

Bài tập lớn: Mô phỏng chương trình Quản Lý Dự Án:

I. Mô tả hệ thống:

- **User**: Người dùng của hệ thống, bao gồm 3 loại:
 - + **CEO**: Tổng giám đốc
 - + **Manager**: Quản lý
 - + **Employee**: Nhân viên
- **Employee** bao gồm 2 loại:
 - + **Developer**: Lập trình viên
 - + **Tester**: Kiểm thử
- **Project**: Dự án
- **Task**: Công việc, một Project có thể có nhiều Task
- **TaskActionLog**: Log ghi lại các thay đổi trạng thái của Task

Tất cả các đối tượng luôn luôn phải có 1 thuộc tính là **id** thuộc kiểu số nguyên.

Mỗi **User** phải bao gồm tối thiểu các thông tin:

- + **username** (tên đăng nhập)
- + **password** (mật khẩu, không cần mã hóa)
- + **name** (tên đầy đủ)

Một **Project** phải bao gồm tối thiểu các thông tin:

- + **name** (tên dự án)
- + **managerId** (id của Manager phụ trách)
- + **employeeIds** (id của các Employee tham gia)

Một **Task** phải bao gồm tối thiểu các thông tin:

- + **projectId** (id của dự án)
- + **developerIds** (id của các Developer thực hiện)
- + **testerIds** (id của các Tester thực hiện)
- + **createdDate** (ngày tạo)
- + **endDate** (ngày kết thúc thực tế)
- + **status** (trạng thái), bao gồm: **Created** (đã tạo mới), **Assigned** (đã gán cho Dev), **Started** (Dev bắt đầu làm), **Completed** (Dev đã xong), **Done** (Test confirm), **Reject** (Test từ chối), **Closed** (Đóng)

CEO là người tạo ra **Project** và có thể gán (giao trách nhiệm) cho 1 **Manager** để quản lý.

CEO có thể tạo mới **User**

Một **Project** bao gồm 1 **Manager** và nhiều **Employee**

Một **Task** gồm nhiều **Developer** và **Tester**

Một **Employee** có thể tham gia nhiều **Project** một lúc

Developer chỉ có thể chuyển trạng thái **Task** như sau: **Assigned->Started**, **Started->Completed**, **Reject->Started**

Tester chỉ có thể chuyển trạng thái **Task** như sau: **Completed->Done**, **Completed->Reject**

Manager chỉ có thể chuyển trạng thái Task như sau: Created (khi tạo mới Task), Created->Assigned (khi gán cho Dev), Done->Closed. Khi Task tạo mới phải cập nhật startDate là thời điểm tạo Task, khi Close phải cập nhật endDate.

II. Các chức năng:

Tạo giao diện dòng lệnh có các chức năng như sau:

1. Đăng nhập: Nhập username và password
 - + Nếu username/password sai thì báo lỗi, nếu **sai quá 3 lần thì tắt chương trình**
 - + Nếu đúng thì chuyển đến bước 2

Lưu ý: Tất cả các bước từ bước 2 về sau mỗi danh sách lựa chọn phải có lựa chọn Back (trở lại) màn hình trước đó để có thể thực hiện các thao tác.

2. Hiện các chức năng cho người dùng lựa chọn, bao gồm:
 - + Hiện danh sách Project -> chuyển sang bước 3
 - + Hiện danh sách Task đã tham gia (**chỉ với Employee**) -> Chuyển sang bước 4
 - + Hiện danh sách User (**chỉ với CEO**) -> Chuyển sang bước 5
 - + Hiện danh sách Employee (**chỉ với Manager**) -> Chuyển sang bước 6
3. Hiện danh sách các Project, bao gồm các thông tin cơ bản (tên dự án, tên người quản lý, trạng thái), số Task hiện có, số Task đã xong. Với Employee và Manager chỉ hiện danh sách mà họ có tham gia. User chọn một Project và hiện các chức năng:
 - + Tạo mới Project (**chỉ với CEO**) chuyển sang bước 3.1
 - + Hiện danh sách các Task -> Chuyển sang bước 4
 - + Xuất ra file csv thông tin các Project
4. Hiện danh sách các Task (các trường hợp ở bước 3 thì chỉ hiện Task thuộc Project đã chọn). Hiện các chức năng:
 - + Lọc danh sách: Hiện tất cả, lọc theo trạng thái, lọc theo ngày bắt đầu
 - + Chọn 1 Task -> Chuyển sang bước 7
 - + Tạo mới Task (**chỉ với Manager**)
 - + Xuất ra file csv thông tin các Task
 - + Xuất ra file csv thông tin các TaskActionLog của tất cả các Task
5. Hiện danh sách User. Hiện các chức năng:
 - + Lọc theo chức vụ
 - + Thêm mới User
6. Hiện danh sách các Employee. Mỗi Employee hiện đầy đủ:
 - + Các thông tin cơ bản
 - + Số project đã tham gia
 - + Số task được giao
7. Hiện thông tin task, danh sách các Employee tham gia và có các hành động:
 - + Đổi trạng thái Task (**tùy vào loại User** mà có các lựa chọn khác nhau), mỗi thay đổi phải **tạo thêm một TaskActionLog**
 - + Gán Task cho các Employee (chỉ với Manager)

III. Yêu cầu kỹ thuật:

- Tất cả các hằng số, các dạng trạng thái phải để static final hoặc Enum
- Bắt buộc phải có quan hệ thừa kế giữa các loại User và các loại Employee. Với mỗi loại User mà có các phương thức, thuộc tính đặc trưng, ví dụ:
 - + Employee có managerId để xác định ai là Manager của họ, các quan hệ khác cũng nên sử dụng id thay vì Object
 - + Developer có các phương thức completeTask(Task task), startTask(Task task),...
 - + Manager có các phương thức assign(Task task, Employee employee), close(Task task),...
 - + ...
- Mọi thao tác thay đổi thêm/sửa đều phải lưu trực tiếp file dữ liệu (khi tắt và mở lại chương trình thì dữ liệu mọi đã thay đổi trước đó đều phải giữ nguyên, không bị mất)
- Mọi thuộc tính phải có getter, setter tương ứng (trừ các thuộc tính static)

IV. Thang điểm: (tối đa 9đ)

Hoàn thành đầy đủ các bước:

- Bước 1: +1đ
- Bước 2: +1đ
- Bước 3: +1.5đ
- Bước 4: +2đ
- Bước 5: +1đ
- Bước 6: +1đ
- Bước 7: +1.5đ

Trừ điểm nếu:

- + Không có lựa chọn Back - Trở lại, phải thoát chương trình và bật lại mới có thể thực hiện các chức năng tiếp (trừ 0.5 điểm mỗi bước làm sai)
- + Làm sai yêu cầu kỹ thuật (trừ 1 điểm mỗi yêu cầu sai)
- + Phân quyền không đúng, User có thể thực hiện thao tác không được phép (trừ 0.25đ với mỗi thao tác sai)