

JAVA – Assignment 4

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

Chú ý: Tạo 1 Project đặt tên là "TestingSystem_Assignment_4", tách mỗi exercise thành 1 class riêng, mỗi Question tách thành 1 method

Exercise 1 (Optional): Constructor

Question 1:

Tạo constructor cho department:

- a) không có parameters
- b) Có 1 parameter là nameDepartment và default id của Department = 0

Khởi tạo 1 Object với mỗi constructor ở trên

Question 2:

Tạo constructor cho Account:

- a) Không có parameters
- b) Có các parameter là id, Email, Username, FirstName, LastName (với FullName = FirstName + LastName)
- c) Có các parameter là id, Email, Username, FirstName, LastName (với FullName = FirstName + LastName) và Position của User, default createDate = now
- d) Có các parameter là id, Email, Username, FirstName, LastName (với FullName = FirstName + LastName) và Position của User, createDate

Khởi tạo 1 Object với mỗi constructor ở trên

Question 3:

Tạo constructor cho Group:

- a) không có parameters
- b) Có các parameter là GroupName, Creator, array Account[] accounts, CreateDate
- c) Có các parameter là GroupName, Creator, array String[] usernames, CreateDate

Với mỗi username thì sẽ khởi tạo 1 Account (chỉ có thông tin username, các thông tin còn lại = null).

Khởi tạo 1 Object với mỗi constructor ở trên

Exercise 2: Package

Question 1:

Tạo các package có tên như sau:

- com.vti.entity
- com.vti.frontend
- com.vti.backend

Question 2:

Trong package entity ta sẽ copy các Object đã tạo từ bài hôm trước như: Department, Account, Group, ...

Trong phần backend tạo các class Excercise1, Excercise2, Excercise3, ... Mỗi method trong mỗi Exercise là 1 question.

Trong front-end tạo các class Program1, Program2, Program3, ... để demo kết quả của các Excercise1, Excercise2, Excercise3, ...

Exercise 3: Access Modifier

Question 1: private access modifier

Thay đổi access modifier và tạo getter/ setter của những class trong package entity, frontend, backend cho phù hợp

Gợi ý:

- Các class trong package entity thì để access modifier của property là private và tạo getter, setter cho từng property.
- Các method là các question ở package back-end sẽ để là public để các class ở frontend có thể gọi được (không để static).

Exercise 4 : Encapsulation

Question 1:

Tạo Object Student có các property id, name, hometown, điểm học lực

- Tất cả các property sẽ để là private để các class khác không chỉnh sửa hay nhìn thấy
- Tạo constructor cho phép khi khởi tạo mỗi student thì người dùng sẽ nhập vào tên, hometown và có điểm học lực = 0
- Tạo 1 method cho phép set điểm vào
- Tạo 1 method cho phép cộng thêm điểm
- Tạo 1 method để in ra thông tin của sinh viên bao gồm có tên, điểm học lực (nếu điểm <4.0 thì sẽ in ra là Yếu, nếu điểm > 4.0 và < 6.0 thì sẽ in ra là trung bình, nếu điểm > 6.0 và < 8.0 thì sẽ in ra là khá, nếu > 8.0 thì in ra là Giỏi)

Demo các chức năng trên bằng class ở front-end.

Question 2 (Optional):

Tạo class phù hợp cho thiết kế sau:

Circle	Account	Date
- radius: double = 1.0 - color: String = 'red'	- id: String, - name: String, - balance: int	- day: int - month: int - year: int
+ Circle() + Circle(radius: double) + Circle(radius: double, color: String) + getRadius(): double + setRadius(radius: double) + getColor(): String + setColor(color: String) + getRadius(): double + getArea(): double + toString(): String	+ Account(id: String, name: String, balance: int) + getID(): String + getName: String, + getBalance(): int + credit(amount: int) + debit(amount: int) + tranferTo(account: Account, int amount)	+ Date(int day, int month, int year) + getDay(): int + getMonth(): int + getYear(): int + setDay(day: int) + setMonth(month: int) + setYear(year: int) + toString(): String + isLeapYear(): boolean

Exercise 5: Inheritance

Chú ý: áp dụng cả encapsulation cho các question ở dưới

Question 1: inheritance

Một đơn vị sản xuất gồm có các cán bộ là công nhân, kỹ sư, nhân viên.
Mỗi cán bộ cần quản lý các dữ liệu: Họ tên, tuổi, giới tính(name, nữ, khác), địa chỉ.

Cấp công nhân sẽ có thêm các thuộc tính riêng: Bậc (1 đến 10).

Cấp kỹ sư có thuộc tính riêng: Ngành đào tạo.

Các nhân viên có thuộc tính riêng: công việc.

Hãy xây dựng các lớp CongNhan, KySu, NhanVien kế thừa từ lớp CanBo.

Question 2: Tiếp tục Question 1

Xây dựng lớp QLCB(quản lý cán bộ) cài đặt các phương thức thực hiện các chức năng sau:

- Thêm mới cán bộ.
- Tìm kiếm theo họ tên.
- Hiện thị thông tin về danh sách các cán bộ.
- Nhập vào tên của cán bộ và delete cán bộ đó
- Thoát khỏi chương trình.

Question 3 (Optional): constructor inheritance

Tạo class abstract Person gồm các property name và tạo constructor có 1 parameter name

Tạo class abstract Student kế thừa Person gồm các property id, name và tạo constructor có 2 parameter id, name

Tạo class HighSchoolStudent kế thừa Student bao gồm các property id, name, clazz (Lớp đang học), desiredUniversity (trường đại học mong muốn vào) và tạo constructor có 4 parameter id, name, clazz, desiredUniversity.

Hãy khởi tạo Object HighSchoolStudent với các giá trị: id = 1, name = "Nam", clazz = "Chuyên Văn", desiredUniversity = "Đại học công nghệ"

Question 4 (Optional):

Một thư viện cần quản lý các tài liệu bao gồm Sách, Tạp chí, Báo. Mỗi tài liệu gồm có các thuộc tính sau: Mã tài liệu(Mã tài liệu là duy nhất), Tên nhà xuất bản, số bản phát hành.

Các loại sách cần quản lý thêm các thuộc tính: tên tác giả, số trang.

Các tạp chí cần quản lý thêm: Số phát hành, tháng phát hành.

Các báo cần quản lý thêm: Ngày phát hành.

Xây dựng chương trình để quản lý tài liệu (QLTV) cho thư viện một cách hiệu quả.

Xây dựng lớp QuanLySach có các chức năng sau

- Thêm mới tài liệu: Sách, tạp chí, báo.

- b) Xóa tài liệu theo mã tài liệu.
- c) Hiện thị thông tin về tài liệu.
- d) Tìm kiếm tài liệu theo loại: Sách, tạp chí, báo.
- e) Thoát khỏi chương trình.

Exercise 6: Abstraction

Question 1: abstract class & abstract method

Tạo 1 class Phone để lưu thông tin liên lạc, trong Phone có chứa 1 property có kiểu dữ liệu là Array contacts, trong từng contact có lưu thông tin number, name

- a) Tạo các abstract method
 - a. void insertContact(String name, String phone)
 - b. void removeContact(String name)
 - c. void updateContact(String name, String newPhone)
 - d. void searchContact(String name)
- b) Tạo class VietnamesePhone kế thừa Phone và triển khai các method abstract
Viết chương trình demo

Question 2 (Optional):

Hãy xây dựng 1 class User với các thuộc tính name kiểu String và salary ratio kiểu double.

- a) Viết các hàm cho phép nhập và trả về name, salary ratio.
- b) Viết hàm calculatePay() dạng abstract trả về thu nhập của nhân viên, kiểu double.
- c) Viết hàm displayInfor()

Question 3 (Optional): Tiếp tục Question 2

Viết class Employee, Manager, Waiter kế thừa User như sau:

Implement method calculatePay() như sau:

Đối với Employee sẽ được tính = salary ratio * 420

Đối với Manager sẽ được tính = salary ratio * 520

Đối với Waiter sẽ được tính = salary ratio * 220

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)