

Đề ôn tập giữa kỳ

Môn: Cấu trúc dữ liệu và giải thuật 1

Đề 05

Sinh viên tự viết lớp theo yêu cầu bên dưới và viết hàm main để kiểm tra lại.

Cho lớp **FractionPoint** để quản lý điểm có tọa độ (x, y) trên mặt phẳng Oxy với x và y ở dạng phân số được mô tả như sau:

- Thuộc tính
 - o Tử số của x : **n1** (số nguyên, kiểu int)
 - o Mẫu số của x : **d1** (số nguyên, kiểu int)
 - o Tử số của y : **n2** (số nguyên, kiểu int)
 - o Mẫu số của y : **d2** (số nguyên, kiểu int)
 - o Số lượng điểm hiện có: **quantity** (số nguyên, kiểu int, thuộc tính tĩnh - static)

Sinh viên thực hiện các yêu cầu bên dưới trong file **FractionPoint.java**.

Câu 1: Viết hàm khởi tạo cho lớp **FractionPoint**:

- Hàm khởi tạo không tham số: **FractionPoint ()** ($n1 = 0, d1 = 1, n2 = 0, d2 = 1$)
- Hàm khởi tạo đầy đủ tham số: **FractionPoint (int n1, int d1, int n2, int d2)**

Câu 2: Viết các phương thức get, set cho lớp **FractionPoint**:

- Hàm nhận vào và cài đặt tử số cho x : **setN1 (int)**
- Hàm nhận vào và cài đặt mẫu số cho x : **setD1 (int)**
- Hàm nhận vào và cài đặt tử số cho y : **setN2 (int)**
- Hàm nhận vào và cài đặt mẫu số cho y : **setD2 (int)**
- Hàm trả về tử số x : **getN1()**
- Hàm trả về mẫu số x : **getD1()**
- Hàm trả về tử số y : **getN2()**
- Hàm trả về mẫu số y : **getD2()**

Câu 3: Viết các hàm get sau:

- Hàm trả về số lượng điểm hiện có: **getQuantity()** (phương thức tĩnh - static)
- Hàm trả về mảng int gồm tử số và mẫu số của x và y : **int[] getPoint()** (ví dụ điểm là $(\frac{1}{2}, \frac{2}{3})$ thì trả về mảng là `int a[] = {1,2,2,3}`)

Câu 4:

Trong file ***FractionPoint.java***, sinh viên cài đặt các phương thức:

- a. Tính khoảng cách giữa hai điểm: **public double distance(FractionPoint f)**

Biết hàm tính khoảng cách giữa 2 điểm $A(x_A, y_A)$ và $B(x_B, y_B)$ có công thức như sau:

$$\text{Khoảng cách} = \sqrt{(x_A - x_B) * (x_A - x_B) + (y_A - y_B) * (y_A - y_B)}$$

- b. Viết hàm tìm điểm đối xứng qua gốc tọa độ O với prototype: **public FractionPoint symmetry()**, điểm đối xứng qua gốc tọa độ là điểm có giá trị x và y trái dấu với giá trị ban đầu.

Ví dụ:

Điểm A có tọa độ là $(\frac{1}{3}, \frac{2}{3})$ thì điểm đối xứng qua gốc tọa độ của điểm A là $(-\frac{1}{3}, -\frac{2}{3})$

- c. Rút gọn phân số cho 2 điểm x và y với prototype: **public void reducer()**, khi chạy hàm này thì nếu phân số ở x hay y chưa được rút gọn thì sẽ rút gọn. Phân số được rút gọn bằng cách cùng chia tử số và mẫu số cho ước chung lớn nhất của tử số và mẫu số, sinh viên tự viết hàm tìm ước chung lớn nhất. Thuật toán gợi ý tìm ước chung lớn nhất có mã giả như sau (giả sử cả a và b đều là số nguyên dương):

```
int gcd(int a, int b)
    while(a != b)
        if(a > b)
            a = a - b
        else
            b = b - a
    endif
endwhile
return a
```

Ví dụ điểm A có tọa độ $(\frac{3}{9}, \frac{2}{3})$ thì khi gọi hàm $A.reducer()$, điểm A sẽ có tọa độ $(\frac{1}{3}, \frac{2}{3})$

Câu 5:

Viết hàm so sánh hai điểm $FractionPoint$, hai điểm bằng nhau khi giá trị x và giá trị y của mỗi điểm sau khi được rút gọn thì bằng nhau tương ứng. Prototype của phương thức là:

public boolean equals(Object obj)

Ví dụ với thông tin sau:

Điểm	n1	d1	n2	d2
p1	1	2	3	5
p2	2	4	6	10

p3	2	3	3	5
p4	1	2	9	15
p5	3	5	1	2

Lời gọi phương thức **p1.equals(p2)** trả về **true**.

Lời gọi phương thức **p1.equals(p3)** trả về **false**.

Lời gọi phương thức **p1.equals(p4)** trả về **true**.

Lời gọi phương thức **p1.equals(p5)** trả về **false**.

-- HẾT --