

## Assignment

### MÔN: LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG

MÃ MÔN: 503005

### Đề bài: Module Quản lý cửa hàng cà phê

*(Sinh viên đọc kỹ tất cả hướng dẫn trước khi làm bài)*

#### I. Giới thiệu bài toán

Một cửa hàng cà phê cần quản lý thông tin về các mặt hàng, khách hàng và đơn hàng. Nhân viên của cửa hàng được chia làm nhân viên chính thức và nhân viên thời vụ. Trong bài này, sinh viên sẽ lập trình một số chức để quản lý các nghiệp vụ của cửa hàng.

Các chức năng mà sinh viên phải thực hiện là:

- Đọc file danh sách nhân viên.
- Dựa vào các thông tin từ đọc file, thao tác truy vấn theo yêu cầu.

#### II. Tài nguyên cung cấp

Source code được cung cấp sẵn bao gồm các file:

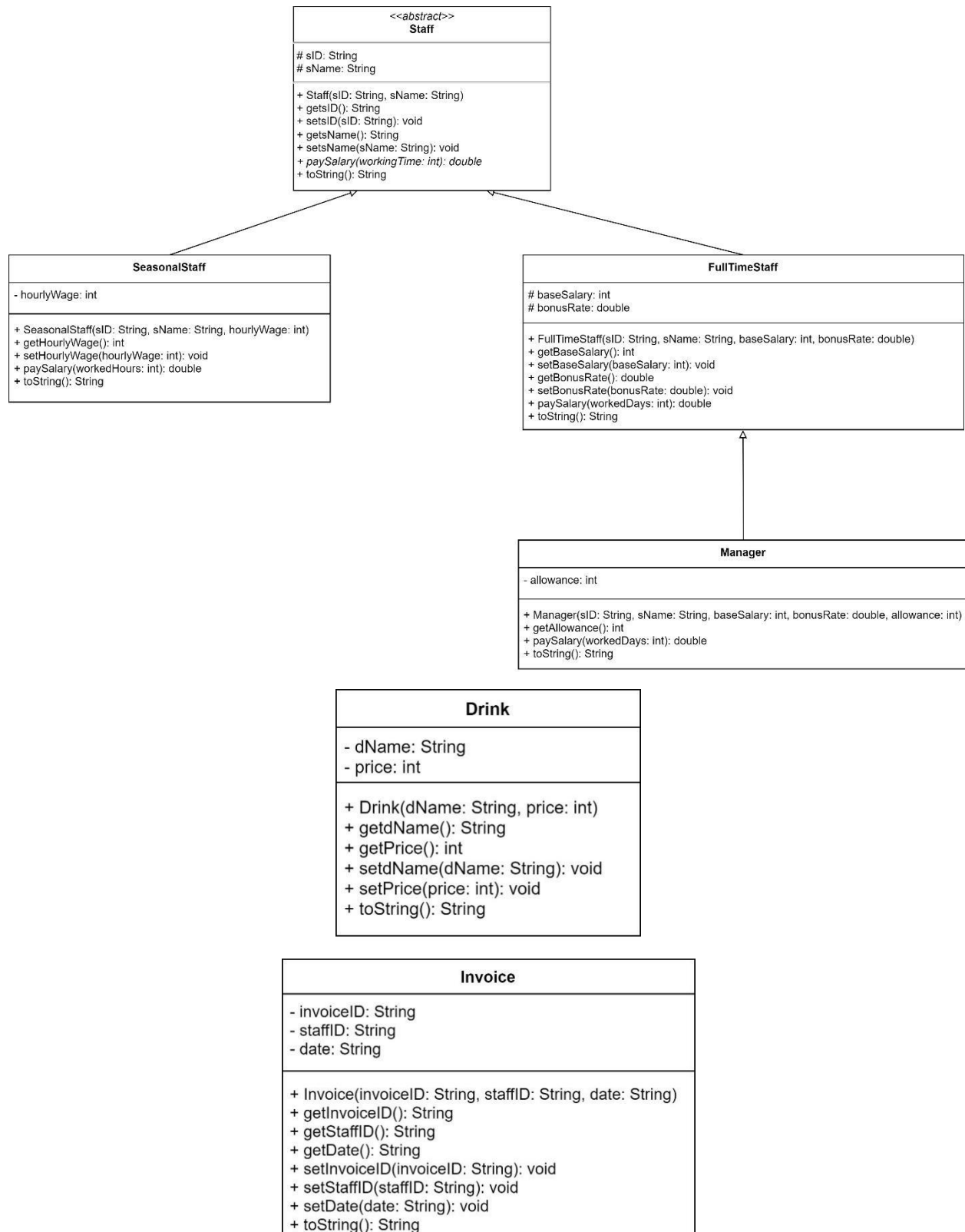
- File đầu vào và kết quả mong muốn:
  - o Folder *input* gồm 5 file:
    - *Drinks.txt*: chứa danh sách các món thức uống của cửa hàng
    - *InvoiceDetail.txt*: chứa chi tiết của các hóa đơn
    - *Invoices.txt*: chứa thông tin hóa đơn
    - *Staffs.txt*: chứa thông tin nhân viên của cửa hàng
    - *Timekeeping.txt*: chứa thông tin chấm công của nhân viên trong tháng
  - o Folder *output* gồm: 5 file *Req1.txt*, *Req2.txt*, *Req3.txt*, *Req4.txt*, *Req5.txt* chứa kết quả mẫu của các yêu cầu trong bài.
- File mã nguồn:
  - o *Test.java*: tạo đối tượng và gọi các phương thức sinh viên sẽ định nghĩa.
  - o *Staff.java*: chứa lớp **Staff** được định nghĩa sẵn. File này sinh viên không được chỉnh sửa.
  - o *Drink.java*, *Invoice.java*, *InvoiceDetails.java*: chứa các lớp được định nghĩa sẵn để tạo ra đối tượng tương ứng. File này sinh viên không được chỉnh sửa.
  - o *StoreManagement.java*: chứa lớp **StoreManagement** được định nghĩa sẵn thuộc tính *staffs* để chứa danh sách sinh viên, một số thuộc tính khác để chứa các danh sách đọc từ file, phương thức khởi tạo, các phương thức đọc file, phương thức ghi

file và một số phương thức trống. Sinh viên sẽ hiện thực thêm vào các phương thức còn trống trong file này và không chỉnh sửa phương thức có sẵn.

### III. Trình tự thực hiện bài

- Sinh viên tải file tài nguyên được cung cấp sẵn và giải nén.
- Trong cùng thư mục với 3 file cung cấp sẵn sinh viên tạo ra 3 file tương ứng với 3 lớp còn là **SeasonalStaff**, **FullTimeStaff**, **Manager**.
- Sinh viên tự hiện thực lớp **SeasonalStaff**, **FullTimeStaff**, **Manager** và code thêm vào lớp **StoreManagement** theo mô tả và yêu cầu trong các phần tiếp theo.
- Hiện thực phương thức đọc file để đọc các dữ liệu **Staff** vào danh sách *staffs*.
- Đọc kỹ các danh sách chứa đối tượng **Drink**, **Invoice** và **InvoiceDetail** đã được đọc sẵn vào chương trình, kết hợp các danh sách này cùng danh sách Staff đã đọc để hiện thực các yêu cầu của bài.
- Sau khi hiện thực các phương thức trong lớp **StoreManagement**, sinh viên biên dịch và chạy với phương thức **main** trong file *Test.java* đã gọi sẵn các phương thức. Sinh viên thực hiện các yêu cầu ở mục dưới và so sánh kết quả của mình với kết quả trong folder output đã được cung cấp sẵn.
- **Đối với các yêu cầu sinh viên không làm được vui lòng không xóa và phải đảm bảo chương trình chạy được với phương thức main trong Test.java được cung cấp sẵn.**
- **Đường link Google Drive gửi đề này cho sinh viên sẽ chứa kèm một file *version.txt*, sinh viên nên thường xuyên vào xem file này, nếu thấy có nội dung mới thì nên đọc kỹ mô tả và tải lại đề mới nhất. File này được dùng để thông báo nội dung chỉnh sửa và ngày chỉnh sửa trong trường hợp đề bị sai hoặc lỗi.**

### IV. Mô tả các lớp trong bài



InvoiceDetails
- invoiceID: String - dName: String - amount: int
+ InvoiceDetails(invoiceID: String, dName: String, amount: int) + getInvoiceID(): String + getDName(): String + getAmount(): int + setInvoiceID(invoiceID: String): void + setDName(dName: String): void + setAmount(amount: int): void + toString(): String

StoreManagement
- staffs: ArrayList<Staff> - workingHours: ArrayList<String> - invoices: ArrayList<Invoice> - invoiceDetails: ArrayList<InvoiceDetails> - drinks: ArrayList<Drink>
+ StoreManagement(staffPath: String, workingHoursPath: String, invoicesPath: String, detailsPath: String, drinkPath: String) + getStaffs(): ArrayList<Staff> + loadDrinks(filePath: String): ArrayList<Drink> + loadInvoices(filePath: String): ArrayList<Invoice> + loadInvoicesDetails(filePath: String): ArrayList<InvoiceDetails> + loadStaffs(filePath: String): ArrayList<Staff> + getTopFiveSeasonalStaffsHighSalary(): ArrayList<SeasonalStaff> + getFullTimeStaffsHaveSalaryGreaterThan(lowerBound: int): ArrayList<FullTimeStaff> + getStaffHighestBillInMonth(month: int): Staff + totalInQuarter(quarter: int): double + loadFile(filePath: String): ArrayList<String> + displayStaffs(): void + writeFile(path: String, list: ArrayList<E>): boolean + writeFile(path: String, object: E): boolean

- Lớp **SeasonalStaff** và lớp **FullTimeStaff** kế thừa từ lớp **Staff**, lớp **Manager** kế thừa từ lớp **FullTimeStaff**.
- Giải thích một số thuộc tính và phương thức như sau:
  - Lớp **Staff**
    - *sID*: mã nhân viên.
    - *sName*: tên nhân viên.
    - Phương thức abstract **paySalary(int workingTime)**.
    - **toString()**: trả về chuỗi theo định dạng: *sID\_sName*
  - Lớp **Drink**
    - *dName*: tên loại thức uống
    - *price*: giá loại thức uống
    - **toString()**: trả về chuỗi theo định dạng: *dName\_price*
  - Lớp **Invoice**
    - *invoiceID*: mã hóa đơn
    - *staffID*: mã nhân viên xuất hóa đơn

- *date*: ngày lập hóa đơn, lưu bằng chuỗi với định dạng dd/mm/yyyy
- **toString()**: trả về chuỗi theo định dạng: *invoiceID\_staffID\_date*
- **Lớp InvoiceDetails**
  - *invoiceID*: mã hóa đơn
  - *dName*: tên loại thức uống
  - *qty*: số lượng của loại thức uống trong hóa đơn
  - **toString()**: trả về chuỗi theo định dạng: *invoiceID\_dName\_qty*
- **Lớp StoreManagement**
  - *staffs*: danh sách nhân viên
  - *workingTime*: danh sách thời gian làm việc, danh sách này đã được đọc sẵn dưới dạng danh sách String.
  - *invoices*: danh sách hóa đơn, danh sách này đã được đọc sẵn dưới dạng danh sách Invoice.
  - *invoiceDetails*: danh sách chi tiết hóa đơn, danh sách này đã được đọc sẵn dưới dạng danh sách InvoiceDetails.
  - *drinks*: danh sách các loại thức uống, danh sách này đã được đọc sẵn dưới dạng danh sách Drink.
  - **StoreManagement(String staffPath, String workingTimePath, String invoicesPath, String detailsPath, String drinksPath)**: phương thức khởi tạo, gọi đến các phương thức đọc file để đọc thông tin từ các file vào các danh sách tương ứng.
  - **getStaffs(), setStaffs(ArrayList<Staff> staffs)**: dùng để lấy ra danh sách Staff và gán lại danh sách Staff. Hai phương thức này phục vụ cho quá trình chấm bài. Sinh viên không được sửa các phương thức này.
  - **loadDrinks(String filePath), loadInvoices(String filePath), loadInvoicesDetails(String filePath)**: các phương thức dùng để đọc thông tin vào file vào danh sách. Sinh viên không được sửa các phương thức này.
  - **public static ArrayList<String> loadFile(String filePath)**: phương thức hỗ trợ đọc thông tin từ file. Sinh viên có thể dùng phương thức này phục vụ cho bài làm của mình.
  - **displayStaffs()**: in danh sách *staffs* ra màn hình command prompt.
  - **writeFile(String path, ArrayList<E> list)**: ghi file danh sách truyền vào. Sinh viên không chỉnh sửa phương thức này.
  - **writeFile(String path, E object)**: ghi file đối tượng truyền vào. Sinh viên không chỉnh sửa phương thức này.
- **Lớp FullTimeStaff – Nhân viên chính thức**
  - *baseSalary*: lương cơ bản của nhân viên chính thức
  - *bonusRate*: hệ số tăng thêm của nhân viên chính thức

- **paySalary(int workedDays):** trả về lương, sẽ được mô tả ở bên dưới. Với workedDays là số ngày đã làm việc trong tháng.
- **toString():** trả về chuỗi theo định dạng:

*sID\_sName\_bonusRate\_baseSalary*

- **Lớp SeasonalStaff – Nhân viên thời vụ**

- *hourlyWage*: lương theo giờ
- **paySalary(int workedHours):** trả về lương, sẽ được mô tả ở bên dưới. Với workedHours là số giờ đã làm việc trong tháng.
- **toString():** trả về chuỗi theo định dạng:

*sID\_sName\_hourlyWage*

- **Lớp Manager – Quản lý:** Manager cũng là một nhân viên chính thức

- *allowance*: phụ cấp
- **paySalary(int workedDays):** trả về lương, sẽ được mô tả ở bên dưới. Với workedDays là số ngày đã làm việc trong tháng.
- **toString():** trả về chuỗi theo định dạng:

*sID\_sName\_bonusRate\_baseSalary\_allowance*

- Mô tả phương thức tính lương:

- **FullTimeStaff**

- **Lương = Lương cơ bản \* hệ số tăng thêm + thưởng**

Cách tính **thưởng**:

- Nếu nhân viên có số ngày làm (workedDays) <= 21: không có thưởng.
- Nếu nhân viên có số ngày làm (workedDays) > 21: từ ngày thứ 22 sẽ được thưởng 100,000 cho mỗi ngày làm.

Ví dụ: một nhân viên chính thức có lương là 10,000,000, có hệ số tăng thêm là 1.2 và đi làm 24 ngày trong tháng thì lương của nhân viên này sẽ là:

$$10,000,000 * 1.2 + (3 * 100,000)$$

- **Manager: FullTimeStaff là Manager** sẽ có thêm phụ cấp.

- **Lương = lương FullTimeStaff + phụ cấp**

- **SeasonalStaff**

- **Lương = lương theo giờ \* số giờ làm việc**

Với số giờ làm việc là workedHours.

## V. Mô tả file input và file output

- File input có tên *Staffs.txt* chứa danh sách nhân viên với mỗi dòng ứng với thuộc tính của 01 nhân viên được cách nhau bằng dấu “,” theo định dạng:

- **FullTimeStaff**

Mã nhân viên,Tên nhân viên,Lương cơ bản,Hệ số tăng thêm

CT004,Nguyen Thanh Huy,4000000,0.68

CT005,Tran Trung Nghia,3000000,5.89

- **Manager**

Mã nhân viên,Tên nhân viên,Lương cơ bản,Hệ số tăng thêm,Phụ cấp

QL003,Nguyen Pham Minh Khoa,10000000,1.5,3000000

- **SeasonalStaff**

Mã nhân viên,Tên nhân viên,Lương theo giờ

TV001,Cao Dang Khoa,12000

TV002,Tran Thanh Hung,25000

TV003,Nguyen Phung Nguyen,10000

- File input có tên *Drinks.txt*, *InvoiceDetails.txt*, *Invoices.txt* và *Timekeeping.txt* chứa thông tin các danh sách tương ứng, với mỗi dòng ứng với thuộc tính của một đối tượng được các nhau bằng dấu “,” theo định dạng:

- **Drink**

Tên loại thức uống,giá loại thức uống

Tra Sua Phuc Long,50000

- **Invoice**

Mã hóa đơn,Mã nhân viên,Ngày lập hóa đơn

HD004,TV008,26/01/2022

HD005,CT001,25/03/2022

HD006,TV006,10/04/2022

Với Mã nhân viên ứng với mã của các nhân viên trong danh sách *staffs*, Ngày lập hóa đơn lưu dưới định dạng dd/mm/yyyy.

- **InvoiceDetails**

Mã hóa đơn,Tên loại thức uống,Số lượng



HD029, Latte Macchiato, 2  
HD001, Tra Dao, 1  
HD045, Ca Phe Sua Da, 3

Với Mã hóa đơn ứng với mã của các hóa đơn trong danh sách *invoices*, Tên loại thức uống ứng với tên các loại thức uống trong danh sách *drinks*.

- **Timekeeping:** chứa danh sách chấm công của một tháng

Mã nhân viên, Thời gian làm việc

TV009, 50

TV010, 8

CT010, 16

QL001, 18

Nếu là nhân viên chính thức thì thời gian làm việc là số ngày làm việc, nếu là nhân viên thời vụ thì thời gian làm việc là số giờ làm việc.

- File output gồm có 5 file ứng với kết quả mẫu cho các yêu cầu:
  - *Req1.txt*: kết quả danh sách nhân viên đọc từ file.
  - *Req2.txt*: kết quả danh sách 05 nhân viên thời vụ có lương cao nhất trong tháng.
  - *Req3.txt*: kết quả danh sách nhân viên chính thức có lương > 15,000,000.
  - *Req4.txt*: kết quả doanh thu của quý 1.
  - *Req5.txt*: kết quả nhân viên có tổng hóa đơn đã lập cao nhất trong tháng 6.
- Mỗi dòng trong từng file output Req1, Req2, Req3, Req5 ứng với một đối tượng có định dạng theo phương thức **toString()** của lớp **tương ứng**.

### **Lưu ý:**

- Sinh viên có thể tự thêm dữ liệu vào file input để thử nhiều trường hợp khác nhau nhưng lưu ý thêm dữ liệu phải đúng định dạng đã được nêu bên trên.
- Sinh viên nên đọc kỹ phương thức **main** để xác định hướng hiện thực các lớp và các phương thức theo thứ tự.
- Sinh viên có thể thêm một số thao tác vào phương thức **main** để kiểm các phương thức đã định nghĩa tuy nhiên đảm bảo bài làm của sinh viên **phải chạy được trên phương thức main ban đầu đã cung cấp sẵn**.
- Sinh viên có thể thêm phương thức mới để phục vụ bài làm tuy nhiên bài làm của sinh viên phải chạy được với file Test.java đã được cung cấp sẵn.
- Sinh viên tuyệt đối không đặt các đường dẫn cứng trên máy khi định nghĩa phương thức liên quan đến đọc file. Nếu sinh viên tự ý đặt đường dẫn cứng dẫn đến quá trình chấm không đọc được file để chấm thì tương đương với bài làm biên dịch lỗi.



- Tuyệt đối không chỉnh sửa **tên của các phương thức** đã có sẵn (tuân theo đúng sơ đồ lớp phía trên).

## VI. Yêu cầu

Sinh viên không tự ý thêm thư viện ngoài, chỉ dùng những thư viện có sẵn trong các file nhận được.

Sinh viên thực hiện bài tập lớn trên Java 11 hoặc Java 8. Sinh viên không được dùng kiểu dữ liệu *var*. Bài làm của sinh viên sẽ được chấm trên Java 11, sinh viên tự chịu trách nhiệm tất cả các lỗi phát sinh nếu dùng các phiên bản Java khác.

### 1. YÊU CẦU 1 (2 điểm)

Sinh viên hiện thực phương thức **public ArrayList<Staff> loadStaffs(String filePath)** để đọc thông tin danh sách nhân viên từ file *Staff.txt* sau đó khởi tạo các đối tượng **Staff** và đưa vào danh sách *staffs*.

Sinh viên hiện thực phương thức, lệnh gọi phương thức khởi tạo *StoreManagement* trong phương thức **main** sẽ gọi đến các phương thức đọc file, nếu sinh viên làm đúng phương thức đọc file thêm nhân viên danh sách *staffs* thì phần ghi file gọi sẵn sẽ ghi ra file “Req1.txt” như file output mẫu.

**Lưu ý:** Đây là phương thức sinh viên phải định nghĩa được mới bắt đầu tính điểm cho bài, nếu không đọc được file thành danh sách các đối tượng **Staff** thì các yêu cầu bên dưới không được tính điểm.

**Các yêu cầu bên dưới thao tác trực tiếp trên các danh sách đã đọc từ file.**

### 2. YÊU CẦU 2 (2 điểm)

Hiện thực phương thức:

**public ArrayList<SeasonalStaff> getTopFiveSeasonalStaffsHighSalary()**

trả về danh sách 05 nhân viên thời vụ có lương cao nhất trong danh nhân viên.

Sinh viên hiện thực phương thức trên sao cho khi thực hiện các lệnh gọi phương thức trong phương thức **main** sẽ cho kết quả ghi ra file “Req2.txt” như file output mẫu.

### 3. YÊU CẦU 3 (2 điểm)

Hiện thực phương thức

**public ArrayList<FullTimeStaff> getFullTimeStaffsHaveSalaryGreaterThan(int lowerBound)**

trả về danh sách các nhân viên chính thức có lương lớn hơn giá trị **lowerBound** truyền vào. Giả sử **lowerBound** truyền vào là 15000000 thì danh sách trả về các nhân viên chính thức có lương > 15,000,000. Lưu ý quản lý (Manager) cũng là nhân viên chính thức.

Sinh viên hiện thực phương thức trên sao cho khi thực hiện các lệnh gọi phương thức trong phương thức **main** sẽ cho kết quả ghi ra file “Req3.txt” như file output mẫu.

#### 4. YÊU CẦU 4 (2 điểm)

Hiện thực phương thức

**public double totalInQuarter(int quarter)**

trả về tổng doanh thu theo quý **quarter** truyền vào. Biết quý 1 gồm các tháng 1, 2, 3, quý 2 gồm các tháng 4, 5, 6 và tương tự cho quý 3, quý 4. Tổng doanh thu được hiểu là tổng tiền các hóa đơn của các tháng trong quý. Tổng tiền của một hóa đơn được tính bằng cách tổng tiền của từng loại thức uống trên hóa đơn. Tổng tiền của từng loại thức uống là giá tiền của loại thức uống đó nhân với số lượng tại chi tiết hóa đơn.

Sinh viên hiện thực phương thức trên sao cho khi thực hiện các lệnh gọi phương thức trong phương thức **main** sẽ cho kết quả ghi ra file “Req4.txt” như file output mẫu.

#### 5. YÊU CẦU 5 (2 điểm)

Hiện thực phương thức

**public Staff getStaffHighestBillInMonth(int month)**

trả về nhân viên có tổng số tiền các hóa đơn lớn nhất trong tháng. Tổng số tiền các hóa đơn của một nhân viên trong tháng được tính bằng cách cộng tất cả tiền của hóa đơn do nhân viên đó lập trong tháng.

Sinh viên hiện thực phương thức trên sao cho khi thực hiện các lệnh gọi phương thức trong phương thức **main** sẽ cho kết quả ghi ra file “Req5.txt” như file output mẫu.

### VII. Lưu ý kiểm tra trước khi nộp bài

- Nếu sinh viên không thực hiện được yêu cầu nào thì để nguyên phương thức của yêu cầu đó, TUYỆT ĐỐI KHÔNG XÓA PHƯƠNG THỨC CỦA YÊU CẦU sẽ dẫn đến lỗi khi chạy phương thức **main**. Trước khi nộp phải kiểm tra chạy được với phương thức **main** được cho sẵn.
- Tất cả các file ReqX.txt ( $X = \{1,2,3,4,5\}$ ) được ghi ra cùng cấp thư mục với thư mục chứa các file source code. Đối với sinh viên sử dụng IDE (Eclipse, Netbean, ...) phải đảm bảo file chạy được bằng command prompt, đảm bảo bài làm không nằm trong package, vị trí ghi file kết quả ReqX.txt phải nằm cùng thư mục với file code.

- File kết quả ghi đúng thư mục khi chạy chương trình sẽ có dạng như sau:

input	08/05/2022 15:34	File folder	
Drink.class	08/05/2022 16:25	CLASS File	2 KB
Drink.java	30/04/2022 22:24	JAVA File	1 KB
FullTimeStaff.class	08/05/2022 16:25	CLASS File	2 KB
FullTimeStaff.java	03/05/2022 13:48	JAVA File	1 KB
Invoice.class	08/05/2022 16:25	CLASS File	2 KB
Invoice.java	28/04/2022 23:33	JAVA File	1 KB
InvoiceDetails.class	08/05/2022 16:25	CLASS File	2 KB
InvoiceDetails.java	30/04/2022 22:40	JAVA File	1 KB
Manager.class	08/05/2022 16:25	CLASS File	1 KB
Manager.java	03/05/2022 13:48	JAVA File	1 KB
Req1.txt	08/05/2022 16:25	Text Document	1 KB
Req2.txt	08/05/2022 16:25	Text Document	1 KB
Req3.txt	08/05/2022 16:25	Text Document	1 KB
Req4.txt	08/05/2022 16:25	Text Document	1 KB
Req5.txt	08/05/2022 16:25	Text Document	1 KB
SeasonalStaff.class	08/05/2022 16:25	CLASS File	2 KB
SeasonalStaff.java	08/05/2022 15:35	JAVA File	1 KB
Staff.class	08/05/2022 16:25	CLASS File	2 KB
Staff.java	03/05/2022 13:42	JAVA File	1 KB
StoreManagement.class	08/05/2022 16:25	CLASS File	9 KB
StoreManagement.java	08/05/2022 16:20	JAVA File	12 KB
Test.class	08/05/2022 16:25	CLASS File	2 KB
Test.java	08/05/2022 16:20	JAVA File	1 KB

## VIII. Hướng dẫn nộp bài

- Khi nộp bài sinh viên nộp lại file *FullTimeStaff.java*, *Manager.java*, *SeasonalStaff.java* và file *StoreManagement.java*, **không nộp kèm bất cứ file nào khác và tuyệt đối không được sửa tên 4 file này.**
- **Sinh viên đặt 4 file bài làm vào thư mục MSSV\_HoTen** (HoTen viết liền, không dấu) và nén lại với định dạng **.zip** nộp theo sự hướng dẫn của giảng viên thực hành.
- Trường hợp làm sai yêu cầu nộp bài (đặt tên thư mục sai, không để bài làm vào thư mục khi nộp, nộp dư file, ...) thì bài làm của sinh viên sẽ bị **0 điểm**.
- File nộp đúng sẽ như sau:
  - o File nén nộp bài:

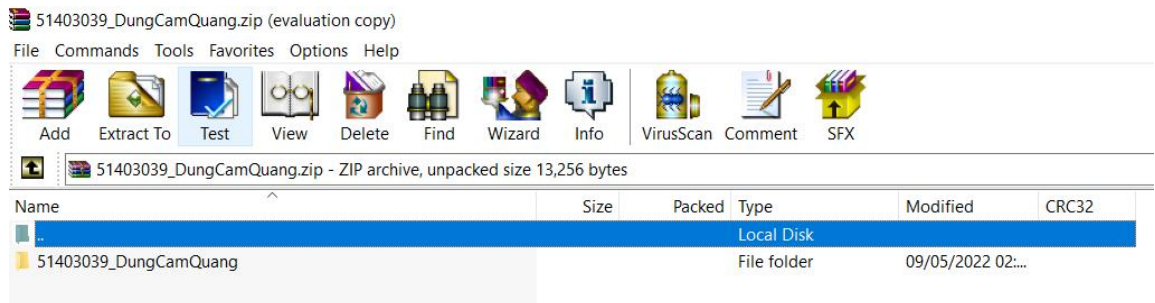
51403039\_DungCamQuang.zip

09/05/2022 02:33

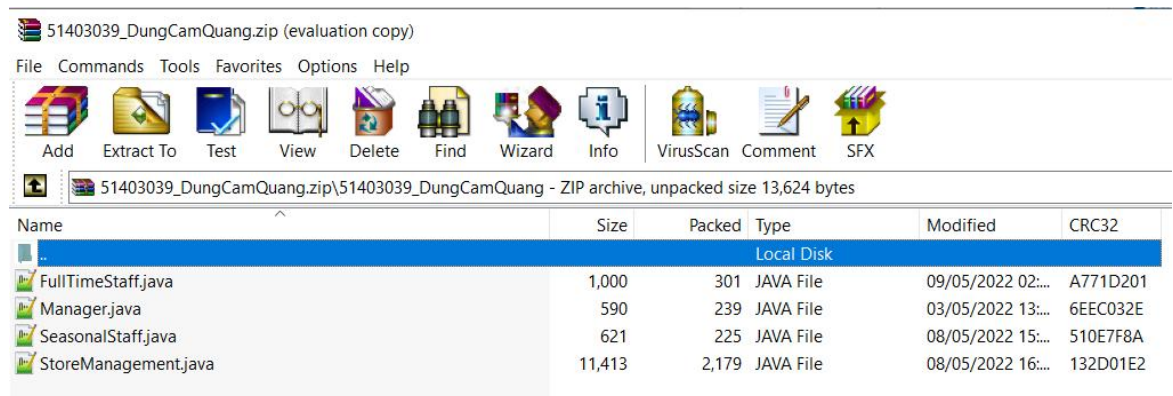
WinRAR ZIP archive

4 KB

- Bên trong file nén:



- Bên trong thư mục:



## IX. Đánh giá và quy định

- Bài làm sẽ được chấm tự động thông qua testcase (file input và output có định dạng như mẫu đã gửi kèm) do đó sinh viên tự chịu trách nhiệm nếu không thực hiện đúng theo Hướng dẫn nộp bài hoặc tự ý sửa tên các phương thức đã có sẵn dẫn đến bài làm không biên dịch được khi chấm.
- **Tất cả các trường hợp sinh viên gán cứng đường dẫn trong quá trình đọc file đều sẽ bị 0 điểm cả bài.**
- Testcase sử dụng để chấm bài là file khác với file sinh viên đã nhận, sinh viên chỉ được điểm mỗi YÊU CẦU khi chạy ra đúng hoàn toàn kết quả của yêu cầu đó.
- Nếu bài làm của sinh viên biên dịch bị lỗi trên máy chấm thì **0 điểm cả bài.**
- **Tất cả code sẽ được kiểm tra đạo văn. Mọi hành vi sao chép code trên mạng, chép bài bạn hoặc cho bạn chép bài nếu bị phát hiện đều sẽ bị điểm 0 vào điểm Quá trình 2 hoặc cấm thi cuối kì.**
- Nếu bài làm của sinh viên có dấu hiệu sao chép trên mạng hoặc sao chép nhau, sinh viên sẽ được gọi lên phòng vấn code để chứng minh bài làm là của mình.
- **Hạn chót nộp bài: 23h00 ngày 25/05/2022.**

– HẾT –