

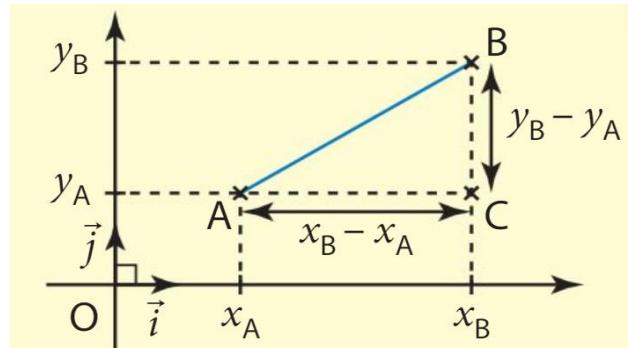
Cette évaluation est composée de 2 exercices. Les réponses sont à donner sur feuille de copie.

### 1- CLASSE POINT :

Soit les points  $A$  et  $B$  de coordonnées :  $A(x_A ; y_A)$   
et  $B(x_B ; y_B)$

La distance  $AB$  est donnée par la formule suivante :

$$AB = \sqrt{(x_B - x_A)^2 + (y_B - y_A)^2}$$



Par exemple si  $A(2 ; 1)$  et  $B(5 ; 3)$ ,  $AB = \sqrt{(5 - 2)^2 + (3 - 1)^2} = \sqrt{3^2 + 2^2} = \sqrt{13} \approx 3,606$

On a créé la classe nommée **Point**. Le constructeur de cette classe initialise les attributs d'instance nommés  $x$  et  $y$ . Cette classe comprend un attribut de classe nommé *nbPoints*. Sa valeur permet de connaître le nombre d'objets de la classe **Point** qui ont été créées. La classe comprend les méthodes suivantes :

- `__str__()` : permet d'afficher le contenu de l'objet avec un `print()`,
- `set()` : prend en argument 2 nouvelles valeurs des attributs  $x$  et  $y$  qui sont ensuite mis à jour.
- `distance()` : prend en argument une autre instance de cette classe et retourne la distance entre ces 2 points (formule ci-dessus).

En analysant avec attention les exemples d'utilisation ci-dessous, écrire **avec soin** sur feuille de copie, le code python de cette classe **Point**.

Exemples d'utilisation : En exécutant le programme principal suivant ....

#Main

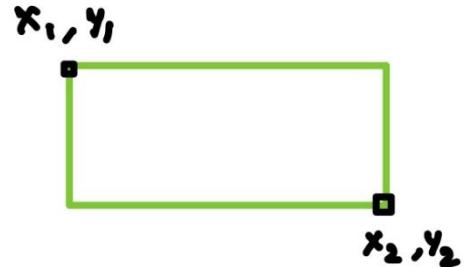
```
A = Point(2,1)
B = Point()
print(f"Nombre d'instance(s) de la classe Point : {Point.nbPoints}")
print(A)
print(B)
B.set(5,3)
print(B)
d = A.distance(B)
print(f"La distance AB est égale à {d}")
d = B.distance(A)
print(f"La distance BA est égale à {d}")
```

on obtient l'affichage suivant dans la console :

```
>>> (executing file "dsPooRectangle.py")
Nombre d'instance(s) de la classe Point : 2
Point de coordonnées (2 , 1)
Point de coordonnées (0 , 0)
Point de coordonnées (5 , 3)
La distance AB est égale à 3.605551275463989
La distance BA est égale à 3.605551275463989
```

## 2- CLASSE RECTANGLE :

On définit un rectangle en donnant les coordonnées des points haut-gauche et bas-droit. Par exemple le rectangle ci-contre est défini par les coordonnées des points  $P_1(x_1 ; y_1)$  et  $P_2(x_2 ; y_2)$ .



L'aire  $\mathcal{A}$  de ce rectangle peut être calculée par la formule suivante :

$$\mathcal{A} = (x_2 - x_1) \times (y_2 - y_1)$$

Par exemple avec  $P_1(2 ; 1)$  et  $P_2(5 ; 3)$ ,  $\mathcal{A} = (5 - 2) \times (3 - 1) = 3 \times 2 = 6$

On a créé la classe nommée **Rectangle**. Le constructeur de cette classe initialise les attributs d'instance nommées  $p1$  et  $p2$  qui ont chacun comme valeur un objet de la classe **Point** définie dans la question précédente. La classe comprend les méthodes suivantes :

- `__str__()` : permet d'afficher le contenu de l'objet avec un `print()`,
- `aire()` : retourne l'aire de ce rectangle (formule ci-dessus).

En analysant avec attention les exemples d'utilisation ci-dessous, écrire **avec soin** sur feuille de copie, le code python de cette classe **Rectangle**.

Exemples d'utilisation : En exécutant le programme principal suivant

```
#Main
P1 = Point(2,1)
r = Rectangle(P1)
print(r)
print()
P1 = Point(2,1)
P2 = Point(5,3)
r = Rectangle(P1,P2)
print(r)
a = r.aire()
print(f'L\'aire de ce rectangle est égale à {a}')
```

On obtient l'affichage suivant dans la console :

Haut-Gauche : Point de coordonnées (2 , 1)  
Bas-Droit : Point de coordonnées (0 , 0)

Haut-Gauche : Point de coordonnées (2 , 1)  
Bas-Droit : Point de coordonnées (5 , 3)  
L'aire de ce rectangle est égale à 6