

## Bài thực hành 2 – Lớp và đối tượng

**Bài 1:** Viết lớp `Diem` để thao tác với điểm trong không gian hai chiều, lớp này chứa 2 thuộc tính kiểu số thực thể hiện hoành độ, tung độ và các phương thức:

- Phương thức khởi tạo hai tham số để tạo đối tượng với hoành độ và tung độ chỉ định.
- Các phương thức lấy giá trị (được gọi là các phương thức getter) và thiết lập giá trị (được gọi là các phương thức setter) của các thuộc tính tương ứng.
- Phương thức hiển thị thông tin một điểm dạng (hoành độ, tung độ).
- Phương thức tính khoảng cách giữa hai điểm. Biết công thức tính khoảng cách hai điểm  $A(x_1, y_1)$  và  $B(x_2, y_2)$  như sau:

$$\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

Viết phương thức `main()` kiểm tra các chức năng của lớp `Diem`.

**Bài 2:** Xây dựng lớp `DoanThang` có 2 thuộc tính kiểu `Diem` thể hiện 2 điểm của 2 đầu đoạn thẳng và các phương thức:

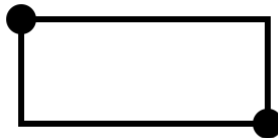
- Phương thức khởi tạo có hai tham số để khởi gán cho hai đầu đoạn thẳng.
- Phương thức getter và setter của các thuộc tính.
- Phương thức hiển thị thông tin đoạn thẳng dạng  $[(x_1, y_1), (x_2, y_2)]$
- Phương thức tính độ dài đoạn thẳng.
- Phương thức tìm trung điểm đoạn thẳng.
- Phương thức kiểm tra hai đoạn thẳng song song? Biết AB song song CD khi  $(A(x_a, y_a), B(x_b, y_b), C(x_c, y_c)$  và  $D(x_d, y_d))$

$$\frac{x_a - x_b}{y_a - y_b} = \frac{x_c - x_d}{y_c - y_d}$$

Viết phương thức `main()` kiểm tra các chức năng của lớp `DoanThang`.

**Bài 3:** Một hình chữ nhật xác định bằng toạ độ điểm trên trái và toạ độ điểm dưới phải.

Điểm trên trái



Điểm dưới phải

Lớp `HinhChuNhat` có 2 thuộc tính là toạ độ điểm trên trái, toạ độ điểm dưới phải của hình chữ nhật và các phương thức:

- Phương thức khởi tạo 2 tham số khởi tạo cho hai thuộc tính.
- Phương thức tính diện tích và chu vi của hình chữ nhật.
- Phương thức hiển thị để hiển thị các thuộc tính, diện tích và chu vi của hình chữ nhật.

**Bài 4:** Viết lớp `PhanSo` gồm hai thuộc tính nguyên là tử số, mẫu số của phân số và một thuộc tính tĩnh chứa số phân số đã được tạo. Lớp này có hai phương thức khởi tạo:

- Phương thức khởi tạo không tham số khởi gán tử số là 0, mẫu là số 1.
- Phương thức khởi tạo hai tham số tương ứng khởi gán giá trị tử số và mẫu số cho phân số.

Các phương thức tĩnh

- Tìm ước số chung lớn nhất hai số nguyên.

Các phương thức getter, setter của tử số, mẫu số và các phương thức khác thực hiện các chức năng sau:

- Rút gọn phân số.
- Các phép tính cộng, trừ, nhân, chia hai phân số. Kết quả trả về của các phép tính này là các phân số đã được rút gọn.
- So sánh bằng, lớn hơn, bé hơn hai phân số.
- Hiện thị phân số dạng tửSố/mẫuSố.

Viết phương thức `main()` kiểm tra các chức năng lớp `PhanSo`.

**Bài 5:** Viết lớp `DsPhanSo` quản lý danh sách phân số có một thuộc tính dạng `ArrayList<PhanSo>` để quản lý danh sách các phân số.

Lớp này có các phương thức sau:

- Thêm một phân số vào danh sách.
- Xoá một phân số khỏi danh sách.
- Tính tổng các phân số, kết quả là phân số đã rút gọn.
- Sắp xếp danh sách phân số tăng dần.
- Hiện thị danh sách phân số.

Viết phương thức `main()` kiểm tra các chức năng của phương thức `DsPhanSo`.

**Bài 6:** Viết chương trình quản lý thông tin học viên của một trung tâm gia sư. Một học viên bao gồm thông tin: mã số học viên, họ tên, ngày sinh, quê quán, điểm của ba môn học chính. Trong đó mã học viên là số nguyên tự động tăng khi tạo đối tượng.

Thông tin học viên sẽ được nhập vào đầu khoá học khi học viên đăng ký, và thông tin điểm 3 môn học chính sẽ được nhập cuối khoá.

Vào cuối khoá học, trung tâm muốn tìm ra một số học viên có thành tích học tập tốt để trao học bổng khuyến khích. Một học viên được đánh giá là tốt nếu điểm trung bình ba môn học chính từ **8.0 trở lên và không có môn nào trong ba môn chính điểm dưới 5**. Chương trình bao gồm các chức năng sau:

- Nhập danh sách học viên.

- Đọc danh sách học viên từ tập tin hocvien.txt cấu trúc tập tin gồm nhiều dòng, mỗi 3 dòng là thông tin của 1 học viên gồm họ tên, quê quán và ngày sinh (dd/MM/yyyy) học viên.
- Nhập điểm cho các học viên.
- Hiện thị danh sách học viên, mỗi học viên hiện thị thông tin họ tên, quê quán, ngày sinh (dd/MM/yyyy) và điểm các môn học (nếu có).
- Tìm kiếm học viên theo mã số hoặc họ tên, hoặc quê quán.
- Sắp xếp danh sách học viên theo điểm các môn học.
- Xác định danh sách học viên để trao học bổng. Xuất danh sách sinh viên có học bổng ra một tập tin hocbong.txt, mỗi học viên hiện thị trên 1 dòng với định dạng: mã học viên – tên học viên – điểm trung bình.
- Thống kê số lượng học viên có độ tuổi nhỏ hơn 18, từ 18 đến 23, từ 24 trở lên.

*Gợi ý*

Để thuộc tính maHV tăng tự động, ta sử dụng một thuộc tính tĩnh và khởi gán giá trị cho mã học viên như sau:

```
class HocVien {  
    private static int dem = 0;  
    private int maHV;  
    {  
        dem++;  
        this.maHV = dem;  
    }  
    // ...  
}
```

Minh hoạ định dạng hiển thị dữ liệu kiểu Date theo định dạng trước

```
Date d = new Date();  
SimpleDateFormat f = new SimpleDateFormat("dd/mm/yyyy");  
System.out.println(f.format(d));
```

Minh hoạ chuyển chuỗi ngày tháng thành kiểu Date

```
SimpleDateFormat f = new SimpleDateFormat("dd/mm/yyyy");  
Date d = f.parse("02/09/2019");
```

**Bài 7\*:** Viết chương trình thi trắc nghiệm ngữ pháp tiếng Anh, mỗi câu hỏi gồm các thông tin: nội dung câu hỏi, 4 phương án chọn và 1 đáp án chính xác. Mỗi câu hỏi là một tập tin .txt lưu trong thư mục CauHoi. Cấu trúc tập tin .txt chứa câu hỏi:

- Dòng 1 là nội dung câu hỏi.
- Dòng 2 là lựa chọn thứ nhất.
- Dòng 3 là lựa chọn thứ hai.
- Dòng 4 là lựa chọn thứ ba.
- Dòng 5 là lựa chọn thứ tư.
- Dòng 6 là đáp án chính xác cho câu hỏi (đánh theo A, B, C, D).

Ví dụ một tập tin chứa nội dung câu hỏi

```
Take care ..., Ann  
you  
your  
yours  
yourself  
D
```

Chương trình gồm các chức năng sau:

- Thêm câu hỏi mới lưu vào tập tin .txt lưu vào thư mục CauHoi, tên tập tin chuỗi UUID sinh ra như sau:

```
import java.util.UUID;  
UUID uuid = UUID.randomUUID();
```

- Luyện thi trắc nghiệm tiếng Anh như sau: nhập số lượng câu hỏi muốn luyện tập → chương trình chọn ngẫu nhiên số lượng câu hỏi đó từ danh sách câu hỏi vừa đọc → hiển thị từng câu hỏi cho người dùng trả lời, sau khi chọn đáp án, thông báo kết quả trả lời cho câu hỏi đó đúng hay sai, nếu đúng thì hiển thị câu hỏi tiếp theo. Quá trình này lặp lại cho đến khi hết số câu hỏi lựa chọn hoặc trả lời sai một câu hỏi bất kỳ.

*Yêu cầu thêm: thiết kế chương trình sao cho có thể mở rộng để tăng số lượng lựa chọn cho một câu hỏi, câu hỏi có thể có nhiều phương án đúng.*