

JAVA – Assignment 10

Cách nộp bài: xem folder "Hướng dẫn nộp bài qua Git"

Chú ý:

Tạo 1 Project đặt tên là "TestingSystem_Assignment_10"

Tạo 3 package entity, backend, front-end

Trong phần backend sẽ tạo các class Exercise1, Exercise2, ... mỗi method là 1 Question

Trong front-end tạo các class Program1, Program2, Program3, ... để demo kết quả của các Exercise1, Exercise2, Exercise3, ...

Exercise 1: Basic

Question 1: (Sử dụng Database Testing System đã xây dựng ở SQL)

Tạo connection tới database Testing System

In ra "Connect success!" khi kết nối thành công.

Question 2:

Tạo method để in ra các thông tin của position gồm có id, name

Question 3:

Tạo method để tạo position, user sẽ nhập vào name.

Question 4:

Tạo method để update tên của position gồm có id = 5 thành "Dev".

Question 5:

Tạo method để delete của position theo id và user sẽ nhập vào id

Exercise 2: CRUD

Tạo class DepartmentDao trong package backend để cung cấp các method chuyên thao tác với table Department trong database. (các method cụ thể ở question bên dưới)

Question 1: read data – get list departments

Tạo method để lấy ra danh sách tất cả các Department

Gợi ý:

Viết method **getDepartments()** và return ra

List<Department>, Nếu có lỗi sẽ throw Exception lên front-end để in ra

Trên front-end sẽ gọi class DepartmentDao và demo method này

Question 2: read data – get department by id

Tạo method để lấy ra Department có id = 5

Gợi ý: Làm tương tự câu trên

Nếu không có department nào có id = 5 thì sẽ in ra là
"Cannot find department which has id = 5"

Nếu có lỗi sẽ throw Exception lên front-end để in ra

Trên front-end sẽ gọi class DepartmentDao và demo method này

Question 3: Tiếp tục Question 2 (read data – get department by id)

Không fix cứng id nữa mà sẽ dùng scanner để yêu cầu người dùng nhập vào id, sau đó trả về thông tin department có id như người dùng nhập vào

Gợi ý:

Trên backend sẽ viết method **getDepartmentById(int id)** và return ra **Department**

Nếu tìm thấy department có id = parameter thì sẽ return về department đó

Nếu không tìm thấy thì sẽ throw ra với message

"Cannot find department which has id = " + id

Nếu có lỗi sẽ throw Exception lên front-end để in ra

Trên front-end sẽ gọi class DepartmentDao và demo method này

Question 4: check data exists – check department name exists

Tạo method để check department name có tồn tại hay không?

Gợi ý:

Trên backend sẽ viết method

isDepartmentNameExists(String name) và return ra **boolean**

Nếu tìm thấy department đã có name = name parameter thì sẽ return true

Nếu không tìm thấy thì sẽ return ra false

Nếu có lỗi sẽ throw Exception lên front-end để in ra

Trên front-end sẽ gọi class DepartmentDao và demo method này

Question 5: create data – create department

Tạo method để người dùng có thể tạo được department

Gợi ý:

Trên backend sẽ viết method **void createDepartment (String name)**

Check xem đã có department nào có tên như parameter chưa (sử dụng method ở Question 4 để check)

Nếu tìm đã có department có tên như parameter thì throw ra Exception "Department Name is Exists!"
Nếu không tìm thấy thì sẽ create department
Nếu có lỗi sẽ throw Exception lên front-end để in ra

Trên front-end sẽ gọi class DepartmentDao và demo method này (dùng scanner để nhập thông tin của department muốn create)

Question 6: update data – update department

Tạo method để người dùng có thể update được department name

Gợi ý:

Trên backend sẽ viết method **void updateDepartmentName (int id, String newName)**

Check xem đã có department nào có id như id parameter chưa (Viết thêm method như Question 6 để check, tên method là **isDepartmentIdExists(int id)**)

Nếu không tìm thấy department có id = id parameter thì sẽ throw ra Exception có message "Cannot find department which has id = " + id

Nếu tìm thấy department có id = id parameter thì sẽ check xem tên mới của department có bị trùng không (sử dụng method ở Question 6 để check)

Nếu tìm đã có department có tên như parameter thì throw ra Exception "Department Name is Exists!"

Nếu không tìm thấy thì sẽ update department

Nếu có lỗi sẽ throw Exception lên front-end để in ra

Trên front-end sẽ gọi class DepartmentDao và demo method này (dùng scanner để nhập thông tin của department muốn update)

Question 7:

Tạo method để người dùng có thể xóa được department theo id mà user nhập vào

Gợi ý:

Trên backend sẽ viết method **void deleteDepartment (int id)**

Check xem đã có department nào có id như id parameter chưa (sử dụng method ở Question 6 để check)

Nếu không tìm thấy department có id = id parameter thì sẽ throw ra Exception có message "Cannot find department which has id = " + id

Nếu tìm thấy thì sẽ delete department đó

Nếu có lỗi sẽ throw Exception lên front-end để in ra

Trên front-end sẽ gọi class DepartmentDao và demo method này (dùng scanner để nhập vào id của department muốn delete)

Question 8:

Làm các chức năng CRUD tương tự với table Account

(Chú ý: trong chức năng getListAccount, getById thì phải in ra cả thông tin tên của department)

Exercise 3 (Optional): Call Procedure

Question 1:

Tạo method để yêu cầu người dùng nhập vào id của department, sau đó sẽ xóa department đó. (sử dụng store procedure ở trong MySQL)

Gợi ý:

B1: Tạo 1 store procedure trong MySQL tên là sp_delete_department() có in parameter là id

B2: Trên backend sẽ viết method **void deleteDepartmentUsingProcedure(int id)**

Check xem đã có department nào có id như id parameter chưa (sử dụng method ở Question 6 để check)

Nếu không tìm thấy department có id = id parameter thì sẽ throw ra Exception có message "Cannot find department which has id = " + id

Nếu tìm thấy thì sẽ delete department đó

Nếu có lỗi sẽ throw Exception lên front-end để in ra

Trên front-end sẽ gọi class DepartmentDao và demo method này (dùng scanner để nhập vào id của department muốn delete)

Exercise 4 (Optional): Transaction

Question 1:

Tạo 1 method để thực thi các bước sau:

B1: xóa tất cả các nhân viên thuộc phòng ban 3

B2: xóa phòng ban 3

Với điều kiện: khi xóa B1 bị lỗi thì sẽ rollback lại

Exercise 5: Tạo JdbcUtils

Question 1:

Tạo method để test xem đã kết nối thành công tới sql chưa?

Gợi ý: Tạo method **void isConnectedForTesting()** và trong method sẽ thực hiện kết nối tới database và in ra "Connect success!", nếu không thì sẽ throw ra exception

Question 2:

Tạo method để get Connect tới database.

Gợi ý:

Tạo method **void connect()** và trong method sẽ thực hiện check đã có connect hiện tại chưa (null & closed).

Nếu chưa connect thì sẽ thực hiện connect và return về connect vừa lấy được

Nếu connect rồi thì chỉ cần return ra connect hiện tại (không cần phải connect lại tới database)

Trong khi thực hiện nếu có lỗi gì thì sẽ throw ra Exception (để message Exception cho phù hợp)

Và Các constant nên config ở trong file properties (VD như : username, password, driverName, message exception)

Question 3:

Tạo method để get disconnect tới database.

Gợi ý:

Tạo method **void disconnect()** và trong method sẽ thực hiện check đã có connect hiện tại chưa (null & closed).

Nếu chưa connect thì sẽ không làm gì cả

Nếu đã connect rồi thì disconnect

Trong khi thực hiện nếu có lỗi gì thì sẽ throw ra Exception (để message Exception cho phù hợp)

Chú ý:

- Viết đúng coding convention
- Tuân thủ các best practice
- Không chép bài người khác (sẽ có hình thức phạt nếu bị phát hiện)