Cours : Programmation orientée objet en C#
Cote du cours : 26978
Session : Hiver 2022

Laboratoire 5

Les classes Point et Rectangle

La classe Point

Dans un premier temps, nous allons écrire une classe nommée **Point**. Un point est caractérisé par son abscisse (entier) et son ordonné (entier). On vous demande de :

1- Définir la classe Point, ses attributs.
2- Définir les propriétés associées aux attributs de la classe Point. Respecter les contraintes suivantes : La plage de valeur permise, pour l'abscisse et l'ordonné, est de -10 à 10. Si les valeurs saisies sont en dehors de cet intervalle, alors la valeur 0 sera affectée.
3- Définir un constructeur sans argument initialisant l'abscisse à 1 et l'ordonné à 1.
4- Définir un constructeur prenant comme argument un entier pour initialiser l'abscisse. L'ordonné sera initialisé à 1.
5- Définir un constructeur prenant comme argument deux entiers pour initialiser l'abscisse et l'ordonné.
6- Définir une méthode nommée ***ToString*** permettant de convertir un point vers une chaine de caractères. La signature de la méthode est comme suit : public String ToString(). La représentation du point est comme suit :

```
[Abscisse = 5, Ordonné = 3]
```

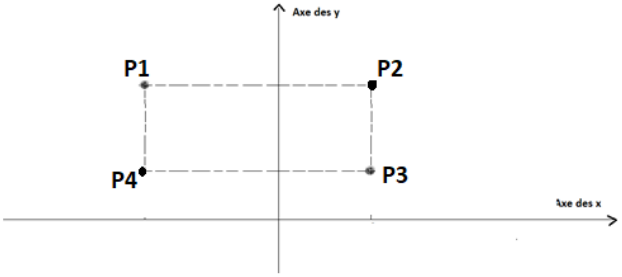
7- Définir une méthode nommée ***Translater*** permettant de translater (déplacer) le point courant d'un vecteur vx et vy. En d'autres termes, déplacer l'abscisse de vx (abscisse = abscisse +vx) et l'ordonné de vy (ordonne = ordonne + vy). La signature de la méthode est comme suit : public void Translater(int vx, int vy)
8- Définir une méthode nommée ***Distance*** permettant de calculer la distance du point courant au point passé comme paramètre. La signature de la méthode est comme suit : public double Distance (Point p)
9- Surcharger la méthode ***distance*** de la question 8.
10- Définir une méthode nommée ***Egalite*** permettant de tester l'égalité du point courant avec le point passé comme paramètre. Deux points sont égaux s'ils ont même abscisse et même ordonnée.

On donne :
La distance entre deux points A1(x1, y1) et A2(x2, y2) est définit par la formule suivante :

$$d = \sqrt{(x1 - x2)^2 + (y1 - y2)^2)}$$

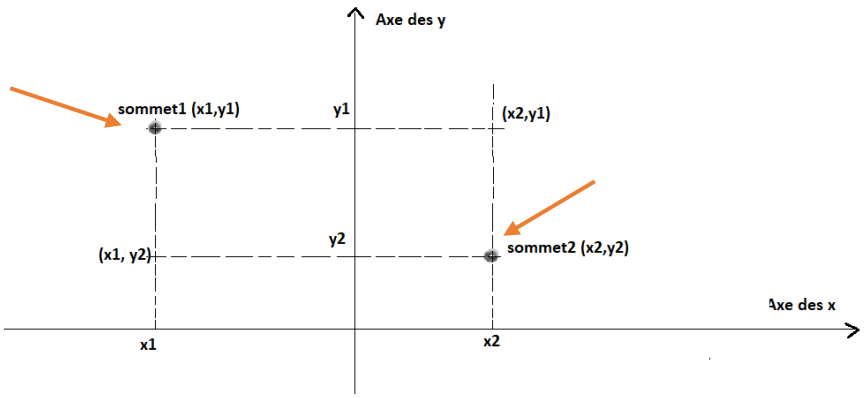
La classe Rectangle

Dans un second temps, nous allons écrire une classe nommée **Rectangle**. Dans cet exercice, on ne traite que les rectangles dont les côtés sont parallèles aux axes. Un rectangle est caractérisé par quatre points ***p1***, ***p2***, ***p3*** et ***p4*** comme présenté dans la figure ci-dessous. Cet ordre des points est important car il vous facilitera le calcul du périmètre et de la surface.



On vous demande de :

1- Définir la classe Rectangle avec ses attributs
2- Définir les propriétés associées aux attributs.
3- Définir un constructeur prenant quatre points permettant d'initialiser les quatre points du rectangle. Les points seront donnés dans **l'ordre** (p1,p2,p3 et p4).
4- Définir un constructeur prenant deux points permettant d'initialiser les quatre points du rectangle. On peut créer un rectangle en fournissant les points (Sommet1 et Sommet2) de deux sommets opposés. La figure ci-dessous montre comment trouver les deux autres points :



5- Définir une méthode nommée ***Perimetre*** permettant de calculer le périmètre du rectangle. Le périmètre est la longueur des quatre cotés du rectangle (utiliser la méthode Distance de la classe point).
6- Définir une méthode nommée ***Surface*** permettant de calculer la surface du rectangle. La surface du rectangle est la multiplication de la largeur par la longueur.
7- Définir une méthode ***ToString*** permettant de convertir un rectangle vers une chaine de caractères. La signature de la méthode est comme suit : public String ToString(). La représentation du rectangle est comme suit :

Rectangle : [Abscisse = 1, Ordonné = 3], [Abscisse = 4, Ordonné = 3], [Abscisse = 4, Ordonné = 2], [Abscisse = 1, Ordonné = 2]

8- Définir une méthode nommée ***Appartenir*** permettant de tester si un point est à l'intérieur du rectangle courant ou non. La signature de la méthode est comme suit : public boolean appartenir(Point p).

9- Définir une méthode nommée ***Appartenir*** permettant de tester si un rectangle est à l'intérieur du rectangle courant ou non. La signature de la méthode est comme suit : public boolean appartenir(Rectangle r). Pour tester si un rectangle r est contenu dans le rectangle courant, il suffit de tester l'appartenance des sommets du rectangle r au rectangle courant.

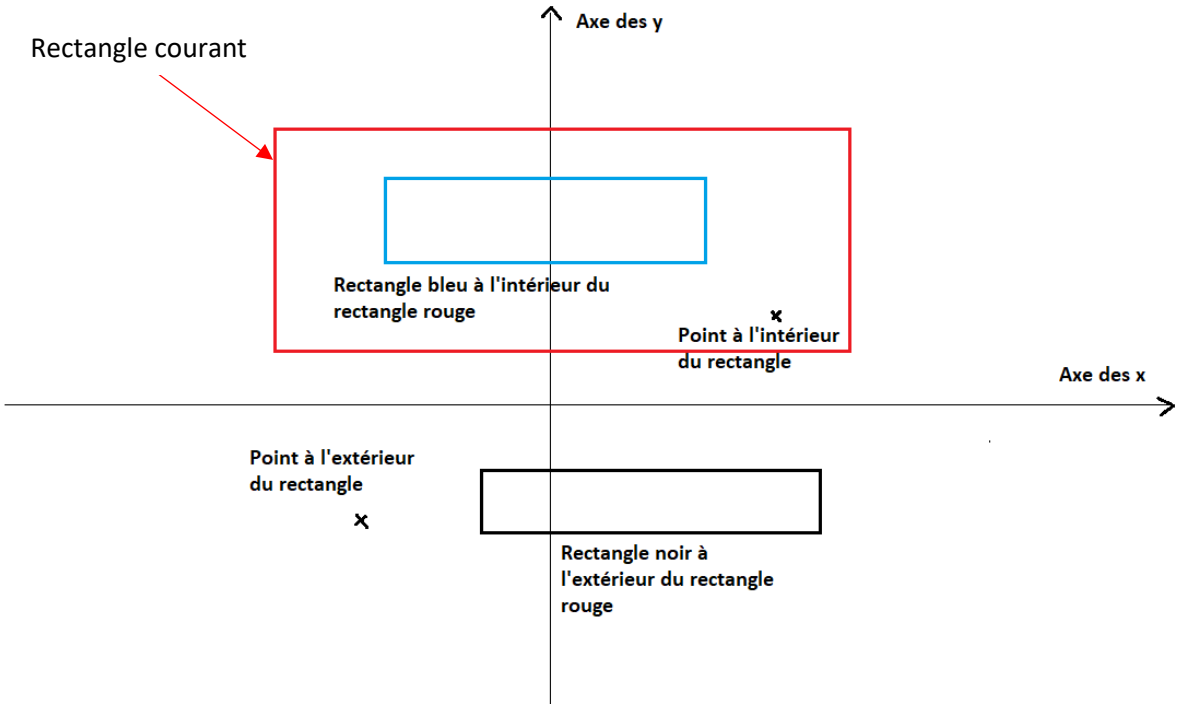


Figure illustrant l'appartenance d'un point ou d'un rectangle au rectangle rouge.

La classe Test

Enfin, on vous demande d'écrire une classe nommée **Test** pour tester les deux classes précédentes. Cette classe contient uniquement la méthode Main et doit réaliser les opérations suivantes :

1- Créer un point nommé ***p1*** avec le premier constructeur de la classe Point.
2- Créer un point nommé ***p2*** avec le second constructeur de la classe Point.
3- Créer un point nommé ***p3*** et ***p4*** avec le troisième constructeur de la classe Point.
4- Translater le point ***p1*** d'un vecteur ***V***(vx = 2, vy = 3).
5- Afficher le point ***p1***.
6- Calculer et afficher la distance entre ***p1*** et ***p3***.
7- Calculer et afficher la distance entre ***p1*** et le point de coordonnés abscisse = 6 et ordonné = 4.

8- Créer un rectangle nommé ***r1*** en fournissant deux points représentants les deux sommets opposés (par exemple : P1(-3,6) et P2(4,2)).
9- Calculer et afficher le périmètre du rectangle ***r1***.
10- Calculer et afficher la surface du rectangle ***r1***.
11- Tester si le point ***p1*** appartient au rectangle ***r1***.
12- Créer un rectangle ***r2***.
13- Tester si un rectangle ***r2*** appartient au rectangle ***r1***;
14- Afficher le rectangle ***r1***.