

POO : Notion de Classe, Objet, Constructeur et Destructeur

Notion d'une classe :

Une classe est la généralisation de la notion de type défini par l'utilisateur, dans lequel se trouvent associées à la fois des données (on parle de « membres donnée ») et des méthodes (on parle de « fonctions membre » ou de fonctions).

- ⇒ Une classe = famille d'objets ayant même structure et même comportement.
- ⇒ Une classe regroupe un ensemble d'attributs ou membres, répartis en données et méthodes.

En P.O.O, les données sont « encapsulées », ce qui signifie que leur accès ne peut se faire que par le biais des méthodes. C++ vous autorise à n'encapsuler qu'une partie seulement des données d'une classe.

- ⇒ Les membres peuvent être : public, protected, private (défaut).

Déclaration et définition d'une classe :

La déclaration d'une classe précise quels sont les membres (données ou méthodes) publics (c'est-à-dire accessibles à l'utilisateur de la classe) et quels sont les membres privés (inaccessibles à l'utilisateur de la classe). On utilise pour cela les mots-clés **public** et **private**, comme dans cet exemple dans lequel la classe *point* comporte deux membres donnée privés *x* et *y* et trois fonctions membres publiques *initialise*, *deplace* et *affiche* :

```
/* -----Déclaration de la classe point-----*/
class point
{
    /* déclaration des membres privés */
private:
    int x;
    int y;
    /* déclaration des membres publics*/
public:
    void initialise(int, int);
    void deplace(int, int);
    void affiche();
};
```

La définition d'une classe consiste à fournir les définitions des fonctions membre. On indique alors le nom de la classe correspondante, à l'aide de l'opérateur de résolution de portée (::) . Au sein de la définition même, les membres (privés ou publics — données ou méthodes) sont directement accessibles sans qu'il soit nécessaire de préciser le nom de la classe. Voici, par exemple, ce que pourrait être la définition de la fonction *initialise* de la classe précédente :

```
void point::initialise(int abs, int ord)
{
    x = abs;
    y = ord;
}
```

Ici, *x* et *y* représentent implicitement les membres *x* et *y* d'un objet de la classe `point`.

Les mots-clés `public` et `private` peuvent apparaître à plusieurs reprises dans la déclaration d'une même classe. Une déclaration telle que la suivante est tout à fait envisageable :

```
class MaClasse
{
    private:
    ...
    public:
    ...
    private:
    ...
    public:
    ...
};
```

Utilisation d'une classe :

Utilisation d'une classe On déclare un « **objet** » d'un type classe donné en faisant précéder son nom de celui de la classe, comme dans l'instruction suivante qui déclare deux objets *a* et *b* de type `point` : *point a, b* ; On peut accéder à n'importe quel membre public (donnée ou fonction) d'une classe en utilisant l'opérateur `"."`. Par exemple : *a.initialise (5, 2)* ; appelle la fonction membre initialise de la classe à laquelle appartient l'objet *a*, c'est-à-dire, ici, la classe **point**.

Constructeur et destructeur :

Une fonction membre portant le même nom que sa classe se nomme un constructeur. Dès qu'une classe comporte un constructeur (au moins un), il n'est plus possible de déclarer un objet du type correspondant, sans fournir des valeurs pour les arguments requis par ce constructeur (sauf si ce dernier ne possède aucun argument).

Le constructeur est appelé après l'allocation de l'espace mémoire destiné à l'objet. Par définition, un constructeur ne renvoie pas de valeur (aucune indication de type, pas même void, ne doit figurer devant sa déclaration ou sa définition).

Une fonction membre portant le même nom que sa classe, précédé du symbole **tilde** (`~`), se nomme un destructeur. Le destructeur est appelé avant la libération de l'espace mémoire associé à l'objet. Par définition, un destructeur ne peut pas comporter d'arguments et il ne renvoie pas de valeur (aucune indication de type ne doit être prévue).

Exploitation d'une classe :

En pratique, à l'utilisateur d'une classe (on dit souvent le « client »), on fournira :

- ⇒ Un fichier en-tête « **.h** » contenant la déclaration de la classe : l'utilisateur l'employant devra l'inclure dans tout programme faisant appel à la classe en question
- ⇒ Le fichier source « **.cpp** » contient la définition des méthodes de la classe déclarée dans le fichier d'en-tête. Il inclut le fichier d'en-tête pour s'assurer que la déclaration de la classe est connue pendant la compilation de ce fichier en utilisant

```
#include "Nom_fichier.h"
```
- ⇒ Un fichier main « **.cpp** » utilise la classe dans la fonction **main()**. Il inclut également le fichier d'en-tête.