



## TP 2 : Programmation C++

### Exercice 1 :

- 1-Ecrire un programme qui décrit une classe **point** avec une abscisse et une ordonnée réelles,
- 2-Ecrire une méthode pour la modification du **point** qui remplace les anciennes coordonnées par les nouvelles données comme arguments de la méthode **modPoint(...)**, utiliser pour cela le pointeur **this**.
- 3-Ecrire un constructeur pour la classe **point** qui prend comme arguments les coordonnées du point. Utiliser pour les affectations le pointeur **this**.
- 4-Ecrire un constructeur vide de la classe **point**. (sans instructions)
- 5-Ecrire un programme dans la fonction **main()** qui crée 4 points en utilisant les 4 méthodes citées dans le cours.
- 6-Utiliser la méthodes **affichePoint()** pour chaque point,
- 7-Ecrire une classe **droite** qui utilise 2 points comme attributs.
- 8-Ecrire une méthode **afficheDroite(..)** qui affiche les coordonnées d'un point en utilisant la méthode **affichePoint(..)**
- 9-Ecrire un constructeur pour la classe **droite** qui initialise une droite par deux points de coordonnées (0,0) en utilisant **modPoint()**.
- 10-Ecrire un constructeur pour la classe **droite** avec 4 arguments, qui sont les coordonnées de 2 **points**. Ce constructeur initialise la **droite** avec les coordonnées des points spécifiés.
- 11-Utilisez les 4 méthodes pour créer des droites.
- 12-Faire appel 4 **afficheDroite()** à partir des 4 objets créés.

### Exercice 2 :

Réaliser une classe **point** permettant de manipuler un point d'un plan. On prévoira :

- un constructeur recevant en arguments les coordonnées (**float**) d'un point ;
- une fonction membre **deplace** effectuant une translation définie par ses deux arguments (**float**) ;
- une fonction membre **affiche** se contentant d'afficher les coordonnées cartésiennes du point.

Les coordonnées du point seront des membres donnés privés.

On écrira séparément :

- un fichier source constituant la **déclaration** de la classe ;
- un fichier source correspondant à sa **définition**.

Écrire, par ailleurs, un petit programme d'essai (main) déclarant un point, l'affichant, le déplaçant et l'affichant à nouveau.

### Exercice 3 :

1. Définir une classe compte bancaire qui contient les attributs et les opérations suivantes :
  - Solde de type réel
  - Numéro de compte : entier
  - Débiter : opération qui débite le compte d'un montant donné en paramètre. Cette opération doit empêcher tout débit quand le compte n'est pas suffisamment approvisionné.
  - Créditer : opération qui crédite le compte d'un montant passé en entrée
  - VirVerser : opération qui verse un montant dans un compte de destination. L'opération prend en argument le montant à verser et le compte de destination.
  - getSolde : permet de retourner le solde
  - getNumero : permet de retourner le numéro de compte
  - Constructeur qui prend deux arguments pour initialiser le numéro et le solde
  - Afficher : opération permet d'afficher le numéro de compte et le solde.
2. Réaliser un programme principal qui teste toutes les opérations précédentes.

### Exercice 4 :

Écrire un programme utilisant une classe rectangle dont le constructeur prend deux paramètres, largeur et hauteur et qui offre les fonctions suivantes :

- calcul du périmètre
- calcul de la surface
- affichage

Ainsi que les getters et les setters (lecture et modification de la largeur et de la hauteur).

Rectangle - Menu

- 1 - Modifier largeur du rectangle
- 2 - Modifier hauteur du rectangle
- 3 - Calculer les propriétés du rectangle
- 4 - Afficher le rectangle
- 5 - Quitter