

BÀI TẬP JAVA OOP CƠ BẢN 1

Bài 1.

Viết chương trình tính diện tích, chu vi hình chữ nhật.

Hãy viết lớp **HìnhChuNhat** gồm có:

- Attributes : chiều dài, chiều rộng.
- Phương thức thiết lập (set), và lấy (get) thông tin chiều dài, chiều rộng.
- Phương thức tính diện tích, chu vi.
- Phương thức toString gồm các thông tin dài, rộng, diện tích, chu vi.

Xây dựng lớp chứa hàm main cho phần kiểm nghiệm. Chiều dài và chiều rộng có thể nhập từ bàn phím.

Bài 2.

Viết chương trình quản lý sinh viên: Nhập, xuất thông tin, tính điểm TB.

Viết lớp **SinhVien** như sau:

❖ Attributes (private):

- Mã sinh viên <int>.
- Họ tên <String> .
- Điểm LT, điểm TH <float>.

❖ Constructor:

- Constructor mặc định (khởi tạo đối tượng với các thông tin kiểu số là 0, kiểu chuỗi là chuỗi rỗng).
- Constructor thứ hai nhận đầy đủ thông tin để khởi tạo giá trị cho tất cả các biến instance.

❖ Methods:

- Các getter và setter cho mỗi thuộc tính.
- Tính điểm trung bình.

- Phương thức toString để diễn tả đối tượng ở dạng chuỗi.

Xây dựng class chứa hàm main: tạo 3 đối tượng sinh viên sv1, sv2, sv3.

- **sv1** chứa thông tin của chính mình (tạo bằng constructor đủ thông số, thông tin biết rồi khỏi nhập từ bàn phím).
- **sv2** là thông tin người bạn thân nhất của em (tạo bằng constructor đủ thông số, thông tin biết rồi khỏi nhập từ bàn phím).
- **sv3** tạo bằng constructor mặc định. Nhập các thông tin cho sv3 từ bàn phím rồi sau đó dùng các setter để gán vào cho các thuộc tính tương ứng.

In bảng danh sách sinh viên gồm 4 cột là MSSV, họ tên, điểm LT, điểm TH, điểm TB (bảng có 3 dòng cho 3 sinh viên).

Bài 3.

Sở giao thông cần theo dõi việc đăng ký xe của người dân. Dựa vào thông tin trị giá xe và dung tích xy lanh của xe, sở giao thông cũng tính mức thuế phải đóng trước bạ khi mua xe như sau:

- Dưới 100cc, 1% trị giá xe.
- Từ 100 đến 200cc, 3% trị giá xe.
- Trên 200cc, 5% trị giá xe.

Hãy thiết kế và cài đặt class **Vehicle** với các attributes và methods phù hợp. Class phải có các constructor và phải bảo đảm tính encapsulation.

Xây dựng class chứa hàm main. Hàm main in ra menu các lựa chọn:

1. Nhập thông tin và tạo các đối tượng xe1, xe2, xe3.
2. Xuất bảng kê khai tiền thuế trước bạ của các xe.
3. Thoát.

Bài 4.

Ngân hàng ABC muốn lưu trữ thông tin của mỗi tài khoản như sau:

❖ Mỗi tài khoản chứa các thông tin:

- Số tài khoản <long>.
- Tên tài khoản <String>.
- Số tiền trong tài khoản <Double>.

(a). Thiết kế lớp **Account** để lưu trữ thông tin bao gồm các phương thức sau:

- Constructor: Có 2 constructor (mặc định và đầy đủ tham số).
- Các phương thức get, set cho từng thuộc tính.
- Phương thức *toString* để trả về chuỗi chứa toàn bộ thông tin tài khoản, yêu cầu định dạng tiền tệ.

(b). Thêm các thông tin sau vào lớp Account

- Hằng số lãi suất có giá trị khởi tạo **0.035**
- *Constructor có 2 đối số*: số tài khoản, tên tài khoản. Constructor này sẽ khởi tạo số tiền mặc định là **50**
- *Phương thức nạp tiền*: Lấy số tiền hiện tại trong tài khoản cộng với số tiền nạp vào.
- *Phương thức rút tiền*: Lấy số tiền hiện tại trong tài khoản – (số tiền muốn rút + phí rút tiền)
- *Phương thức đáo hạn*: Mỗi lần đến kỳ đáo hạn thì số tiền trong tài khoản = số tiền trong tài khoản + số tiền trong tài khoản * LAISUAT
- Phương thức chuyển khoản từ tài khoản này sang tài khoản khác.

Bài 5.

Một phân số gồm có tử số (numerator) và mẫu số (denominator).

Hãy thiết kế class **PhanSo** (Rational) để thực hiện các chức năng sau:

- Tối giản phân số (reduce)
- Nghịch đảo phân số (reciprocal)
- Cộng 2 phân số (add) - Trừ 2 phân số (subtract)
- Nhân 2 phân số (multiply)
- Chia 2 phân số (divide)
- So sánh 2 phân số, sử dụng sai số 0.0001 (tolerance) để so sánh.

Bài 6.

Viết lớp **HangThucPham** mô tả một hàng hóa là hàng thực phẩm trong kho của một siêu thị, có các thuộc tính:

- mã hàng (*không cho phép sửa, không được để rỗng*),
- tên hàng (*không được để rỗng*),
- đơn giá (>0),
- ngày sản xuất và ngày hết hạn (*ngày không được để rỗng, ngày hết hạn phải sau ngày sản xuất*).

Ràng buộc chặt chẽ các ràng buộc trên các trường dữ liệu. Nếu dữ liệu không hợp lệ thì gán giá trị mặc định cho phép tương ứng của trường đó.

- Tạo 1 constructor có đầy đủ tham số, 1 constructor có tham số là mã hàng.
- Viết các phương thức setters/getters.
- Viết phương thức kiểm tra một hàng thực phẩm đã hết hạn chưa?
- Phương thức toString, trả về chuỗi chứa thông tin của hàng thực phẩm.

Trong đó: Định dạng đơn giá có phân cách hàng nghìn. Định dạng kiểu ngày là dd/MM/yyyy. Viết lớp cho phần kiểm nghiệm.

Bài 7.

Viết chương trình xây dựng đối tượng Sinh viên gồm có các thuộc tính sau:

- Mã sinh viên <int>,
- Họ tên <String>,
- Địa chỉ <String>,
- Số điện thoại <long>
- Các thuộc tính khai báo private, định nghĩa các phương thức get/set cho từng thuộc tính.
- Viết các constructor để khởi tạo đối tượng (constructor mặc định, constructor có tham số).
- Override phương thức toString của lớp Object.

Xây dựng lớp cho phép nhập vào N sinh viên (dùng mảng). Xuất danh sách trên ra màn hình theo thứ tự tăng dần của mã sinh viên.

Bài 8.

Viết chương trình xây dựng đối tượng CD gồm có các thuộc tính sau:

- Mã CD <int>,
- Tựa CD <String>,
- Ca sỹ <String>,
- Số bài hát <int> (>0)
- Giá thành <float> (>0)

Note: Các thuộc tính khai báo private, định nghĩa các phương thức get/set cho từng thuộc tính.

- Viết các constructor để khởi tạo đối tượng CD.
- Override phương thức toString của lớp Object.

Xây dựng lớp lưu danh sách các CD (dùng mảng).

- Phương thức thêm 1 CD vào danh sách, thêm thành công nếu không trùng mã CD và kích thước mảng còn cho phép.
- Tính số lượng CD có trong danh sách.
- Tính tổng giá thành của các CD.
- Phương thức sắp xếp danh sách giảm dần theo giá thành.
- Phương thức sắp xếp danh sách tăng dần theo tựa CD.
- Phương thức xuất toàn bộ danh sách.

Viết lớp cho phần kiểm nghiệm. Dùng menu case thực hiện các chức năng theo yêu cầu