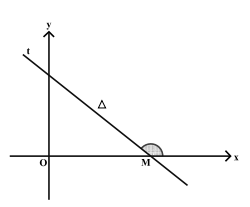
**Bài tập:**

1. 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.

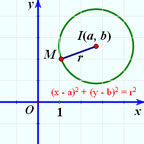
1. Xây dựng các hàm cho các đối tượng của lớp đường thẳng sau đây:

* 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.

1. Để mở rộng bài toán trên, người ta xây dựng lớp đường tròn trong đó mỗi đường tròn được xác định bởi phương trình: (x - a)2 + (y - b)2 = R2



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

* Định nghĩa các thuộc tính của lớp đường tròn.
* 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 tròn sau đây:

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

1. Thay đổi chương trình chính của bài 1 cho phép:

* Tạo ra một danh sách các đường thẳng và đường tròn
* Đếm xem có bao nhiêu hình chứa điểm A(xa, ya)
* Tính tổng khoảng cách từ gốc tọa độ O(0, 0) đến các hình.

1. Để mở rộng bài toán sau này, người ta xây dựng một ***Interface*** là Shape cho phép định nghĩa các phương thức mà các hình cần xây dựng: tính khoảng cách, kiểm tra, …

* Hãy xây dựng ***Interface*** Shape và sửa lại các lớp đường thẳng, đường tròn cho phù hợp.
* Hãy sửa lại chương trình chính dể đơn giản hóa các yêu cầu trong mục 2.c.

SODOKU:

1. NODE

* Attributes
  + row [1-9]
  + column [1-9]
  + zone [1-9]
  + value [0-9]
  + reserved: Boolean
    - true: không cho phép thay đổi.
    - false: cho phép người chơi thay đổi.
  + etc…
* Methods
  + Constructors, getters, setters
  + validate\_row(GAME, row)
  + validate\_column(GAME, column)
  + validate\_zone(GAME, zone)

1. GAME

* Attributes:
  + ArrayList [9][9]: Nodes.
* Methods

***static***

* + readData();
  + showGame();
  + validateGame();
  + Node getElement(int row, int column);

1. Chương trình chính
   1. Game g = new Game();
   2. G.readData();
   3. while(){

r = sc.nextInt();

Node node = g.getElement(r, c);

node.validate\_row …

node.validate\_column …

node.validate\_zone …

g.validateGame();

}