Plag

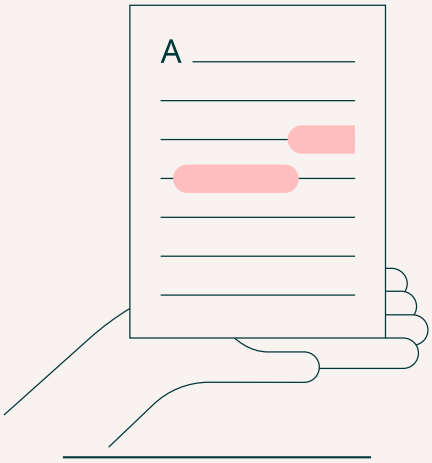
Звіт про оригінальність

● Оцінка схожості 10%

● Ризик плагіату НАЙВИЩИЙ

Olga Kagalo 2025-06-19 22:49

Посилання на звіт: 10mBh / Посилання користувача: qEAc



Ось вона –
Ваша звіт про оригінальність!

Ми раді повідомити, що перевірка вашого документа завершена, і результати вже готові! Наші алгоритми старанно працювали, щоб знайти збіги в наших базах даних.

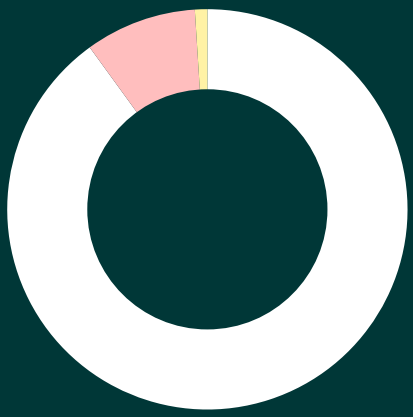
На наступних сторінках ви знайдете результати перевірки:

Бали
Збіги
Посилання

Ваш документ було перевірено за такими джерелами:

- ✓ База даних інтернет-джерел
- ✓ База даних наукових статей
- ○ Глибока перевірка (наш вдосконалений алгоритм)

Бали



Збіги тексту	9%
Перефразування	1%
Цитований текст	0%
Неправильне цитування	0%
Збігів не знайдено	90%

Ризик плагіату НАЙВИЩИЙ

Ризик плагіату вказує, як збіги тексту розподілені по документу. Вищий ризик виникає, коли збіги з'являються близько один до одного, наприклад, у тому самому абзаці або розділі.

Оцінка схожості 10%

Оцінка схожості показує, скільки слів або символів у вашому документі збігаються з текстами інших документів, включаючи перефразовані тексти або неправильні цитати.

Збіги

1 Огляд технології APACHE POI

1.1 Компоненти бібліотеки apache poi

Часто для створення звітів у форматі файлів Microsoft Excel потрібний програмний додаток. Іноді навіть очікується, що додаток одержить файли Excel у якості вхідних даних. Наприклад, додаток, розроблений для фінансового відділу компанії, повинен буде генерувати всі свої результати в Excel.

Будь-який програміст на Java, що бажає створювати файли MS Office у якості вихідних даних, повинен використовувати для цього визначений API тільки для читання.

Apache POI — це популярний API, який дозволяє програмістам створювати, змінювати й відображати файли MS Office за допомогою програм Java. Це бібліотека з відкритим вихідним кодом, розроблена й розповсюджувана Apache Software Foundation для розробки або зміни файлів Microsoft Office з використанням програми Java. Він містить класи й методи для декодування даних, що вводяться користувачем, або файлу в документи MS Office.

Компоненти Apache POI

Apache POI містить класи й методи для роботи з усіма складеними документами OLE2 MS Office. Список компонентів цього API наведений нижче.

POIFS (Файлова система реалізації) — цей компонент є основним чинником усіх інших елементів POI. Він використовується для явного читання різних файлів.

HSSF (формат електронної таблиці) — використовується для читання й запису у форматі xls файлів MS-excel.

XSSF (XML Spreadsheet Format) — використовується для формату файлів xlsx Ms-excel.

HPSF (формат набору властивостей) — використовується для витягу наборів властивостей з файлів MS-office.

HWPF (формат текстового процесора) — використовується для читання й запису файлів розширень doc MS-word.

XWPF (Xml-Формат текстового процесора) — використовується для читання й запису файлів розширення docx в MS-word.

HSLF (формат макета слайда) — використовується для читання, створення й редагування презентацій PowerPoint.

HDGF (формат Horrible Diagram) — містить класи й методи для двійкових файлів MS-visio.

HPBF (формат Horrible Publisher) — використовується для читання й запису файлів MS-publisher .

У проекті ми використовуємо роботу з файлами Excel з використанням Java. Тому обговорення обмежується компонентами HSSF і XSSF.

Більш старі версії POI підтримують двійкові формати файлів, такі як doc, xls, ppt і т. д. Версія 3.5 і вище, POI підтримує формати файлів OOXML MS-office, такі як docx, xlsx, pptx і т. д.

Як і в Apache POI, існують інші бібліотеки, надавані різними постачальниками для генерації файлів Excel. До них відносяться Aspose для Java від Aspose, JXL від Commons Libraries і Jexcel від Team Dev.

1.2 Apache POI — API Java Excel

Розглянемо деяких різновиди Java Excel API і їх функції. Є багато постачальників, які надають API, пов'язані з Java Excel; розглянемо деякі з них:

Aspose Cells для Java

Aspose Cells для Java — це ліцензований API Java Excel, розроблений і розповсюджуваний постачальником Aspose. Остання версія цього API — 8.1.2, випущена в липні 2014 року. Це багатий і складний API (комбінація простих класів Java і класів AWT) для розробки компонента Excel, який може читати, писати й маніпулювати електронними таблицями.

Загальні застосування цього API наступні:

Звіти Excel, створювати динамічні звіти Excel

Високоякісний рендеринг і друк Excel

Імпорт і експорт даних з таблиць Excel

Створювати, редагувати й конвертувати електронні таблиці

JXL

JXL — це сторонній фреймворк, розроблений для Selenium, який підтримує автоматизацію на основі даних у веб-браузерах (автоматичне відновлення даних у веб-браузерах). Однак він також використовується в якості загальної бібліотеки підтримки для API Jexcel, оскільки він має базові функції для створення, читання й запису електронних таблиць.

Основні функції полягають у наступному —

Генерація файлів Excel

Імпорт даних з робочих книг і електронних таблиць

Одержати загальну кількість рядків і стовпців

JXL підтримує тільки формат файлу .xls і не може обробляти більші обсяги даних.

Jexcel

Jexcel — це чисто ліцензований API, надаваний Team Dev. Використовуючи це, програмісти можуть легко читати, писати, відображати й змінювати книги Excel у форматах.xls і .xlsx . Цей API може бути легко вбудований в Java Swing і AWT. Остання версія цього API — Jexcel-2.6.12, випущена в 2009 році.

Основні характеристики полягають у наступному —

Автоматизувати додаток Excel, робочі зошити, електронні таблиці і т. д.

Вбудовувати книги в додаток Java Swing як звичайний компонент Swing

Додає слухачів подій у робочі книги й таблиці

Додає оброблювачі подій, щоб обробляти поведінка робочої книги й електронних таблиць

Apache POI — це бібліотека з відкритим вихідним кодом, надана Apache Software Foundation. Більшість розроблювачів малих і середніх додатків сильно залежать від Apache POI (HSSF + XSSF). Він підтримує всі основні функції бібліотек Excel (рис. 1.1); однак рендеринг і витяг тексту є його основними характеристиками.

Рисунок 1.1 Функції Excel, які підтримує бібліотека POI

Розгянемо настроювання Apache POI у системах на базі Windows і Linux. Apache POI можна легко встановити й інтегрувати в поточне середовище Java, виконавши кілька простих кроків без яких-небудь складних процедур настроювання. Користувачу потрібні права адміністратора при установці.

Опишемо етапи установки Apache POI.

Крок 1. Перевірте вашу установку Java

Насамперед, вам необхідно встановити Java Software Development Kit (SDK) у вашій системі. Щоб переконатися в цьому, виконаєте кожну із двох команд залежно від платформи, на якій ви працюєте.

Якщо установка Java була виконана правильно, то на ній відобразиться поточна версія й специфікація вашої установки Java.

Якщо у вас немає Java SDK, завантажите його поточну версію з https://www.oracle.com/technetwork/java/javase/downloads/index.html і встановите його.

Крок 2. Встановіть середовище Java

Установите змінне середовища JAVA_HOME, щоб вона вказувала на місце розташування базової директорії, де встановлена java на вашім комп'ютері. Наприклад,

Windows

Установите JAVA_HOME в C: \ Programfiles \ java \ jdk1.7.0_60

Linux

Експорт JAVA_HOME = / usr / local / java-current

Додайте повний шлях розташування компілятора Java до системного шляху.

Платформа й опис

Windows

Додайте рядок «C: \ Program Files \ Java \ jdk1.7.0_60 \ bin» у кінець системної змінної PATH.

Linux

Експорт PATH = $ PATH: $ JAVA_HOME / bin /

Windows

Додайте рядок «C: \ Program Files \ Java \ jdk1.7.0_60 \ bin» у кінець системної змінної PATH.

Linux

Експорт PATH = $ PATH: $ JAVA_HOME / bin /

Виконаєте команду java -version з командного рядка, як описано вище.

Крок 3: Встановіть Apache POI Library

Завантажите останню версію Apache POI за адресу https://poi.apache.org/download.html і разархивируйте його вміст у папку, з якої необхідні бібліотеки можуть бути пов'язані з вашою програмою Java. Припустимо, що файли зібрані в папці на диску C.

На наступних зображеннях показані каталоги й файлова структура усередині завантаженої папки (рис. 1.2.).

Рисунок 1.2 – Каталоги й файлова структура папки POI

Додайте повний шлях до п'ятьох jar, як показано на зображенні вище, до CLASSPATH.

Платформа й опис

Windows

Додайте наступні рядки в кінець користувацької змінної

CLASSPATH —

«C: \ poi-3,9 \ poi -3.9-20121203.jar;»

«C: \ poi -3,9 \ poi -OOXML-3.9-20121203.jar;»

«C: \ poi -3,9 \ ooxml-схеми-3.9-20121203.jar;»

«C: \ poi -3,9 \ OOXML Пб \ dom4j-1.6.1.jar;»

«C:. \ poi -3,9 \ OOXML Пб \ Xmlbeans-2.3.0.jar;»;

Linux

Експортувати CLASSPATH = $ CLASSPATH:

/usr/share/poi-3.9/poi-3.9-20121203.tar:

/usr/share/poi-3.9/poi-ooxml-schemas-3.9-20121203.tar:

/usr/share/poi-3.9/poi-ooxml-3.9-20121203.tar:

/usr/share/poi-3.9/ooxml-lib/dom4j-1.6.1.tar:

/usr/share/poi-3.9/ooxml-lib/xmlbeans-2.3.0.tar

2 Основні класи Apache POI

Опишемо кілька класів і методів в Api-Інтерфейсі Apache POI, які мають вирішальне значення для роботи з файлами Excel за допомогою програм на Java.

Робоча книга

Це супер-інтерфейс усіх класів, які створюють або підтримують книги Excel. Він належить пакету org.apache.poi.ss.usermodel. Два класи, які реалізують цей інтерфейс, що випливають:

Hssfworkbook — у цьому класі є методи для читання й запису файлів Microsoft Excel у форматі .xls. Він сумісний з версіями Ms-office 97–2003.

Xssfworkbook — у цьому класі є методи для читання й запису Xml-Файлів Microsoft Excel і Openoffice у форматі .xls або .xlsx. Це сумісне з версіями Ms-office 2007 або пізніше.

Hssfworkbook — у цьому класі є методи для читання й запису файлів Microsoft Excel у форматі .xls. Він сумісний з версіями Ms-office 97–2003.

Xssfworkbook — у цьому класі є методи для читання й запису Xml-Файлів Microsoft Excel і Openoffice у форматі .xls або .xlsx. Це сумісне з версіями Ms-office 2007 або пізніше.

Hssfworkbook

Це клас високого рівня в пакеті org.apache.poi.hssf.usermodel . Він реалізує інтерфейс Workbook і використовується для файлів Excel у форматі .xls. Нижче перераховані деякі методи й конструктори цього класу.

Конструктори класів

Конструктор і опис

Hssfworkbook ()

Створює новий об'єкт Hssfworkbook з нуля.

Hssfworkbook (каталог Directorynode, логічні preservenodes)

Створює новий об'єкт Hssfwworkbook усередині певного каталогу.

Hssfworkbook (каталог Directorynode, Poifsfilesystem fs, логічні preservenodes)

Враховуючи об'єкт Poifsfilesystem і певний каталог у ньому, він створює об'єкт Ssfworkbook для читання зазначеної книги.

Hssfworkbook (java.io.Inputstream s)

Створює новий об'єкт Hssfworkbook, використовуючи вхідний потік.

Hssfworkbook (java.io.Inputstream s, логічні preservenodes)

Створює файлову систему POI навколо вашого вхідного потоку.

Hssfworkbook (Poifsfilesystem fs)

Створює новий об'єкт Hssfworkbook, використовуючи об'єкт Poifsfilesystem.

Hssfworkbook (Poifsfilesystem fs, логічні preservenodes)

Враховуючи об'єкт Poifsfilesystem, він створює новий об'єкт Hssfworkbook для читання зазначеної книги.

Hssfworkbook ()

Створює новий об'єкт Hssfworkbook з нуля.

Hssfworkbook (каталог Directorynode, логічні preservenodes)

Створює новий об'єкт Hssfwworkbook усередині певного каталогу.

Hssfworkbook (каталог Directorynode, Poifsfilesystem fs, логічні preservenodes)

Враховуючи об'єкт Poifsfilesystem і певний каталог у ньому, він створює об'єкт Ssfworkbook для читання зазначеної книги.

Hssfworkbook (java.io.Inputstream s)

Створює новий об'єкт Hssfworkbook, використовуючи вхідний потік.

Hssfworkbook (java.io.Inputstream s, логічні preservenodes)

Створює файлову систему POI навколо вашого вхідного потоку.

Hssfworkbook (Poifsfilesystem fs)

Створює новий об'єкт Hssfworkbook, використовуючи об'єкт Poifsfilesystem.

Hssfworkbook (Poifsfilesystem fs, логічні preservenodes)

Враховуючи об'єкт Poifsfilesystem, він створює новий об'єкт Hssfworkbook для читання зазначеної книги.

Часто використовувані параметри усередині цих конструкторів:

каталог — це каталог файлової системи POI для обробки.

fs — це файлова система POI, яка містить потік робочої книги.

preservenodes — це необов'язковий параметр, який вирішує, чи зберігати інші вузли, такі як макроси. Він споживає багато пам'яті, тому що зберігає всю систему Poifilesystem у пам'яті (якщо встановлена).

каталог — це каталог файлової системи POI для обробки.

fs — це файлова система POI, яка містить потік робочої книги.

preservenodes — це необов'язковий параметр, який вирішує, чи зберігати інші вузли, такі як макроси. Він споживає багато пам'яті, тому що зберігає всю систему Poifilesystem у пам'яті (якщо встановлена).

Примітка. Клас Hssfworkbook містить кілька методів; однак вони сумісні тільки з форматом xls. У цьому керівництві основна увага приділяється останньої версії форматів файлів Excel. Отже, методи класу Hssfworkbook тут не перераховані. Якщо вам потрібні ці методи класу, звернетеся до API класу Poi-hssfworkbook за адресою https://poi.apache.org/apidocs/org/apache/poi/hssf/usermodel/HSSFWorkbook.html.

Xssfworkbook

Це клас, який використовується для вистави форматів файлів Excel як високого, так і низького рівня. Він належить пакету org.apache.xssf.usemodel і реалізує інтерфейс Workbook . Нижче перераховані методи й конструктори цього класу.

Xssfworkbook ()

Створює новий об'єкт Xssfworkbook з нуля.

Xssfworkbook (файл java.io.File)

Створює об'єкт Xssfworkbook із заданого файлу.

Xssfworkbook (java.io.Inputstream is)

Створює об'єкт Xssfworkbook, буферизуя весь вхідний потік на згадку й потім відкриваючи для нього об'єкт Opcpackage.

Xssfworkbook (шлях java.lang.String)

Створює об'єкт Xssfworkbook з повним шляхом до файлу.

Xssfworkbook ()

Створює новий об'єкт Xssfworkbook з нуля.

Xssfworkbook (файл java.io.File)

Створює об'єкт Xssfworkbook із заданого файлу.

Xssfworkbook (java.io.Inputstream is)

Створює об'єкт Xssfworkbook, буферизуя весь вхідний потік на згадку й потім відкриваючи для нього об'єкт Opcpackage.

Xssfworkbook (шлях java.lang.String)

Створює об'єкт Xssfworkbook з повним шляхом до файлу.

Методи класу

Метод і опис

createsheet ()

Створює аркуш XSSFS для цієї книги, додає його на аркуші й повертає виставу високого рівня.

createsheet (ім'я аркуша java.lang.String)

Створює новий аркуш для цієї робочої книги й повертає виставу високого рівня.

Createfont ()

Створює новий шрифт і додає його в таблицю шрифтів робочої книги.

createcellstyle ()

Створює новий Xssfcellstyle і додає його в таблицю стилів робочої книги.

Createfont ()

Створює новий шрифт і додає його в таблицю шрифтів робочої книги.

2 setprintarea (int sheetindex, int startcolumn, int endcolumn, int startrow, int endrow)

2 Встановлює область друку даного аркуша відповідно до зазначених параметрів.

createsheet ()

Створює аркуш XSSFS для цієї книги, додає його на аркуші й повертає виставу високого рівня.

createsheet (ім'я аркуша java.lang.String)

Створює новий аркуш для цієї робочої книги й повертає виставу високого рівня.

Createfont ()

Створює новий шрифт і додає його в таблицю шрифтів робочої книги.

createcellstyle ()

Створює новий Xssfcellstyle і додає його в таблицю стилів робочої книги.

Createfont ()

Створює новий шрифт і додає його в таблицю шрифтів робочої книги.

2 setprintarea (int sheetindex, int startcolumn, int endcolumn, int startrow, int endrow)

2 Установлює область печатки даного аркуша відповідно до зазначених параметрів.

Інші методи цього класу див. У повному документі API за адресою: https://poi.apache.org/apidocs/org/apache/poi/xssf/usermodel/XSSFWorkbook.html. для повного списку методів.

Листи

Sheet — це інтерфейс у пакеті org.apache.poi.ss.usermodel, і це суперінтерфейс усіх класів, які створюють електронні таблиці високого або низького рівня з конкретними іменами. Найпоширенішим типом електронної таблиці є робочий аркуш, який представлений у вигляді сітки гнізд.

Hssfsheet

Це клас у пакеті org.apache.poi.hssf.usermodel . Він може створювати таблиці Excel і дозволяє форматувати стиль аркуша й дані аркуша.

Конструктори класів

Конструктор і опис

Hssfsheet (робоча книга Hssfworkbook)

Створює новий аркуш HSSFS, викликуваний Hssfworkbook для створення аркуша з нуля.

Hssfsheet (робоча книга Hssfworkbook, аркуш Internalsheet)

Створює аркуш HSSFS, що представляє даний об'єкт аркуша.

Hssfsheet (робоча книга Hssfworkbook)

Створює новий аркуш HSSFS, викликуваний Hssfworkbook для створення аркуша з нуля.

Hssfsheet (робоча книга Hssfworkbook, аркуш Internalsheet)

Створює аркуш HSSFS, що представляє даний об'єкт аркуша.

Xssfsheet

Це клас, який представляє высокоуровневое вистава таблиці Excel. Він перебуває в пакеті org.apache.poi.hssf.usermodel .

Конструктори класів

Конструктор і опис

Xssfsheet ()

Створює новий аркуш XSSFS — викликається Xssfworkbook для створення аркуша з нуля.

Xssfsheet (частина Packagepart, rel)

Створює Xssfsheet, що представляє дану частину пакета й відносини.

Xssfsheet ()

Створює новий аркуш XSSFS — викликається Xssfworkbook для створення аркуша з

нуля.

Xssfsheet (частина Packagepart, rel)

Створює Xssfsheet, що представляє дану частину пакета й відносини.

Методи класу

Метод і опис

addmergedregion (Cellrangeaddress region)

Додає об'єднану область гнізд (отже, ці гнізда утворюють одиницю).

autosizecolumn (стовпець int)

Регулює ширину стовпця відповідно до вмісту.

итератора ()

Цей метод є псевдонімом для rowiterator (), щоб дозволити цикли foreach.

addhyperlink (гіперпосилання Xssfhyperlink)

Реєструє гіперпосилання в колекції гіперпосилань на цьому аркуші

addmergedregion (Cellrangeaddress region)

Додає об'єднану область гнізд (отже, ці гнізда утворюють одиницю).

autosizecolumn (стовпець int)

Регулює ширину стовпця відповідно до вмісту.

итератора ()

Цей метод є псевдонімом для rowiterator (), щоб дозволити цикли foreach.

addhyperlink (гіперпосилання Xssfhyperlink)

Реєструє гіперпосилання в колекції гіперпосилань на цьому аркуші

Для інших методів цього класу, звернетеся до повного API за адресою: https://poi.apache.org/apidocs/org/apache/poi/xssf/usermodel/XSSFSheet.html.

Рядок

Це інтерфейс у пакеті org.apache.poi.ss.usermodel . Він використовується для высокоуровневого вистави рядка електронної таблиці. Це суперінтерфейс усіх класів, які представляють рядки в бібліотеці POI.

Xssfrow

Це клас у пакеті org.apache.poi.xssf.usermodel . Він реалізує інтерфейс рядків, тому він може створювати рядка в електронній таблиці. Нижче перераховані методи й конструктори цього класу.

Методи класу

Метод і опис

createcell (int columnindex)

Створює нові гнізда в рядку й повертає її.

setheight (коротка висота)

Установлює висоту в коротких одиницях.

createcell (int columnindex)

Створює нові гнізда в рядку й повертає її.

setheight (коротка висота)

Установлює висоту в коротких одиницях.

Для інших методів цього класу перейдіть по зазначенім посиланню https://poi.apache.org/apidocs/org/apache/poi/xssf/usermodel/XSSFRow.html.

Комірка

Це інтерфейс у пакеті org.apache.poi.ss.usermodel . Це суперінтерфейс усіх класів, які представляють комірки в рядках електронної таблиці.

Комірки можуть мати різні атрибути, такі як порожнє, числове, дата, помилка і т. д. Комірки повинні мати свої власні номери (на основі 0) перед додаванням у рядок.

Xssfcell

Це клас у пакеті org.apache.poi.xssf.usermodel . Він реалізує інтерфейс Cell. Це высокоуровневое вистава гнізд у рядках електронної таблиці.

Зведення по полю

Нижче перераховані деякі поля класу Xssfcell разом з їхнім описом.

Тип комірки й опис

CELL_TYPE_BLANK

Представляє порожню комірку

CELL_TYPE_BOOLEAN

Представляє логічне гніздо (true або false)

CELL_TYPE_ERROR

Представляє значення помилки в комірці

CELL_TYPE_FORMULA

Представляє результат формули в комірці

CELL_TYPE_NUMERIC

Представляє числові дані в комірці

CELL_TYPE_STRING

Представляє рядок (текст) у комірку

Методи класу

Метод і опис

setcellstyle (стиль Cellstyle)

Установлює стиль для комірки.

setcelltype (int celltype)

Установлює тип гнізд (числовий, формули або рядки).

setcellvalue (логічне значення)

Встановлює логічне значення для комірки.

setcellvalue (значення java.util.Calendar)

Установлює значення дати для комірки.

setcellvalue (подвійне значення)

Установлює числове значення для комірки.

setcellvalue (java.lang.String str)

Установлює строкове значення для комірки.

sethyperlink (гіперпосилання гіперпосилання)

Призначає гіперпосилання на цю комірку.

setcellstyle (стиль Cellstyle)

Установлює стиль для комірки.

setcelltype (int celltype)

Встановлює тип комірок (числовий, формули або рядки).

setcellvalue (логічне значення)

Встановлює логічне значення для комірки.

setcellvalue (значення java.util.Calendar)

Установлює значення дати для комірки.

setcellvalue (подвійне значення)

Установлює числове значення для комірки.

setcellvalue (java.lang.String str)

Встановлює строкове значення для комірки.

sethyperlink (гіперпосилання гіперпосилання)

Призначає гіперпосилання на цю комірку.

Для інших методів і полів цього класу перейдіть по наступнім посиланню: https://poi.apache.org/apidocs/org/apache/poi/xssf/usermodel/XSSFCell.html.

Xssfcellstyle

Це клас у пакеті org.apache.poi.xssf.usermodel . Він надасть можливу інформацію про формат вмісту в комірки електронної таблиці. Він також надає варіанти для зміни цього формату. Він реалізує інтерфейс Cellstyle.

Зведення по полю

У наступній таблиці перераховано кілька полів, які успадковані від інтерфейсу Cellstyle.

Поле й опис

ALIGN_CENTER

Центр вирівняти вміст комірки

ALIGN_CENTER_SELECTION

Горизонтальне вирівнювання по центру

ALIGN_FILL

Комірка відповідає розміру контенту

ALIGN_JUSTIFY

Підігнати вміст комірки по ширині

ВИРІВНЯТИ ПО ЛІВОМУ КРАЮ

Вирівняти по лівому краю вміст комірки

ALIGN_RIGHT

Вирівняйте вміст комірки вправо

BORDER_DASH_DOT

Стиль комірки з тире й крапкою

BORDER_DOTTED

Стиль комірки з пунктирною рамкою

BORDER_DASHED

Стиль комірки з пунктирною обрамівкою

BORDER_THICK

Стиль комірки з товстою обрамівкою

BORDER_THIN

Стиль гнізда з тонкою облямівкою

VERTICAL_BOTTOM

Вирівняйте вміст комірки по вертикалі

VERTICAL_CENTER

Вирівняйте вміст комірки по центру

VERTICAL_JUSTIFY

Вирівняйте й вирівняйте вміст комірки по вертикалі

VERTICAL_TOP

Вертикальне вирівнювання зверху

ALIGN_CENTER

Центр вирівняти вміст комірки

ALIGN_CENTER_SELECTION

Горизонтальне вирівнювання по центру

ALIGN_FILL

комірки відповідає розміру контенту

ALIGN_JUSTIFY

Підігнати вміст комірки по ширині

ВИРІВНЯТИ ПО ЛІВОМУ КРАЮ

Вирівняти по лівому краю вміст комірки

ALIGN_RIGHT

Вирівняйте вміст комірки вправо

BORDER_DASH_DOT

Стиль комірки з тире й крапкою

BORDER_DOTTED

Стиль комірки з пунктирною рамкою

BORDER_DASHED

Стиль комірки з пунктирною обрамівкою

BORDER_THICK

Стиль комірки з товстою обрамівкою

BORDER_THIN

Стиль комірки з тонкою обрамівкою

VERTICAL_BOTTOM

Вирівняйте вміст комірки по вертикалі

VERTICAL_CENTER

Вирівняйте вміст комірки по центру

VERTICAL_JUSTIFY

Вирівняйте й вирівняйте вміст комірки по вертикалі

VERTICAL_TOP

Вертикальне вирівнювання зверху

Конструктори класів

Конструктор і опис

З Xssfcellstyle (int cellxfid, int cellstylexfid, Stylestable stylessource, тема Themestable)

Створює стиль комірки із частин, що поставляються

З Xssfcellstyle (Stylestable stylessource)

З Створює порожню комірку Стиль

Методи класу

Метод і опис

setalignment (коротке вирівнювання)

Установлює тип горизонтального вирівнювання для комірки

setborderbottom (коротка границя)

Установлює тип границі для нижньої границі комірки

setbordercolor (сторона Xssfcellborder.Borderside, колір Xssfcolor)

Установлює колір для обраної границі

setborderleft (Коротка границя)

Установлює тип границі для лівої границі комірки

setborderright (коротка границя)

Установлює тип границі для правої границі комірки

setbordertop (коротка границя)

Установлює тип границі для верхньої границі гнізда

setfillbackgroundcolor (колір Xssfcolor)

Установлює колір заливання тла, представлений у вигляді значення Xssfcolor.

setfillforegroundcolor (колір Xssfcolor)

Установлює колір заливання переднього плану, представлений у вигляді значення Xssfcolor.

setfillpattern (короткий фп)

Визначає інформацію про заливання гнізд для заливання гнізд по шаблонові й суцільному кольору.

setfont (шрифт шрифту)

Установлює шрифт для цього стилю.

setrotation (коротке обертання)

Установлює ступінь повороту для тексту в комірки.

setverticalalignment (коротке вирівнювання)

Установлює тип вертикального вирівнювання для комірки.

setalignment (коротке вирівнювання)

Установлює тип горизонтального вирівнювання для комірки

setborderbottom (коротка границя)

Установлює тип границі для нижньої границі комірки

setbordercolor (сторона Xssfcellborder.Borderside, колір Xssfcolor)

Установлює колір для обраної границі

setborderleft (Коротка границя)

Установлює тип границі для лівої границі комірки

setborderright (коротка границя)

Установлює тип границі для правої границі комірки

setbordertop (коротка границя)

Установлює тип границі для верхньої границі комірки

setfillbackgroundcolor (колір Xssfcolor)

Установлює колір заливання тла, представлений у вигляді значення Xssfcolor.

setfillforegroundcolor (колір Xssfcolor)

Установлює колір заливання переднього плану, представлений у вигляді значення Xssfcolor.

setfillpattern (короткий фп)

Визначає інформацію про заливання гнізд для заливання комірки по шаблонові й суцільному кольору.

setfont (шрифт шрифту)

Установлює шрифт для цього стилю.

setrotation (коротке обертання)

Установлює ступінь повороту для тексту в комірки.

setverticalalignment (коротке вирівнювання)

Установлює тип вертикального вирівнювання для комірки.

Для інших методів і полів у цьому класі перейдіть по наступнім посиланню: https://poi.apache.org/apidocs/org/apache/poi/xssf/usermodel/XSSFCellStyle.html.

Hssfcolor

Це клас у пакеті org.apache.poi.hssf.util . Він надає різні кольори як вкладені класи. Звичайно ці вкладені класи представлені з використанням своїх власних індексів. Він реалізує інтерфейс Color.

Вкладені класи

Усі вкладені класи цього класу є статичними, і в кожного класу є свій індекс. Ці вкладені колірні класи використовуються для форматування комірок, такого як уміст комірок, границі, передній план і тло. Нижче перераховані деякі із вкладених класів.

Методи класу

Важливий тільки один метод цього класу, який використовується для одержання значення індексу.

Getindex ()

Цей метод використовується для одержання значення індексу вкладеного класу.

Інші методи й вкладені класи див. По наступнім посиланню: https://poi.apache.org/apidocs/org/apache/poi/hssf/util/HSSFColor.html.

Xssfcolor

Це клас у пакеті org.apache.poi.xssf.usermodel . Він використовується для встановлення кольору в електронній таблиці. Він реалізує інтерфейс Color. Нижче перераховані деякі з його методів і конструкторів.

Xssfcolor ()

Створює новий екземпляр Xssfcolor.

Xssfcolor (byte [] rgb)

Створює новий екземпляр Xssfcolor, використовуючи RGB.

Xssfcolor (java.awt.Color clr)

Створює новий екземпляр Xssfcolor, використовуючи клас Color з пакета awt.

Xssfcolor ()

Створює новий екземпляр Xssfcolor.

Xssfcolor (byte [] rgb)

Створює новий екземпляр Xssfcolor, використовуючи RGB.

Xssfcolor (java.awt.Color clr)

Створює новий екземпляр Xssfcolor, використовуючи клас Color з пакета awt.

Методи класу

setauto (логічне авто)

Установлює логічне значення, що вказує, що ctcolor є автоматичним і системний ctcolor є залежним.

setindexed (int indexed)

Установлює індексоване значення ctcolor як системний ctcolor.

Для інших методів перейдіть по наступнім

посиланню: https://poi.apache.org/apidocs/org/apache/poi/xssf/usermodel/XSSFColor.html.

Xssffont

Це клас у пакеті org.apache.poi.xssf.usermodel . Він реалізує інтерфейс Font і тому може обробляти різні шрифти в книзі.

Конструктор класів

Xssffont ()

Створює новий екземпляр Xssfont.

Методи класу

Метод і опис

setbold (логічне напівжирне)

Установлює логічне значення для атрибута «напівжирний».

setcolor (короткий колір)

Установлює індексований колір для шрифту.

setcolor (Xssfcolor color)

Установлює колір для шрифту в стандартному альфа RGB колірнім значенні.

setfontheight (коротка висота)

Установлює висоту шрифту в пунктах.

setfontname (ім'я java.lang.String)

Установлює ім'я для шрифту.

setitalic (логічний курсив)

Установлює логічне значення для властивості ‘italic’.

setbold (логічне напівжирне)

Установлює логічне значення для атрибута «напівжирний».

setcolor (короткий колір)

Установлює індексований колір для шрифту.

setcolor (Xssfcolor color)

Установлює колір для шрифту в стандартному альфа RGB колірнім значенні.

setfontheight (коротка висота)

Установлює висоту шрифту в пунктах.

setfontname (ім'я java.lang.String)

Встановлює ім'я для шрифту.

setitalic (логічний курсив)

Встановлює логічне значення для властивості ‘italic’.

Для інших методів перейдіть по наступнім посиланню: https://poi.apache.org/apidocs/org/apache/poi/xssf/usermodel/XSSFFont.html.

Xssfhyperlink

Це клас у пакеті org.apache.poi.xssf.usermodel . Він реалізує інтерфейс Hyperlink. Він використовується для установки гіперпосилання на вміст гнізда електронної таблиці.

поля

Поля цього класу наступні. Тут поля означають типи використовуваних гіперпосилань.

LINK_DOCUMENT

Використовується для посилання на будь-який інший документ

LINK_EMAIL

Використовується для посилання на електронну пошту

LINK_FILE

Використовується для зв'язку будь-якого іншого файлу в будь-якому форматі

LINK_URL

Використовується для посилання на веб-URL

Методи класу

setaddress (адреса java.lang.String)

Адреса гіперпосилання.

Для інших методів перейдіть по наступнім посиланню: https://poi.apache.org/apidocs/org/apache/poi/xssf/usermodel/XSSFHyperlink.html.

Це клас у пакеті org.apache.poi.xssf.usermodel . Він реалізує інтерфейс Creationhelper. Він використовується як клас підтримки для оцінки формул і настроювання гіперпосилань.

3 Розробка програм для формування звітів у MS Excel

3.1 Розробка програми Weekly Timesheet

Створюємо програму, що створює щотижневий табель обліку робочого часу, з

автоматичним розрахунком загальної кількості годин (рис. 3.1).

Рисунок 3.1 – Вигляд табелю обліку робочого часу

Наведемо текст програми на мові Java.

```
1 import java.io.FileOutputStream;
1 import java.util.HashMap;
1 import java.util.Map;
1 import org.apache.poi.hssf.usermodel.HSSFWorkbook;
1 import org.apache.poi.ss.usermodel.BorderStyle;
1 import org.apache.poi.ss.usermodel.Cell;
1 import org.apache.poi.ss.usermodel.CellStyle;
1 import org.apache.poi.ss.usermodel.FillPatternType;
1 import org.apache.poi.ss.usermodel.Font;
1 import org.apache.poi.ss.usermodel.HorizontalAlignment;
1 import org.apache.poi.ss.usermodel.IndexedColors;
1 import org.apache.poi.ss.usermodel.PrintSetup;
1 private static final String[] titles = 1 {
1 "Person", "ID", "Mon", "Tue", "Wed", "Thu", "Fri", "Sat", "Sun",
1 "Total\nHrs", "Overtime\nHrs", "Regular\nHrs"
1 };
1 private static final Object[][] sample_data = 1 {
1 {"Yegor Kozlov", "YK", 5.0, 8.0, 10.0, 5.0, 5.0, 7.0, 6.0},
1 {"Gisella Bronzetti", "GB", 4.0, 3.0, 1.0, 3.5, null, null, 4.0},
1 };
private TimesheetDemo() {}
```

```
public static void main(String[] args) throws Exception {
Workbook wb;
if(args.length > 0 && args[0].equals("-xls")) wb = new HSSFWorkbook();
else wb = new XSSFWorkbook();
Map<String, CellStyle> styles = createStyles(wb);
Sheet sheet = wb.createSheet("Timesheet");
PrintSetup printSetup = sheet.getPrintSetup();
printSetup.setLandscape(true);
sheet.setFitToPage(true);
sheet.setHorizontallyCenter(true);
//title row
Row titleRow = sheet.createRow(0);
titleRow.setHeightInPoints(45);
Cell titleCell = titleRow.createCell(0);
titleCell.setCellValue("Weekly Timesheet");
titleCell.setCellStyle(styles.get("title"));
sheet.addMergedRegion(CellRangeAddress.valueOf("$A$1:$L$1"));
//header row
Row headerRow = sheet.createRow(1);
headerRow.setHeightInPoints(40);
Cell headerCell;
for (int i = 0; i < titles.length; i++) {
headerCell = headerRow.createCell(i);
headerCell.setCellValue(titles[i]);
```

```
1 headerCell.setCellStyle(styles.get("header"));

1 }

1 int rownum = 1 2;

1 for (int i = 0; i < 10; i++) {

Row row = 1 sheet.createRow(rownum++);

1 for (int j = 0; j < titles.length; j++) {

Cell cell = row.createCell(j);

if(j == 1 9){

1 //the 10th cell contains sum over week days, e.g. SUM(C3:I3)

1 String ref = "C" +rownum+ ":I" + rownum;

1 cell.setCellFormula("SUM("+ref+")");

1 cell.setCellStyle(styles.get("formula"));

1 } else if (j == 11){

1 cell.setCellFormula("J" +rownum+ "-K" + rownum);

cell.setCellStyle(styles.get("formula"));

} else {

cell.setCellStyle(styles.get("cell"));

}

}

}

1 //row with totals below

1 Row sumRow = sheet.createRow(rownum++);

1 sumRow.setHeightInPoints(35);

1 Cell cell;
```

```
1 cell = sumRow.createCell(0);
1 cell.setCellStyle(styles.get("formula"));
1 cell = sumRow.createCell(1);
1 cell.setCellValue("Total Hrs:");
1 cell.setCellStyle(styles.get("formula"));
1 for (int j = 2; j < 12; j++) {
cell = sumRow.createCell(j);
String ref = (char)('A' + j) + "3:" + (char)('A' + j) + "12";
cell.setCellFormula("SUM(" + ref + ")");
if(j >= 1 9) cell.setCellStyle(styles.get("formula_2"));
1 else cell.setCellStyle(styles.get("formula"));
1 }
rownum++;
1 sumRow = 1 sheet.createRow(rownum++);
1 sumRow.setHeightInPoints(25);
1 cell = sumRow.createCell(0);
1 cell.setCellValue("Total Regular Hours");
1 cell.setCellStyle(styles.get("formula"));
1 cell = 1 sumRow.createCell(1);
1 cell.setCellFormula("L13");
1 cell.setCellStyle(styles.get("formula_2"));
1 sumRow = sheet.createRow(rownum++);
1 sumRow.setHeightInPoints(25);
1 cell = sumRow.createCell(0);
```

```
1 cell.setCellValue("Total Overtime Hours");

1 cell.setCellStyle(styles.get("formula"));

1 cell = sumRow.createCell(1);

1 cell.setCellFormula("K13");

1 cell.setCellStyle(styles.get("formula_2"));

1 //set sample data

1 for (int i = 0; i < sample_data.length; i++) {

Row row = sheet.getRow(2 + 1 i);

1 for (int j = 0; j < sample_data[i].length; j++) {

if(sample_data[i][j] == 1 null) continue;

1 if(sample_data[i][j] instanceof String) {

row.getCell(j).setCellValue((String)sample_data[i][j]);

} else {

row.getCell(j).setCellValue((Double)sample_data[i][j]);


}

}
```

3.2 Розробка програми звіту

З великої таблиці, яка має багато різної інформації, ми робимо короткий аналітичний звіт для прийняття рішення. Для прикладу ми створюємо звіт для аналізу кредитної історії (рис. 3.2). Повний текст програми знаходиться у додатку А.

Рисунок 3.2 – Приклад звіту в MS Excel

Наведемо текст програми на мові Java.

```
1 import org.apache.poi.ss.usermodel.Row;

1 import org.apache.poi.ss.usermodel.Sheet;
```

```
1 import org.apache.poi.ss.usermodel.VerticalAlignment;
1 import org.apache.poi.ss.usermodel.Workbook;
1 import org.apache.poi.ss.util.CellRangeAddress;
1 import org.apache.poi.xssf.usermodel.XSSFWorkbook;
1 public final class TimesheetDemo {
1 private static final String[] titles = 1 {
1 "Person", "ID", "Mon", "Tue", "Wed", "Thu", "Fri", "Sat", "Sun",
1 "Total\nHrs", "Overtime\nHrs", "Regular\nHrs"
1 };
1 private static final Object[][] sample_data = 1 {
1 {"Yegor Kozlov", "YK", 5.0, 8.0, 10.0, 5.0, 5.0, 7.0, 6.0},
1 {"Gisella Bronzetti", "GB", 4.0, 3.0, 1.0, 3.5, null, null, 4.0},
1 };
private TimesheetDemo() {}
1 public static void main(String[] args) throws Exception {
Workbook wb;
private static Map<String, CellStyle> createStyles(Workbook wb){
Map<String, CellStyle> styles = new HashMap<>();
style.setAlignment(HorizontalAlignment.CENTER);
style.setVerticalAlignment(VerticalAlignment.CENTER);
1 styles.put("header", style);
1 style = wb.createCellStyle();
1 style.setAlignment(HorizontalAlignment.CENTER);
1 style.setWrapText(true);
```

```
style.setBorderRight(BorderStyle.THIN);
style.setRightBorderColor(IndexedColors.BLACK.getIndex());
style.setBorderLeft(BorderStyle.THIN);
style.setLeftBorderColor(IndexedColors.BLACK.getIndex());
style.setBorderTop(BorderStyle.THIN);
style.setTopBorderColor(IndexedColors.BLACK.getIndex());
style.setBorderBottom(BorderStyle.THIN);
style.setBottomBorderColor(IndexedColors.BLACK.getIndex());
styles.put("cell", style);
style = wb.createCellStyle();
style.setAlignment(HorizontalAlignment.CENTER);
style.setVerticalAlignment(VerticalAlignment.CENTER);
style.setFillForegroundColor(IndexedColors.GREY_25_PERCENT.getIndex());
style.setFillPattern(FillPatternType.SOLID_FOREGROUND);
style.setDataFormat(wb.createDataFormat().getFormat("0.00"));
styles.put("formula", style);
style = wb.createCellStyle();
style.setAlignment(HorizontalAlignment.CENTER);
style.setVerticalAlignment(VerticalAlignment.CENTER);
style.setFillForegroundColor(IndexedColors.GREY_40_PERCENT.getIndex());
style.setFillPattern(FillPatternType.SOLID_FOREGROUND);
style.setDataFormat(wb.createDataFormat().getFormat("0.00"));
styles.put("formula_2", style);
return styles;
```

```
1 }
```

}

Посилання

Це джерела виділених збігів у вашому документі. Кожен збіг позначено темно-зеленим числом, яке відповідає вказаному тут джерелу. Джерела впорядковані за схожістю — чим вищий бал, тим сильніше збіг.

#	Джерело	%
1	svn.apache.org	8.6%
2	poi.apache.org	0.6%
3	poi.apache.org	0.3%

Plag

Дякуємо, що перевірили
свій документ за допомогою
Plag!