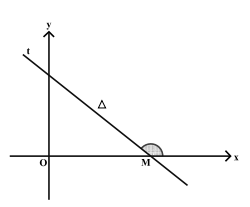
**Bài tập – 01**

Để quản lý các đường thẳng, người ta xây dựng lớp đường thẳng trong đó mỗi đường thẳng được xác định bởi phương trình: y = a.x + b



1. Hãy xây dựng lớp đường thẳng theo các bước sau đây:

* Định nghĩa các thuộc tính của lớp đường thẳng.
* Xây dựng các hàm constructors (không tham số và có tham số).
* Xây dựng hàm cơ bản Getters, Setters, toString.

1. Xây dựng các hàm cho các đối tượng của lớp đường thẳng sau đây:

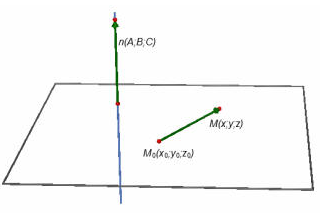
* Tính khoảng cách từ gốc tọa độ O(0, 0) đến đường thẳng.
* Kiểm tra một điểm A(xa, ya) có thuộc đường thẳng hay không.

1. Xây dựng chương trình chính cho phép:

* Tạo ra một danh sách các đường thẳng.
* Đếm xem có bao nhiêu đường thẳng đi qua điểm A(xa, ya)
* Tính tổng khoảng cách từ gốc tọa độ O(0, 0) đến các đường thẳng.
* Trong phiên bản 1.1, người dùng mong muốn thực hiện việc tính tổng khoảng cách từ một điểm A(xa, ya) đến các đường thẳng. Hãy cập nhật chương trình để thực hiện chức năng này.

**Bài tập – 02**

Để mở rộng trong không gian, người ta xây dựng các mặt phẳng trong đó mỗi mặt phẳng được xác định bởi phương trình: z = a.x + b.y + c.



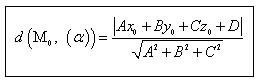
1. Hãy xây dựng lớp mặt phẳng kế thừa từ lớp đường thẳng

* Định nghĩa các thuộc tính của lớp mặt phẳng.
* Xây dựng các hàm constructors (không tham số và có tham số).
* Xây dựng hàm cơ bản Getters, Setters, toString.

1. Xây dựng các hàm cho các đối tượng của lớp mặt phẳng sau đây:

* Tính khoảng cách từ gốc tọa độ O(0, 0, 0) đến mặt phẳng.

i.e. Ax + By + Cz + D = 0 và điểm M0(x0; y0; z0)



* Kiểm tra một điểm A(xa, ya, za) có thuộc mặt phẳng hay không.

1. Sửa đổi chương trình chính cho phép:

* Tạo thêm các mặt phẳng.
* Đếm xem có bao nhiêu đường thẳng đi qua điểm A(xa, ya) hoặc mặt phẳng đi qua điểm A(xa, ya, 0)
* Tính tổng khoảng cách từ gốc tọa độ O(0, 0, 0) đến các đường thẳng, mặt phẳng.

**Bài tập – 03**

Trong phần lý thuyết, để xây dựng ứng dụng Paint, chúng ta đã xây dựng các interface nhằm trừu tượng hóa các lớp đối tượng sử dụng như Circle, Rectangle hay Triangle.

1. Trên cơ sở các interface đã có, hãy xây dựng lớp Point và lớp Line để có thể tích hợp vào ứng dụng Paint.
2. Người ta mong muốn tính khoảng cách của các hình đến một điểm Point, hãy áp dụng lý thuyết trừu tượng để xây dựng phương thức này.
3. Khi quan tâm đến các phép biến đổi hình học, các biến đổi này đều dựa vào tâm của hình để có thể thực hiện việc biến đổi. Hãy xây dựng phương thức xác định tâm của các hình.
4. Bản chất của các biến đổi như phóng to thu nhỏ hay xoay các hình thực chất là việc phóng to thu nhỏ khoảng cách từ các đỉnh đến tâm hay xoay các đỉnh đi một góc so với tâm.

Hãy cài đặt phương thức:

* Phóng to thu nhỏ: từ một đỉnh đến tâm.
* Xoay: từ một đỉnh đến tâm.

Trên cơ sở các gợi ý: chúng ta có thể hoàn thành các chức năng của bài toán với các hình hiện tại hoặc mở rộng cho các hình mới.

**Bài tập – 04**

Cho một tệp tin data.txt có dữ liệu được biểu diễn như cấu trúc sau đây:

*Bảng danh sách công việc và thời gian cần thực hiện*

*1 A 10:30 12:00*

*2 B 12:30 14:15*

*3 C 15:10 18:00*

4 D 15:35 16:00 *…*

Trong đó:

* Cột 1: định danh mã công việc
* Cột 2: định danh tên công việc
* Cột 3: thời gian bắt đầu của công việc
* Cột 4: thời gian kết thúc của công việc

Yêu cầu của bài toán:

1. Xây dựng lớp Task với các thuộc tính tương ứng.
2. Hãy thực hiện đọc dữ liệu từ tệp tin [kiểm soát các lỗi khi đọc dữ liệu] đưa vào danh sách các Task.
3. Hãy xác định khoảng thời gian trễ lớn nhất giữa 2 công việc biết rằng khoảng thời gian trễ là khoảng thời gian mà tại đó không có công việc nào được thực hiện.

Chú ý: thời gian trễ không tính các thời gian trước khi công việc sớm nhất thực hiện hay sau khi công việc kết thúc muộn nhất.