

Bài tập thực hành

Đặc tả bài toán:

Cho mô hình lớp một phần của ứng dụng quản lý khám điều trị bệnh tại bệnh viện. Một bác sĩ được quản lý bởi một khoa. Một bệnh nhân theo thời gian có thể được điều trị bệnh bởi nhiều bác sĩ. Với mỗi lượt điều trị bệnh, cần lưu lại chẩn đoán bệnh, ngày bắt đầu và kết thúc điều trị (*nếu ngày kết thúc là null thì bệnh nhân vẫn còn đang điều trị*).

Thông tin của khoa (*Department*) bao gồm: Mã khoa (*id*), tên khoa duy nhất (*name*) và địa điểm (*location*).

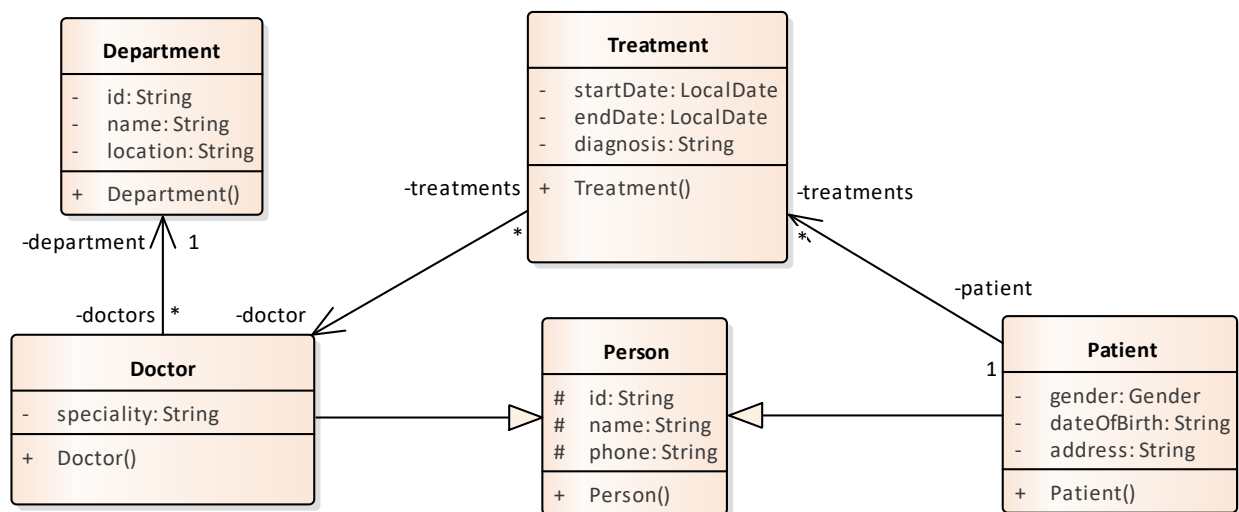
Thông tin chung của bệnh nhân (*Patient*) và bác sĩ (*Doctor*) cần lưu trữ bao gồm: Mã số (*id*), họ tên (*name*) và số điện thoại (*phone*).

Đối với bệnh nhân, cần lưu thêm: Giới tính (*gender*, gồm 3 giá trị: *Male*, *Famale* và *Other*), ngày sinh (*dateOfBirth*), địa chỉ (*address*).

Đối với bác sĩ, cần lưu thêm: Chuyên khoa (*speciality*).

Thông tin điều trị bệnh (*Treatment*) cần lưu thêm: Ngày bắt đầu (*startDate*), ngày kết thúc (*endDate*) và chẩn đoán (*diagnosis*).

Class diagram (mô hình lớp)



Graph model (mô hình đồ thị)



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 Lan và có mã số sinh viên là 19898981; thì tên database là Lan19898981.*

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.

Lưu ý: Sinh viên tạo thư mục và tạo file trong cùng project làm bài để lưu các scripts

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

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

a. Thêm mới một bác sỹ.

+ addDoctor (doctor: Doctor) : boolean

b. Thống kê số bác sỹ theo từng chuyên khoa (*speciality*) của một khoa (*department*) nào đó khi biết tên khoa.

+ getNoOfDoctorsBySpeciality (departmentName: String) : Map<String, Long>

//Key: Chuyên khoa; Value: Số bác sỹ

c. Dùng full-text search, tìm kiếm các bác sỹ theo chuyên khoa.

+ listDoctorsBySpeciality (keyword: String): List<Doctor>

d. Cập nhật lại chẩn đoán (*diagnosis*) của một lượt điều trị khi biết mã số bác sỹ và mã số bệnh nhân. Lưu ý, chỉ được phép cập nhật khi lượt điều trị này vẫn còn đang điều trị (*tức ngày kết thúc điều trị là null*)

+ updateDiagnosis(patientID: String, doctorID: String, newDiagnosis: String): boolean

Câu 4: Dùng Junit 5, viết lớp kiểm thử, tạo các test case cho các phương thức CRUD trên. Thêm bác sỹ mới phải là thông tin của sinh viên làm bài.

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