

Bài tập 1 – Multi thread

1. Viết một lớp để tạo task đếm tất cả các ước dương của các số từ 1 đến n ($n > 0$).
Viết lớp với phương thức main để tạo một số thread để thực hiện một số task trên (các task truyền tham số n khác nhau).
2. Viết một lớp để tạo task tính tổng $S(n) = x + x^2 + x^3 + \dots + x^n$ (x, n là số nguyên dương > 0 , S kiểu long).
Viết lớp với phương thức main để tạo một số thread để thực hiện một số task trên (các task truyền tham số n khác nhau).
3. Viết một lớp để tạo task đếm tất cả các số thỏa mãn chia hết cho 2 hoặc chia hết cho 3 nhưng không chia hết cho 6 của các số từ 1 đến n ($n > 1$).
Viết lớp với phương thức main để tạo một số thread để thực hiện một số task trên (các task truyền tham số n khác nhau).
4. Viết một lớp để tạo task tìm giá trị lớn nhất của mảng các số nguyên.
Viết lớp với phương thức main để tạo một số thread để thực hiện một số task trên (các task truyền tham số mảng khác nhau).
5. Viết một lớp để tạo task tìm số chẵn đầu tiên trong mảng các số nguyên. Nếu mảng không có giá trị chẵn thì trả về giá trị không chẵn là -1.
Viết lớp với phương thức main để tạo một số thread để thực hiện một số task trên (các task truyền tham số mảng khác nhau).

Bài tập 2 - IO

- a) Tạo ngẫu nhiên 100 số thực lưu vào file.
- b) Đọc từ file input, thống kê theo tiêu chí nào đó, lưu kết quả vào các file output.

Bài tập 3 - IO

Bạn biết bao nhiêu cách để có thể Reading/Writing nội dung của text file trong Java.

Đọc từ 1 text file bất kỳ:

- a) Thống kê số lượng từng từ trong file đó, sắp xếp theo chữ cái alphabet.

+ `getWordList(fileName : String) : Map<String, Integer>`

- b) Ghi kết quả câu a vào file khác.

+ `write2File(wordList: Map<String, Integer>, fileName: String): void`

- c) Đếm số lines có trong file.

+ `getLines(fileName: String): long`

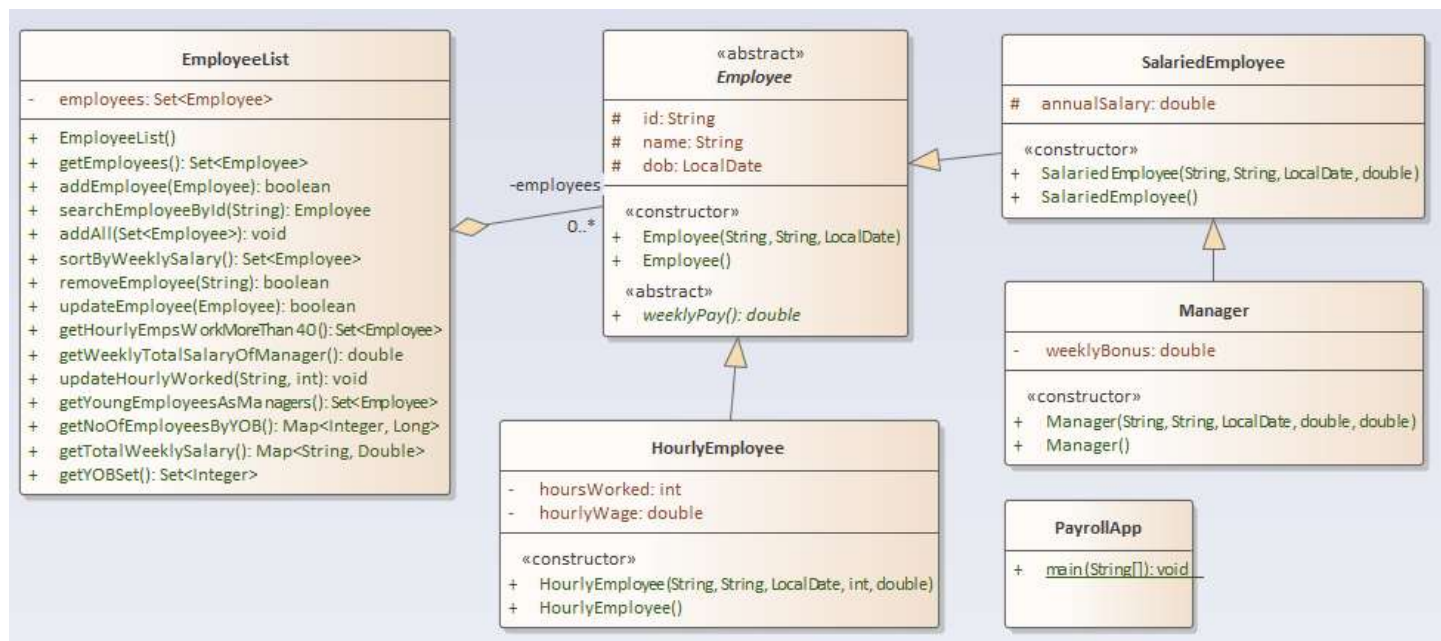
Bài tập 4 - Java Stream

Hiện thực xử lý bảng lương cho các loại nhân viên khác nhau.

- Nhân viên làm việc theo giờ được trả lương theo giờ, nhưng nếu họ làm việc hơn 40 giờ mỗi tuần, phần vượt quá sẽ được trả một rưỡi.
- Nhân viên làm công ăn lương được trả lương bất kể họ làm việc bao nhiêu giờ.
- Người quản lý là những nhân viên được trả lương, tiền lương gồm: Lương và thưởng.

Chương trình quản lý thông tin các nhân viên và tính toán tiền lương cần phải trả hằng tuần cho một tập hợp nhân viên (*dùng Set*).

1/ Hiện thực mô hình lớp sau



a. Với các lớp thực thể, viết các phương thức:

- Tạo 1 constructor đầy đủ tham số, 1 default constructor (*khởi tạo các giá trị mặc định*)
- Các phương thức setters / getters.
- Phương thức weeklyPay: Thanh toán tiền lương hằng tuần cho các nhân viên (*biết 1 năm có 52 tuần*)
- Viết phương thức toString biểu diễn dữ liệu như mẫu in ra bên dưới. Trong đó, kiểu ngày định dạng dd/MM/yyyy; Phân cách hàng nghìn lấy 2 số lẻ và đơn vị tiền tệ là \$ cho kiểu dữ liệu là tiền tệ.

b. Lớp EmployeeList, viết các phương thức

- Default constructor: Khởi tạo tập các nhân viên, dùng HashSet
- Phương thức trả về danh sách toàn bộ các nhân viên
+ getEmployees(): Set<Employee>
- Phương thức thêm 1 nhân viên vào danh sách, thêm không trùng mã số nhân viên.
+ addEmployee(emp: Employee): boolean
- Phương thức tìm kiếm nhân viên khi biết mã số. Trả về null, nếu không tìm thấy
+ searchEmployeeById(id: String): Employee
- Phương thức thêm một danh sách các nhân viên mới vào danh sách có trước, vẫn đảm bảo không bị trùng mã số nhân viên.

```
+ addAll (emps: Set<Employee>): void
```

- Phương thức sắp xếp danh sách nhân viên theo lương tuần tăng dần, trả về danh sách sau khi sắp xếp.

```
+ sortByWeeklySalary (): Set<Employee>
```

- Phương thức trích ra năm sinh của nhân viên, kết quả không trùng và sắp xếp giảm dần

```
+ getYOBSet(): Set<Integer>
```

- Phương thức xóa 1 nhân viên khi biết mã số.

+ removeEmployee (id: String): boolean

- Phương thức cập nhật thông tin của nhân viên khi biết mã số (*dùng chung 1 phương thức cho cả 3 loại nhân viên*)

+ updateEmployee (newInfor: Employee): boolean

- Phương thức trả về danh sách các nhân viên là nhân viên theo giờ và làm việc hơn 40 giờ một tuần

```
+ getHourlyEmpsWorkMoreThan40 (): Set<Employee>
```

- Phương thức trả về tổng lương hàng tuần của tất cả các nhân viên là người quản lý

```
+ getWeeklyTotalSalaryOfManager (): double
```

- Phương thức cập nhật giờ làm việc của một nhân viên theo giờ khi biết mã số. Ném lỗi nếu giờ làm việc mới nhỏ hơn 0, hoặc nếu mã nhân viên không tồn tại, hoặc nhân viên không phải là nhân viên theo giờ

```
+ updateHourlyWorked (id: String, newHour: int): void
```

- Phương thức trả về danh sách nhân viên trẻ làm quản lý (*có tuổi < 30, tuổi = năm hiện tại – năm sinh*)

```
+ getYoungEmployeesAsManagers (): Set<Employee>
```

- Phương thức thống kê số nhân viên theo năm sinh

```
+ getNoOfEmployeesByYOB(): Map<Integer, Long>
```

- Phương thức thống kê tổng tiền lương của từng loại nhân viên, kết quả sắp xếp theo tiền lương

```
+ getTotalWeeklySalary(): Map<String, Double>
```

Tất cả các Exception type được sử dụng là `IllegalArgumentException` và message phù hợp

3/ Viết lớp chứa phương thức main cho phần kiểm nghiệm.