Эксперименты: скорость записи



# Результаты

В ходе данной работы были достигнуты следующие результаты:

- ▶ Реализован предложенный в рамках работы алгоритм записи файлов формата XLSX
- ▶ Реализация оформлена в виде библиотеки для потоковой записи файлов формата XLSX с открытым исходным кодом и артефактом в Maven
- ▶ Проведено сравнение реализации с существующими библиотеками по количеству используемой RAM и скорости работы при создании документа