Apache POI

**Nhiệm vụ:**

Tìm hiểu về Apache POI dùng để xử lý tập tin Microsoft Excel trên Java.

**Giáo viên hướng dẫn:**

Khuất Thùy Phương

**Học** **viên thực hiện:**

Trần Ngọc Đản



Tháng 9/2016

Trung tâm tin học

Đại học Khoa học Tự nhiên

Mục lục

[I. Tổng quan 4](#_Toc462833151)

[II. Java Excel APIs 6](#_Toc462833152)

[III. Core Classes 7](#_Toc462833153)

[**1.** **Workbook** 7](#_Toc462833154)

[**2.** **Sheet** 7](#_Toc462833155)

[**3.** **Row** 7](#_Toc462833156)

[**4.** **Cell** 7](#_Toc462833157)

[**5.** **CellStyle** 8](#_Toc462833158)

[**6.** **Color** 8](#_Toc462833159)

[**7.** **XSSFFont** 8](#_Toc462833160)

[**8.** **XSSFHyperlink** 8](#_Toc462833161)

[**9.** **XSSFCreationHelper** 8](#_Toc462833162)

[**10.** **XSSFPrintSetup** 8](#_Toc462833163)

[IV. Thiết lập Java Excel APIs 9](#_Toc462833164)

[**1.** **Workbook** 12](#_Toc462833165)

[**Create Blank Workbook** (Tạo workbook rỗng) 12](#_Toc462833166)

[**Open Existing Workbook** (mở một workbook đã tồn tại) 13](#_Toc462833167)

[**2.** **Spreadsheets** (bảng tính) 14](#_Toc462833168)

[**Create a Spreadsheet** (tạo bảng tính) 14](#_Toc462833169)

[**Write into a Spreadsheet** (ghi nội dung vào bảng tính) 14](#_Toc462833170)

[**Read from a Spreadsheet** (Đọc dữ liệu từ một bảng tính) 17](#_Toc462833171)

[**3.** **Cells (ô)** 19](#_Toc462833172)

[**Create a Cell** (tạo cell) 19](#_Toc462833173)

[**Types of Cells (kiểu dữ liệu của cell)** 19](#_Toc462833174)

[**Cell Styles (Định dạng phong cách trình bày của cell)** 21](#_Toc462833175)

[**4.** **Formula** (công thức) 26](#_Toc462833176)

[Tính tổng 2 số 27](#_Toc462833177)

[Tính mũ 27](#_Toc462833178)

[Tính giá trị lớn nhất giữa 2 số 27](#_Toc462833179)

[Tính giai thừa 27](#_Toc462833180)

[Tính căn bậc 2 28](#_Toc462833181)

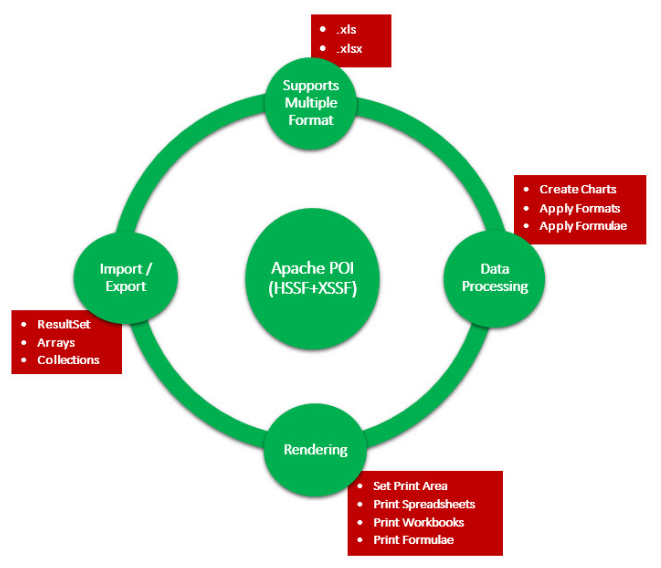
[VI. Kết luận 29](#_Toc462833182)

[VII. Tham khảo 30](#_Toc462833183)

1. **Tổng quan**

**The Apache POI Project**

Apache POI là thư viện mã nguồn mở 100% được cung cấp bởi Apache Software Foundation. Hầu hết các nhà phát triển ứng dụng vừa và nhỏ thích sử dụng Apache POI (HSSF + XSSF). Nó hỗ trợ tất cả những đặc tính cơ bản của thư viện Excel; tuy nhiên, rending và text extraction là những đặc tính chính của nó.



Nhiệm vụ của dự án Apache POI là để cung cấp những Java API cho việc thao tác với những định dạng tập tin khác nhau dựa trên **Office Open XML standards (OOXML)** và **Microsoft's OLE 2 Compound Document format (OLE2)**. Một cách ngắn gọn, dựa vào API này, chúng ta có thể đọc, ghi tập tin Microsoft Office Excel bởi Java. Thêm vào đó API này cũng hỗ trợ đọc và ghi MS Word và MS PowerPoint.

Vậy OOXML và OLE2 là định dạng gì?

* OLE2 là định dạng bao gồm hầu hết những tập tin thuộc Microsoft Office như XLS, DOC và PPT.
* Office OpenXML Format là định dạng mới dựa trên định dạng của tập tin XML. Những tập tin này có trong Microsoft Office 2007, 2008 bao gồm những tập tin XLSX, DOCX và PPTX.

Những ứng dụng java liên quan đến MS Office đều cố gắng có một thành phần cung cấp API để xử lý tài liệu có định dạng OLE2 và OOXML. API này hầu như được phát triển cho tập tin [MS Excel (SS = HSSF + XSSF)](http://poi.apache.org/spreadsheet/index.html). Công việc đang được hoàn thiện là làm cho cả tập tin [Word (WP=HWPF+XWPF)](http://poi.apache.org/document/index.html) và tập tin [PowerPoint (SL=HSLF+XSLF).](http://poi.apache.org/slideshow/index.html) Dực án nãy cũng hộp trợ cho  [Visio (HDGF and XDGF)](http://poi.apache.org/diagram/index.html), [TNEF (HMEF)](http://poi.apache.org/hmef/index.html), và [Publisher (HPBF)](http://poi.apache.org/hpbf/index.html).

1. **Java Excel APIs**

POI – HSSF và POI – XSSF là hai Java API cho phép truy cập những tập tin định dạng Microsoft Excel.

* HSSF là API dành cho định dạng tập tin excel có đuôi mở rộng là .xls, có trong các phiên bản tiền nhiệm của Microsoft Office 2007. Dữ liệu được sắp xếp theo tứng record.
* XSSF là API dành cho định dạng tập tin Excel 2007 trở về sau, có đuôi mở rộng là .xlsx. Dữ liệu được chứa trong những đối tượng.

HSSF và XSSF cung cấp những phương thức để thao tác trên bảng tính, ví dụ: tạo, chỉnh sửa, đọc, ghi bảng tính. Cụ thể, cung cấp:

* Cấu trúc cấp thấp cho những nhu cầu đặc biệt.
* Một API mô hình sự kiện (eventmodel) cho việc truy cập chỉ đọc (read-only).
* Một API mô hình người dùng để tạo, đọc và thay đổi những tập tin XLS.

Từ phiên bản POI 3.8 beta 3, có thêm POI – SXSSF là một mở rộng của XSSF, được sử dụng cho bảng tính lớn và bộ nhớ heap bị giới hạn. SXSSF chiếm ít bộ nhớ nhờ hạn chế sự truy cập vào hàng chỉ trong một cửa số trượt, trong khi XSSF cho phép truy cập đến tất cả các hàng trong tài liệu.

1. **Core Classes**
2. **Workbook**

Workbook là super – interface của tất cả những class cho phép tạo và duy trì bảng tính excel. Nó thuộc về package **org.apache.poi.ss.usermodel.** Hai class implements interface này là:

* **HSSFWorkbook** : Class này có những phương thức đọc và ghi tập tin Microsoft Exel có đuôi mở rộng là .xls. Tương thích với MS-Office versions 97–2003.
* **XSSFWorkbook** :: Class này có những phương thức đọc, ghi tập tin Microsoft Exel và OpenOffice xml có đuôi mở rộng là .xls hoặc .xlsx. Tương thích với MS-Office versions 2007 or later.

1. **Sheet**

Sheet là một interface thuộc package **org.apache.poi.ss.usermodel.** Nó là một super – interface của những lớp dùng để tạo bảng tính với một tên gọi cụ thể. Bảng tính là một lưới các cell sắp xếp có thứ tự. Hai trong số những class implements interface này là:

* **HSSFSheet**: class này có những phương thức tạo bảng tính excel, định dạng bảng tính và dữ liệu của bảng tính (xls).
* **XSSFSheet**: class này có chức năng giống HSSFSheet nhưng thể hiện định dạng cấp cao của tập tin excel (xlsx).

1. **Row**

Row là một interface thuộc package **org.apache.poi.ss.usermodel.** Nó là một super – interface của tất cả các lớp đại diện cho rows trong thư viện POI. Hai trong số những class implements interface này: **HSSFRow** và **XSSFRow** - class thuộc package **org.apache.poi.xssf.usermodel**.

1. **Cell**

Cell là một interface thuộc package **org.apache.poi.ss.usermodel.** Nó là một super – interface của các lớp đại diện cho cells trong rows của một spreadsheet.

Cell có thể chứa dữ liệu kiểu số (numeric), kiểu lỗi (error), kiểu chuỗi (string),…

Hai trong số những class implements interface này: **FSSFCell** và **XSSFCell.**

1. **CellStyle**

**CellStyle** là một interface được implements bởi class **HSSFCellStyle** và **XSSFCellStyle** - là class thuộc package **org.apache.poi.xssf.usermodel**. Cung cấp những thông tin về định dạng nội dung của một cell trong bảng tính. Nó cũng cung cấp những lựa chọn để thay đổi định dạng cell.

1. **Color**

Color là một interface sử dụng cho định dạng màu sắc của cells. Được implements bởi 2 class:

* **HSSFColor** là một class thuộc package **org.apache.poi.hssf.util**.
* **XSSFColor** là một class thuộc package **org.apache.poi.xssf.usermodel**.

1. **XSSFFont**

**XSSFFont** là một class thuộc package **org.apache.poi.xssf.usermodel.** Nó implements **Font** interface, được sử dụng để thay đổi những fonts khác nhau của một workbook.

1. **XSSFHyperlink**

**XSSFHyperlink** là mọt class thuộc package **org.apache.poi.xssf.usermodel.** Nó implements **Hyperlink** interface. Được sử dụng để thiết lập một hyperlink đến nội dung của cell trong một bảng tính.

1. **XSSFCreationHelper**

**XSSFCreationHelper** là một class thuộc package **org.apache.poi.xssf.usermodel**. Nó implements **CreationHelper** interface. Được sử dụng sử một lớp hỗ trợ cho việc định giá biểu thức (formula evaluation) và cài đặt hyperlinks. (Trong ví dụ ở dưới đây sẽ giải thích tại sao phải sử dụng class này).

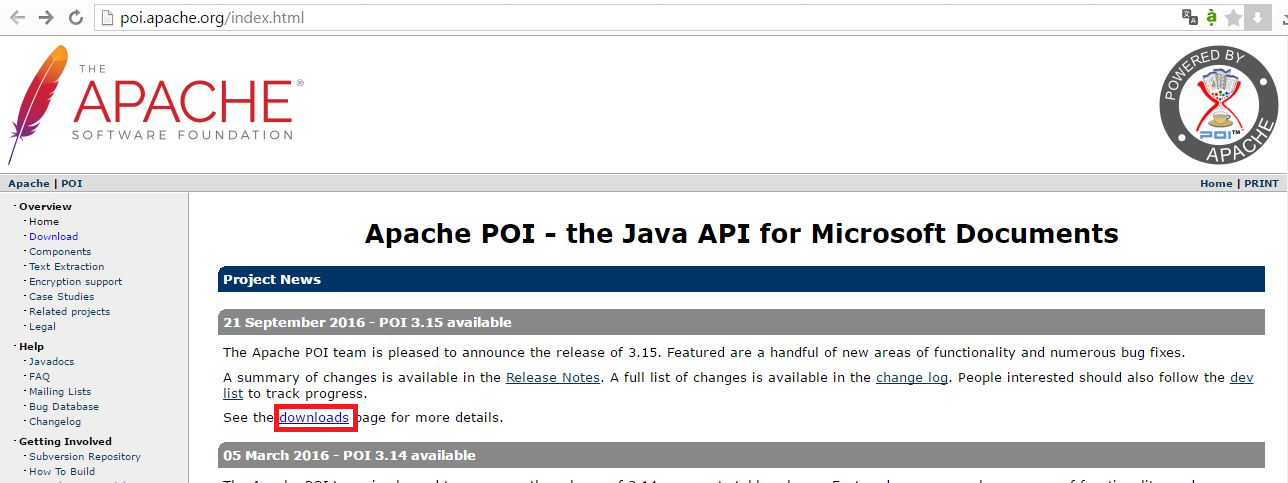
1. **XSSFPrintSetup**

**XSSFPrintSetup** là một class thuộc package **org.apache.poi.xsssf.usermodel.** Nó implements **PrintSetup** interface. Được sử dụng để thiết lập kích thước trang in (print page size), area, options và settings.

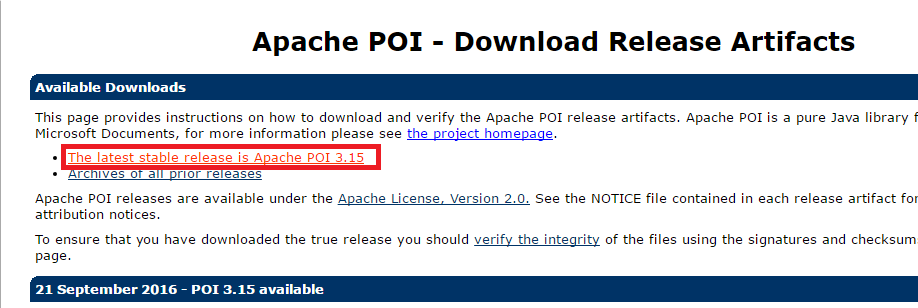
1. **Cấu hình Java Excel APIs**

Trước khi sử dụng API cần thêm các thư viện từ gói API vào project.

**Bước** 1: Download Apache POI API từ <http://poi.apache.org/>



Chọn download để tìm phiên bản API mong muốn (ở đây, tôi download bản mới nhất POI 3.15). Sau đó giải nén gói API vừa download.

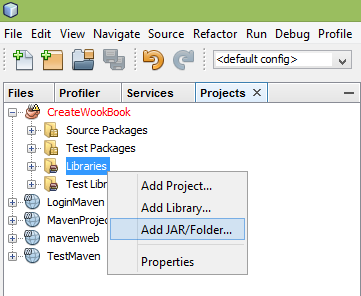


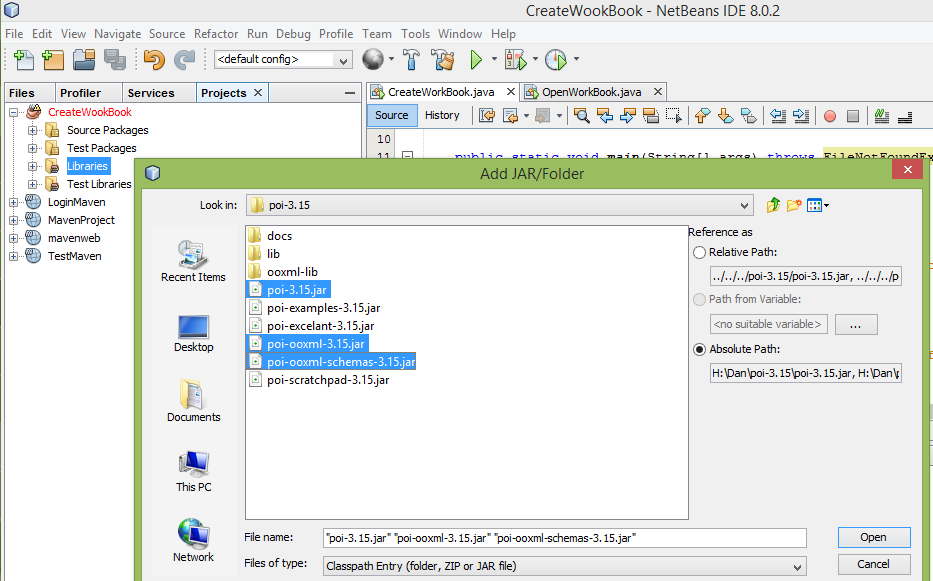
**Bước** 2: Thêm các file thư viện từ gói API vừa download vào project của mình:

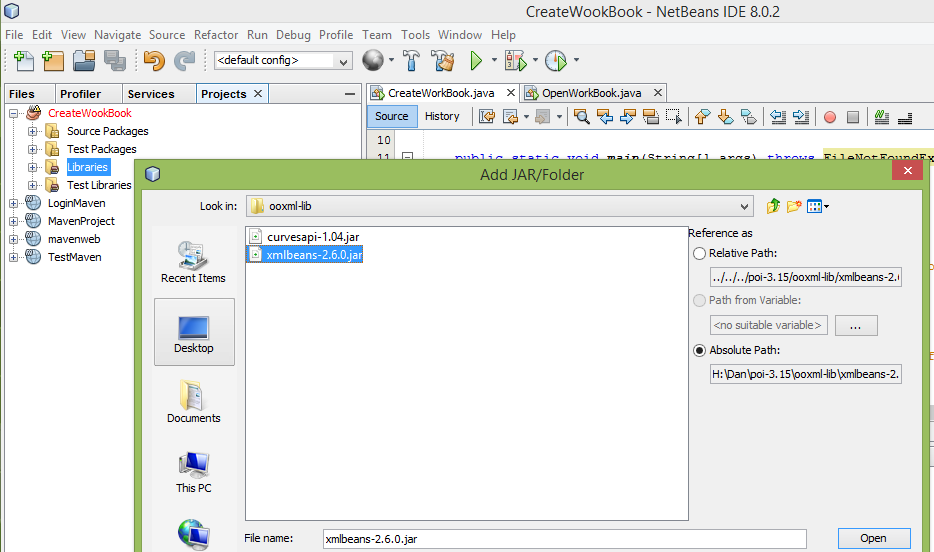
\*IDE tôi sử dụng ở đây là NetBeans.

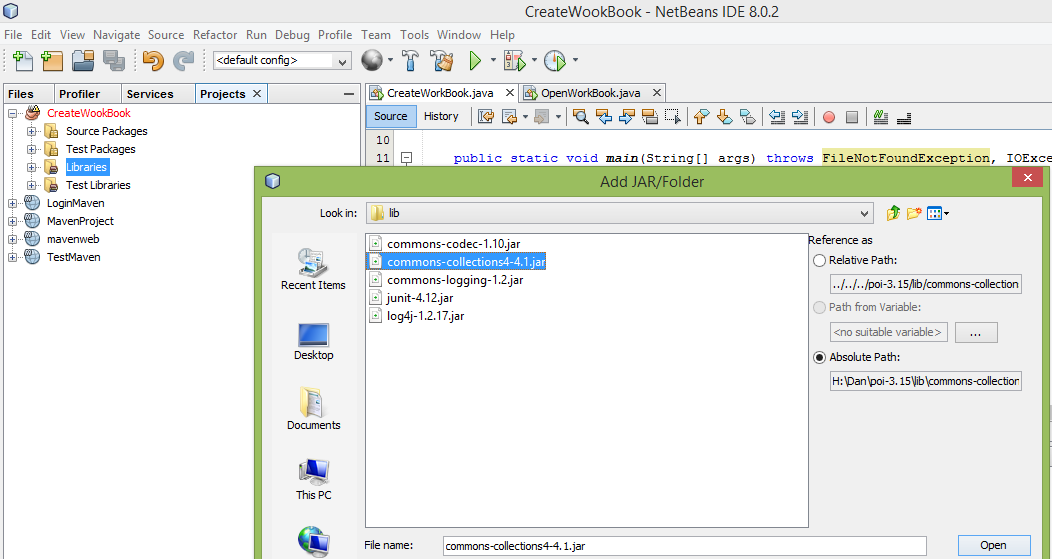
Sau khi tạo project, thêm các file jar từ gói API vào thư mục Libraries.

\*Một số thư viện sau đây chỉ mang tính minh họa, nếu không cần thiết cho project của bạn, thì có thể không thêm vào.



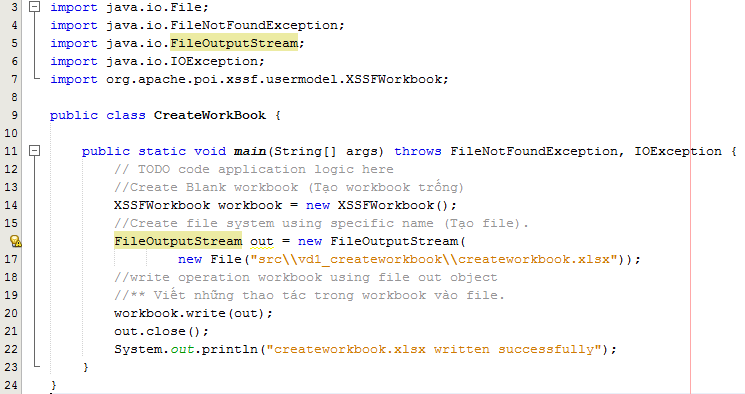






1. **Thao tác tập tin Excel với Apache POI**
2. **Workbook**

Sử dụng workbook để thực hiện một số thao tác như đọc, ghi file MS Excel. Ở đây, workbook có thể hiểu như một tập tin excel.

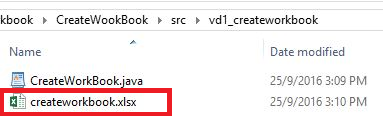
**Create Blank Workbook** (Tạo workbook rỗng)

**Kết quả:**

Cửa sổ console thông báo tạo file thành công.



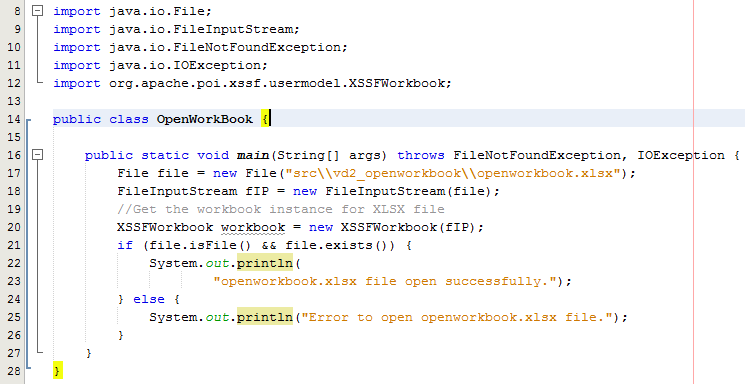
Trong thư mục project xuất hiện file createworkbook.xlsx (mặc dù vẫn tạo được file xlsx nhưng không thể mở bằng Microsoft Office, vì chỉ tạo workbook rỗng, chưa có nội dung ở bên trong).

****

**Open Existing Workbook** (mở một workbook đã tồn tại)

Trước hết cần sử dụng Microsoft Office Excel tạo một file và lưu nó lại (cụ thể ở đây file xlsx có tên là openworkbook.xlsx).

Tiếp theo, sử dụng đoạn mã nguồn sau để mở file .xlsx vừa tạo.



**Kết quả:**

Màn hình console thông báo mở file thành công.



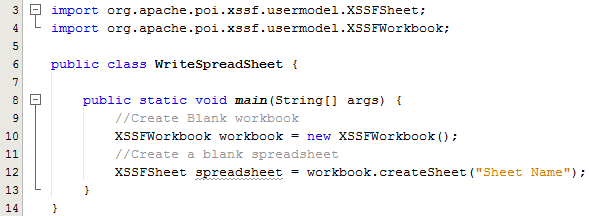
1. **Spreadsheets** (bảng tính)

Trong file excel, chúng ta sẽ thao tác trên các bảng tính – là mộ trang trong file excel, gồm một lưới các ô được sắp xếp theo hàng và cột (rows & columns). Cột được xác định bởi ký tự chữ cái, hàng được xác định bằng ký tự chữ số.

Sau đây là một số ví dụ về thao tác trên một bảng .

* **Create a Spreadsheet** (tạo bảng tính)

Sau khi tạo workbook (file excel), tạo một spreadsheet với tên do người dùng chọn.

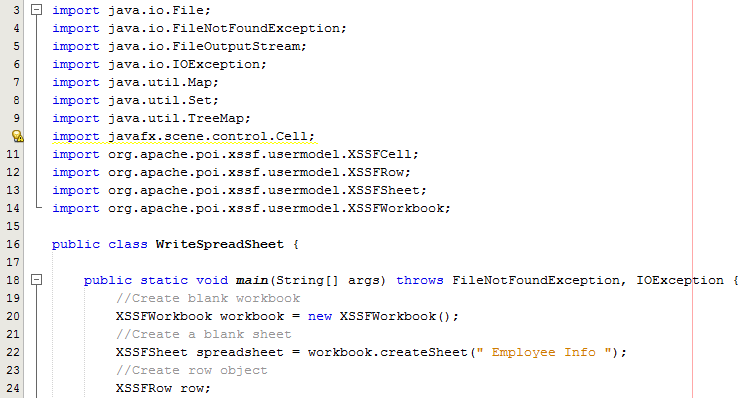


* **Write into a Spreadsheet** (ghi nội dung vào bảng tính)

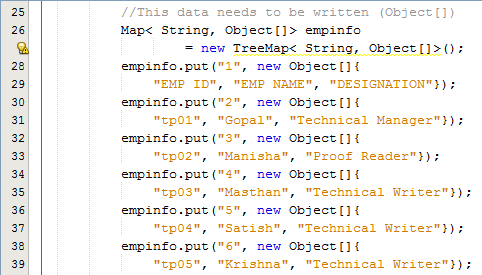
Ví dụ: ghi bảng tính có nội dung như sau vào tập tin xlsx.

|  |  |  |
| --- | --- | --- |
| Emp Id | Emp Name | Designation |
| Tp01 | Gopal | Technical Manager |
| TP02 | Manisha | Proof Reader |
| Tp03 | Masthan | Technical Writer |
| Tp04 | Satish | Technical Writer |
| Tp05 | Krishna | Technical Writer |

**Bước** 1: Tạo workbook, spreadsheet, và row để chứa dữ liệu của hàng.

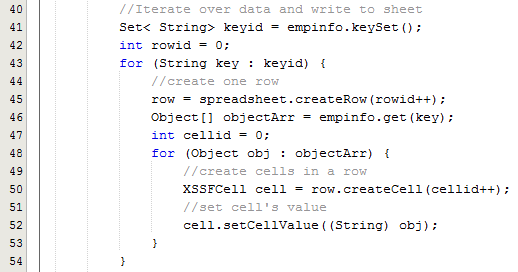


**Bước** 2: Chuẩn bị dữ liệu để ghi vào bảng excel

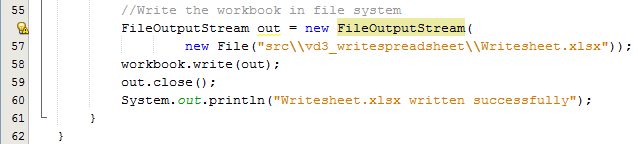


**Bước** 3: Trong mỗi row (hàng), có các cell (sẽ được nói rõ ở phần sau).

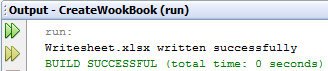
Đưa dữ liệu vào mỗi cell.

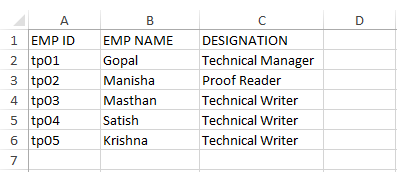


**Bước** 4: Ghi workbook vào file **Writesheet.xlsx**



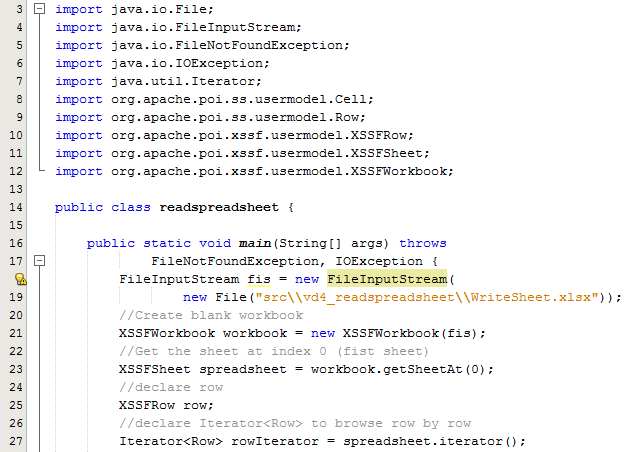
Kết quả hiển thị trên console thông báo ghi vào tập tin thành công.



Mở tập tin **Writesheet.xlsx** và xem kết quả.

* **Read from a Spreadsheet** (Đọc dữ liệu từ một bảng tính)

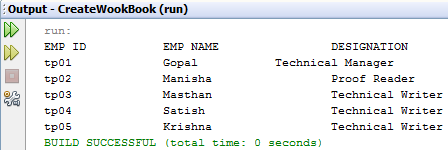
Đọc file WriteSheet.xlsx vừa tạo ở trên.

**Bước** 1: Khai báo workbook, spreadsheet để chứa dữ liệu từ file.

**Bước** 2: Duyệt để lấy dữ liệu vừa spreadsheet.



Kết quả hiển thị trên console như sau:



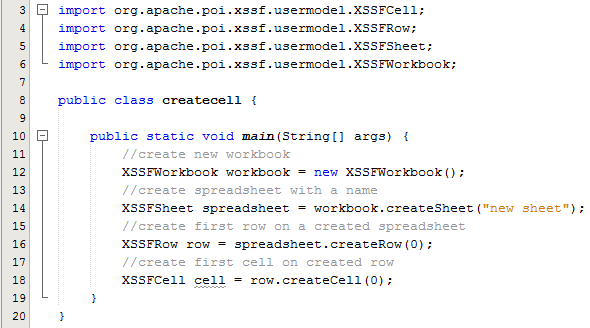
1. **Cells (ô)**

Bất kỳ dữ liệu trong một spreadsheet (bảng tính) thường nằm trong các cell (ô). Tọa độ của cell được xác định bằng tên của cột và thứ tự của hàng.

Sau đây là một vài ví dụ về thao tác trên cell.

* **Create a Cell** (tạo cell)

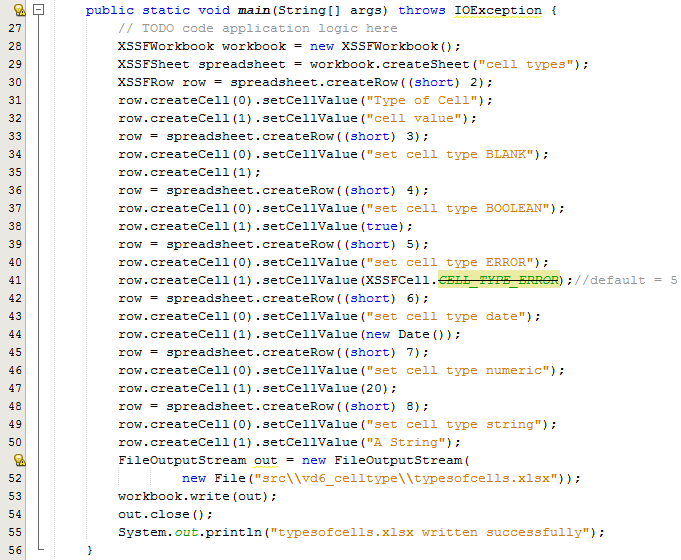
Trước khi tạo một cell, cần phải tạo một row (hàng), vì cell phải thuộc về một row nào đó.



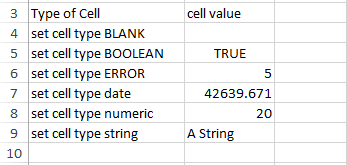
* **Types of Cells (kiểu dữ liệu của cell)**

Khi làm việc với cell, dữ liệu của cell có thể khác nhau tùy vào nhu cầu của người dùng, có thể là chuỗi, số hay biểu thức... Sau đây là danh sách các kiểu giá trị của cell.

|  |  |
| --- | --- |
| Type of cell value | Type Syntax |
| Blank cell value (không có giá trị) | XSSFCell.CELL\_TYPE\_BLANK |
| Boolean cell value | XSSFCell.CELL.TYPE\_BOOLEAN |
| Error cell value | XSSFCell.CELL\_TYPE\_ERROR |
| Numeric cell value | XSSFCell.CELL\_TYPE\_NUMERIC |
| String cell value | XSSFCell.CELL\_TYPE\_STRING |

Ví dụ về các giá trị của cell.

Kết quả ở file **typeofcell.xlsx** như sau:

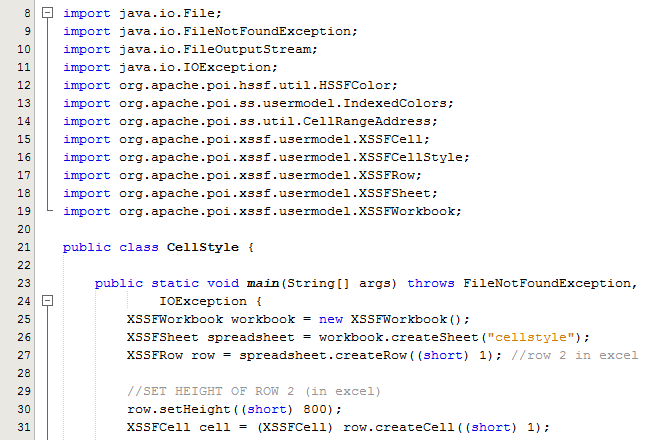


* **Cell Styles (Định dạng phong cách trình bày của cell)**

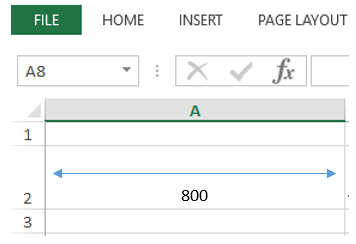
Khi làm việc với một bảng tính, việc định dạng phong cách trình bày của cell cũng được chú trọng, giúp bảng tính được rõ ràng, đẹp mắt hơn. Ví dụ: Gộp các cell lại với nhau, tô viên cho cell, tô màu nền cho cell được nổi bật hơn các cell xung quanh,…

Apache POI cũng hỗ trợ các phương thức để thực hiện việc định dạng phong cách cho cell. Sau đây là ví dụ cho một vài định dạng cell.

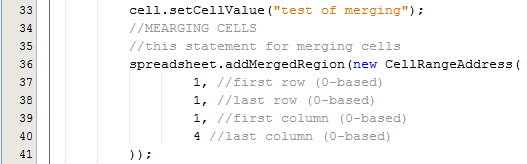
* **Thiết lập độ cao của row** (setting high of row)



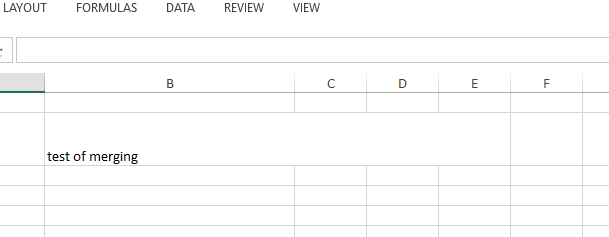
Mở tập tin excel sau khi ghi thành công, có được kết quả của định dạng trên như sau:

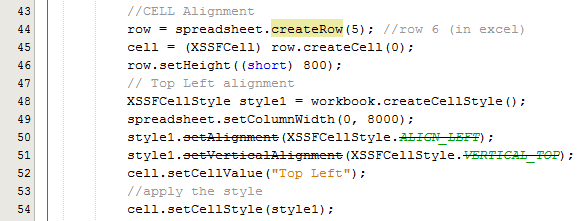


* **Gộp các cell lại với nhau** (merging cells)



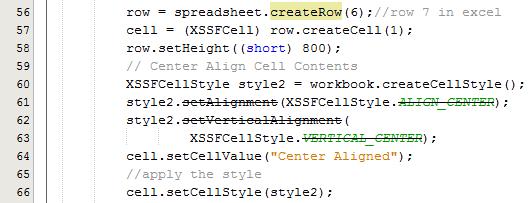
Kết quả:



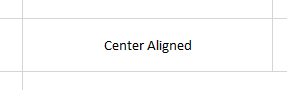
* **Căn lề nội dung trong cell** (cell alignment)

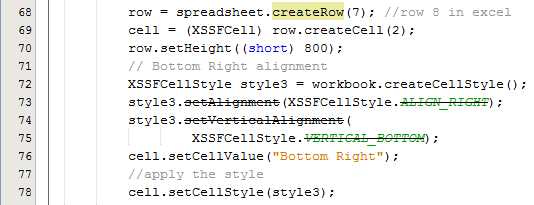
Kết quả:



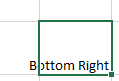


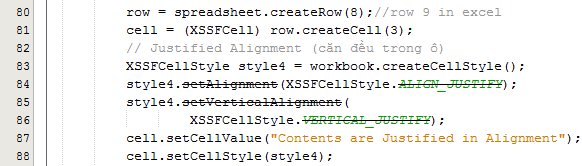
Kết quả:





Kết quả:

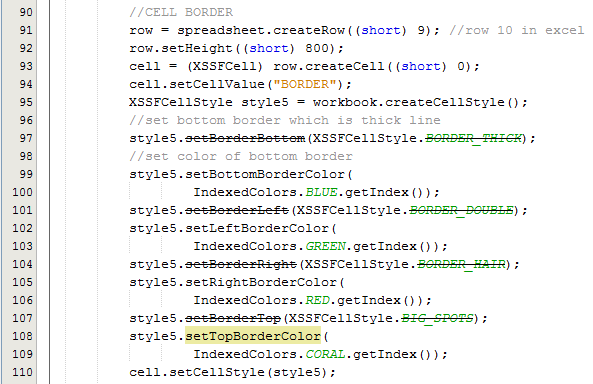




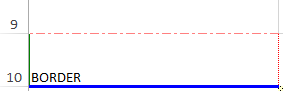
Kết quả:



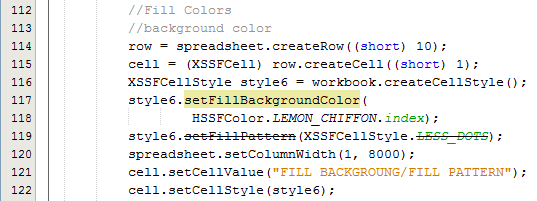
* **Tô viền cho cell** (cell border)



Kết quả:



* **Tô nền cho cell** (cell background)



Kết quả:



Xem thêm các định dạng cho cell ở <https://poi.apache.org/apidocs/org/apache/poi/ss/usermodel/>

1. **Formula** (công thức)

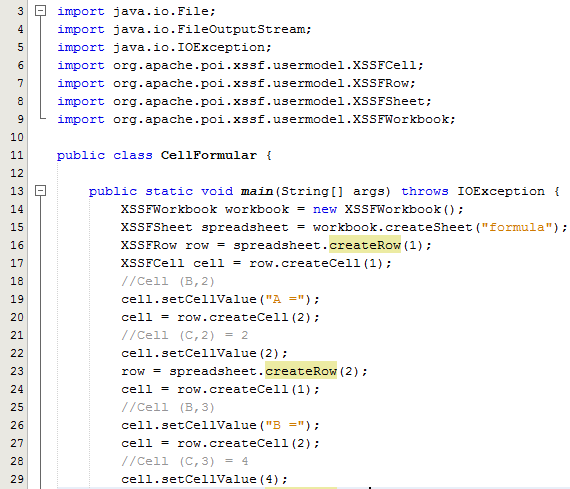
Thông thường, mục đích khi chúng ta sử dụng Excel là để thao tác trên dữ liệu số. Ví dụ tính lương, thống kê số lượng hàng hóa, lọc dữ liệu … bằng cách sử dụng các công thức được hỗ trợ trong Excel.

Apache POI cũng cung cấp các công thức có chức năng tương đồng với các công thức tính toán trên Excel. Interface Cell có phương thức **setCellFormula("formulaString"),** được sử dụng để thêm một biểu thức vào bảng, biểu thức “**formulaString**” là những gì mong muốn khi gõ trong excel (chú ý, không có dấu ‘=’ trước chuỗi biểu thức). Sau đây là một số công thức cơ bản:

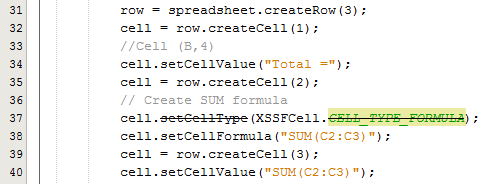
|  |  |
| --- | --- |
| Operation | Syntax |
| Adding multiple numbers | **SUM**(Loc1:Locn) or **SUM**(n1,n2,) |
| Count | **COUNT**(Loc1:Locn) or **COUNT**(n1,n2,) |
| Power of two numbers | **POWER**(Loc1,Loc2) or **POWER**(number, power) |
| Max of multiple numbers | **MAX**(Loc1:Locn) or **MAX**(n1,n2,) |
| Product | **PRODUCT**(Loc1:Locn) or **PRODUCT**(n1,n2,) |
| Factorial | **FACT**(Locn) or **FACT**(number) |
| Absolute number | **ABS**(Locn) or **ABS**(number) |
| Today date | **TODAY**() |
| Converts lowercase | **LOWER**(Locn) or **LOWER**(text) |
| Square root | **SQRT**(locn) or **SQRT**(number) |

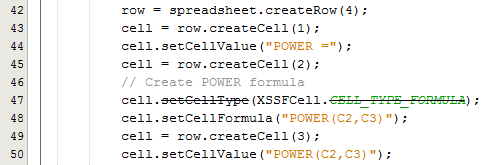
Ví dụ: Thực hiện một số phép tính trên 2 số.

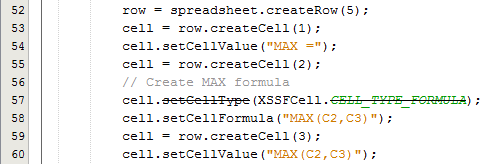
Khai báo 2 cell chứa 2 số có giá trị là 2 và 4.

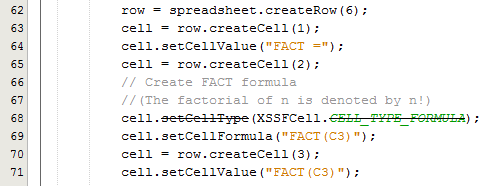


Tính tổng 2 số

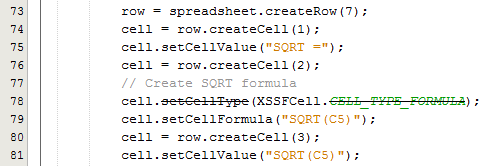


Tính mũ

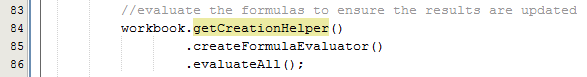
Tính giá trị lớn nhất giữa 2 số

Tính giai thừa

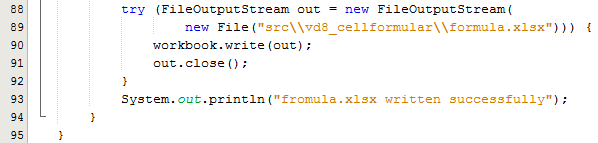
Tính căn bậc 2



Sau khi thực hiện các bước tính toán (bằng biểu thức) trên cells (update cells), cần phải “định giá” ([evaluate](http://www.codejava.net/coding/working-with-formula-cells-in-excel-using-apache-poi)) các biểu thức để đảm bảo rằng kết quả đã được cập nhật.



Ghi nhận vào file formula.xlsx



1. **Kết luận**

Apache POI Project đã cung cấp bộ thư viện thuận tiện cho việc thao tác trên các tập tin Excel bằng Java. Giúp các nhà phát triển có thể sử dụng dữ liệu trực tiếp từ tập tin excel mà không cần phải sử dụng thêm phần mềm bên thứ 3 (MS Office).

Các nổ lực để hoàn thiện các API cho Word và PowerPoint đang được thực hiện và hứa hẹn sẽ mang lại nhiều thuận lợi cho cộng đồng Java.

1. **Tham khảo**

[1] Apache POI from poi.apache.org [[link](http://poi.apache.org/index.html)]

[2] OLE2 FileSystem (POIFS) [[link](http://poi.apache.org/poifs/index.html)]

[3] POI-HSSF and POI-XSSF - Java API To Access Microsoft Excel Format Files [[link](http://poi.apache.org/spreadsheet/index.html)]

[4] High and Low level APIs [[link](https://poi.apache.org/spreadsheet/how-to.html#low_level_api)]

[5] Apache POI from tutorialspoint [[link](https://www.tutorialspoint.com/apache_poi/index.htm)]