BÀI TẬP THỰC HÀNH

Đặc tả bài toán:

Cho mô hình lớp của ứng dụng quản lý trung tâm thể thao. Một huấn luyện viên có thể tổ chức nhiều buổi tập với các thành viên khác nhau. Đồng thời, một thành viên có thể tham gia nhiều buổi tập với các huấn luyện viên khác nhau theo thời gian. Mỗi buổi tập cần lưu lại thông tin về ngày bắt đầu, ngày kết thúc và thời lượng (số giờ).

Trung tâm thể thao (Center) có nhiều huấn luyện viên làm việc. Mỗi huấn luyện viên chỉ thuộc về một trung tâm.

Trong hệ thống, chúng ta quản lý ba đối tượng chính là Trung tâm thể thao (Center), Huấn luyện viên (Trainer) và Thành viên (Member), cả hai đều kế thừa từ đối tượng Người (Person).

* Đối tượng Người (Person) có các thuộc tính cơ bản: Mã số (id), họ tên (name) và số điện thoại (phone).
* Đối với huấn luyện viên (Trainer), cần lưu thêm chuyên môn (specialty).
* Đối với thành viên (Member), cần lưu thêm ngày tham gia (joinDate).
* Đối với trung tâm (Center), cần lưu thêm địa chỉ (address).

Đặc biệt, buổi tập không được biểu diễn như một đối tượng riêng biệt mà được mô hình hóa như một mối quan hệ (relationship) giữa Huấn luyện viên và Thành viên. Mối quan hệ này có tên là TRAINED\_BY, chứa các thuộc tính: Ngày bắt đầu (startDate), Ngày kết thúc (endDate), Thời lượng (duration).

Class diagram (mô hình lớp):

Trainer (extends Person):

* specialty: String

Member (extends Person):

* joinDate: LocalDate

Person:

* id: String
* name: String
* phone: String

Center:

* id: String
* name: String
* address: String

Relationships:

* Member -[TRAINED\_BY {startDate, endDate, duration}]-> Trainer
* Trainer -[BELONGS\_TO]-> Center

Tạo project tên gồm: HọTênSV\_MãSốSV\_SốMáy\_[Server/Client]. Dùng ngôn ngữ lập trình JAVA kết nối Neo4j Java Driver và hiện thực chương trình dựa trên mô hình lập trình mạng client - server sử dụng socket.

Câu 1: Tạo database có tên là [tên và mã số của sinh viên] làm bài. Ví dụ: sinh viên tên Bảo và có mã số sinh viên là 20240123; thì tên database là Bao20240123.

* Viết các scripts để:
  + Tạo các unique index trên các thuộc tính id cho tất cả các node.
  + Load dữ liệu từ các CSV file cho trước vào database.
  + Thiết lập các relationships giữa các node như mô hình đồ thị cho trên.

Câu 2: Viết các lớp thực thể theo mô hình lớp trên.

Câu 3: Xây dựng các lớp với các phương thức CRUD theo các yêu cầu sau:

1. Thống kê số huấn luyện viên theo từng chuyên môn của một trung tâm thể thao nào đó khi biết tên trung tâm.

* getNoOfTrainersBySpecialty (centerName: String) : Map<String, Long> //Key: Chuyên môn; Value: Số huấn luyện viên

1. Thêm mới một huấn luyện viên vào trung tâm.

* addTrainerToCenter (trainer: Trainer, centerName: String): boolean

1. Dùng full-text search, tìm kiếm các huấn luyện viên theo chuyên môn.

* listTrainersBySpecialty (keyword: String): List

1. Cập nhật lại thời lượng buổi tập khi biết mã số thành viên và mã số huấn luyện viên. Lưu ý, chỉ được phép cập nhật khi buổi tập này vẫn còn đang diễn ra (tức ngày kết thúc buổi tập là null).

* updateSessionDuration(memberID: String, trainerID: String, newDuration: Float): boolean

Câu 4: Dùng Junit 5, viết lớp kiểm thử, tạo các test case cho các phương thức CRUD trên.

Câu 5: Hiện thực chương trình multi clients - server để thực thi các phương thức ở câu 3.