

Classe et objet

Comme on l'a dit en cours, une classe c'est comme une structure mais avec des fonctions membres. Dans ce cadre toutes les fonctions deviennent fonctions membres.

Considérons la structure point :

```
struct Point
{
    double x ;
    double y ;
};
```

Une fonction affichage serait :

```
void afficher(Point & P).
```

C'est-à-dire que la fonction d'affichage prend comme argument un point.

Si on déclare un point comme une classe, la fonction d'affichage ferait partie de la classe :

```
classe Point
{
    double x ;
    double y ;

    public :
        void afficher() ;
};
```

Vous remarquez que dans ce deuxième cas, la fonction d'affichage ne prend aucun argument. En fait il y en a un, mais il est implicite (l'objet point qui appelle la fonction) :

```
Point P ; // création d'un point P
....
P.afficher() ; // l'objet P appelle la fonction afficher
```

Exo1 : Écrire la classe point avec une fonction d'affichage. Écrire un main dans lequel vous créez un point et vous l'affichez

Exo2 : Écrire dans la classe une fonction de saisie. Écrire un main dans lequel vous créez un point, vous le saisissez et ensuite vous l'affichez

Exo3 :

a- Écrire le constructeur qui prend en argument deux réels. Ce constructeur nous permettra de faire :

```
Point P(5.2 , 3.8);
```

b- Écrire le constructeur sans arguments. Ce constructeur nous permettra de faire :

Point P;

Rappel : le constructeur est une fonction membre ayant le même nom que la classe et qui n'a pas de retour. Son objectif est de créer un objet cohérent apte à exister et à être utilisé.

Exo4 : Ecrire la fonction qui retourne la distance entre deux points :

double Point::distance(Point & P) ;

Pourquoi la fonction distance qui doit avoir 2 arguments n'en fait apparaître que 1 seul

Exo5 : Ecrire une fonction qui affiche un tableau de Points (cette fonction ne fait pas partie de la classe Point) : vous êtes obligés d'utiliser la fonction d'affichage de la classe Point.

Exo6 : Ecrire une fonction qui saisie un point :

void Point::saisir() ;

Exo7 : Ecrire la fonction saisir qui permet de saisir un tableau de points

*void saisir (Point *T, int N)*

Exo8 : Ecrire la fonction saisir_points ayant le prototype suivant :

*Point *saisir_points(int &n) ;*

qui alloue dynamiquement un tableau de n point. n est demandée à l'utilisateur et renvoyée par variable à la fonction appelante :

Exemple :

```
int main()
{
    int N ;
    Point *T=saisir_points(N) ;
    afficher(T,N) ;
}
```

Exo9 : Rajouter une instruction au main() précédent pour libérer la mémoire allouée pour le tableau de points après la fonction d'affichage et avant la fin du programme.

Exo10 : Définir la classe Cercle. Proposer un ou plusieurs constructeurs. Écrire la fonction d'affichage. Écrire une fonction de saisie.