

Работа с Excel на Java с помощью Apache POI

Семченко Регина Викторовна

Приамурский государственный университет имени Шолом-Алейхема

Студент

Еровлев Павел Андреевич

Приамурский государственный университет имени Шолом-Алейхема

Студент

Аннотация

В данной статье рассмотрена возможность работы с Excel файлами с помощью библиотеки Apache POI. В ходе предсталения работы будет продемонстрирована возможность создания файлов, редактирование строк и столбцов, запись и удаление.

Ключевые слова: Excel, Java, Apache POI

Working with Excel in Java with Apache POI

Semchenko Regina Viktorovna

Sholom-Aleichem Priamursky State University

Student

Erovlev Pavel Andreevich

Sholom-Aleichem Priamursky State University

Student

Abstract

This article discusses the ability to work with Excel files using the Apache POI library. The presentation of the work will demonstrate the ability to create files, edit rows and columns, write and delete.

Keywords: Excel, Java, Apache POI

Задача проекта Apache POI - создавать и поддерживать API Java для управления различными форматами файлов на основе стандартов Office Open XML (OOXML) и формата составных документов Microsoft OLE 2 (OLE2). Главная функция это чтение и запись файлов MS Excel с помощью Java. Кроме того, можно читать и писать файлы MS Word и MS PowerPoint с помощью Java.

Цель данной статьи рассмотреть возможности Apache POI на работе с документами Excel.

В своей работе М.К.Ермаков, С.П.Вартанов рассмотрели вопросы проведения анализа программ интерпретируемых языков программирования

[1]. А.А.Шейн, Д.Г.Залевский, С.В. Автайкин, С.В.Карташев, С.А.Скороход разработали и описали действия программы предназначенной для автоматического создания набора классов для представления объектов модели Decode в виде нативных объектов языка Java [2]. С.В. Мельников провел обзор на работу с отладочным интерфейсом Java и методом модификации функциональности приложения, не изменяющий его бинарные файлы [3]. Так же А.А. Птицын, Н.Л. Подколотный, Д.А.Григорович, С.В. Лаврюшев разработали молекулярно-биологический сервер и на его базе создали ряд информационно-вычислительных систем, для изучения регуляции экспрессии генов с использованием новейших технологий Java [4]. Н.Н. Глибовец проанализировала в своей работе особенности агентных технологий и перспективы их использования для разработки сложных многопользовательских программных систем [5].

Основные интерфейсы включают «Workbook», «Sheet», «Row» и «Cell». Для базового форматирования используйте интерфейсы «CellStyle» и «Font». Конкретные классы реализации включают:

- Excel 2003: HSSFWorkbook, HSSFSheet, HSSFRow, HSSFCell.
- Excel 2007: XSSFWorkbook, XSSFSheet, XSSFRow, XSSFCell.

Теперь рассмотрим на примерах работу. Следующий фрагмент кода демонстрирует запись списка книг в файл Excel (рис.1).

```

1 package net.codejava.excel;
2
3 import java.io.FileOutputStream;
4 import java.io.IOException;
5
6 import org.apache.poi.ss.usermodel.Cell;
7 import org.apache.poi.ss.usermodel.Row;
8 import org.apache.poi.xssf.usermodel.XSSFSheet;
9 import org.apache.poi.xssf.usermodel.XSSFWorkbook;
10
11 public class SimpleExcelWriterExample {
12
13     public static void main(String[] args) throws IOException {
14         XSSFWorkbook workbook = new XSSFWorkbook();
15         XSSFSheet sheet = workbook.createSheet("Запись Книг ");
16
17         Object[][] bookData = {
18             {"Название 1", "Автор 1", 79},
19             {"Название 2", "Автор 2", 36},
20             {"Название 3", "Автор 3", 42},
21             {"Название 4", "Автор 4", 35},
22         };
23
24         int rowCount = 0;
25
26         for (Object[] aBook : bookData) {
27             Row row = sheet.createRow(++rowCount);
28
29             int columnCount = 0;
30
31             for (Object field : aBook) {
32                 Cell cell = row.createCell(++columnCount);
33                 if (field instanceof String) {
34                     cell.setCellValue((String) field);
35                 } else if (field instanceof Integer) {
36                     cell.setCellValue((Integer) field);
37                 }
38             }
39
40         }
41
42         try (FileOutputStream outputStream = new FileOutputStream("JavaBooks.xlsx")) {
43             workbook.write(outputStream);
44         }
45     }
46 }
47
48 }
49

```

Рисунок 1 – Создание файла

Далее создается файл «JavaBooks.xlsx» (Рис.2).

	A	B	C	D	E
1					
2		Название 1	Автор 1	79	
3		Название 2	Автор 2	36	
4		Название 3	Автор 3	42	
5		Название 4	Автор 4	35	
6					

Рисунок 2 – файл.xlsx

Для того чтобы сделать код более гибким и простым в расширении, перепишем код основываясь на подходе объектно-ориентированном программировании.

Для начала создадим класс «Book.java» и опишем переменные (рис3).

```
1 package net.codejava.excel;
2
3 public class Book {
4     private String title;
5     private String author;
6     private float price;
7
8     public Book() {
9     }
10
11    public Book(String title, String author, double price) {
12        this.title = title;
13        this.author = author;
14        this.price = price;
15    }
16
17 }
```

Рисунок 3 – Запись переменных

Далее напишем метод, который записывает данные в файл excel (рис.4).

```
1 public void writeExcel(List<Book> listBook, String excelFilePath) throws IOException {
2     Workbook workbook = new HSSFWorkbook();
3     Sheet sheet = workbook.createSheet();
4
5     int rowCount = 0;
6
7     for (Book aBook : listBook) {
8         Row row = sheet.createRow(++rowCount);
9         writeBook(aBook, row);
10    }
11
12    try (FileOutputStream outputStream = new FileOutputStream(excelFilePath)) {
13        workbook.write(outputStream);
14    }
15 }
```

Рисунок 4 – Метод записи в файл

Следующим шагом напишем метод позволяющий записывать данные в ячейки (рис.5).

```
1 private void writeBook(Book aBook, Row row) {
2     Cell cell = row.createCell(1);
3     cell.setCellValue(aBook.getTitle());
4
5     cell = row.createCell(2);
6     cell.setCellValue(aBook.getAuthor());
7
8     cell = row.createCell(3);
9     cell.setCellValue(aBook.getPrice());
10 }
```

Рисунок 5 – Запись в ячейки

И теперь напишем массив, где будут данные, которые запишутся в ячейки (рис.6).

```
1 private List<Book> getListBook() {
2     Book book1 = new Book("Книга 1", "Автор 1", 79);
3     Book book2 = new Book("Книга 2", "Автор 2", 36);
4     Book book3 = new Book("Книга 3", "Автор 3", 42);
5     Book book4 = new Book("Книга 4", "Автор 4", 35);
6
7     List<Book> listBook = Arrays.asList(book1, book2, book3, book4);
8
9     return listBook;
10 }
```

Рисунок 6 – Массив данных

Теперь рассмотрим возможность сделать небольшое форматирование ячеек, добавить шрифт, размер. Напишем код, который будет записывать строку заголовка для документа с полужирным шрифтом и размером 16 (рис.7).

```
1 private void createHeaderRow(Sheet sheet) {
2
3     CellStyle cellStyle = sheet.getWorkbook().createCellStyle();
4     Font font = sheet.getWorkbook().createFont();
5     font.setBold(true);
6     font.setFontHeightInPoints((short) 16);
7     cellStyle.setFont(font);
8
9     Row row = sheet.createRow(0);
10    Cell cellTitle = row.createCell(1);
11
12    cellTitle.setCellStyle(cellStyle);
13    cellTitle.setCellValue("Title");
14
15    Cell cellAuthor = row.createCell(2);
16    cellAuthor.setCellStyle(cellStyle);
17    cellAuthor.setCellValue("Author");
18
19    Cell cellPrice = row.createCell(3);
20    cellPrice.setCellStyle(cellStyle);
21    cellPrice.setCellValue("Price");
22 }
```

Рисунок 7 – Форматирование ячеек

В данной статье было рассмотрено создание, редактирование, добавление и форматирование ячеек и книг Excel.

Библиографический список

1. Ермаков М.К., Варганов С.П. Подход к проведению динамического анализа java-программ методом модификации виртуальной машины java // Научные труды Винницкого национального технологического университета. 2018. №6. С. 10-17.
2. Шейн А.А., Залевский Д.Г., Автайкин С.В., Карташев С.В., Скороход С.А. Генератор исходного кода на языке java по описанию бортовых компонентов decode (decode java generator 0.2) // Вестник волжского университета им. в.н. татищева. 2019. №3. С. 26-32.
3. Мельников С.В. Обзор и применение отладочного интерфейса java (jdi) для обратимой модификации программных продуктов // Современные проблемы науки и образования. 2018. №8. С. 8-19.
4. Птицын А.А., Подколотный Н.Л., Григорович Д.А., Лаврюшев С.В.

Создание молекулярно-биологического сервера www с использованием новейших технологий java // Заметки по информатике и математике. 2020. №1. С. 11-20.

5. Глибовец Н.Н. Использование jade (java agent development environment) для разработки компьютерных систем поддержки дистанционного обучения агентного типа // Заметки по информатике и математике. 2019. №10. С. 15-20.