

Notion d'objet :

C'est un bloc de code mêlant des variables et des fonctions, appelées respectivement attributs et méthodes. Les attributs définissent les caractéristiques d'un objet d'une classe, les méthodes définissent quant à elles les fonctions propres aux instances d'une classe.

Notion de classe :

Une classe va permettre de regrouper en une seule entité des données membres et des fonctions membres appelées méthodes. Cependant, contrairement au langage C, les structures du C++ permettent cela également

Constructeur/Destructeur :

Constructeur:

Le constructeur est la fonction membre appelée automatiquement lors de la création d'un objet. Un constructeur permet d'initialiser des variables.
Il porte le même nom que la Classe et ne possède AUCUNE valeur de retour.
Lorsqu'on crée un objet, le constructeur est appelé aussi.

Exemple de constructeur:

```
NomClasse::NomClasse(int parametre1, char parametre2, float parametre3){  
  
    this->parametre1 = parametre1;  
  
    this->parametre2 = parametre2;  
  
    this->parametre3 = parametre3;  
  
}
```

Destructeur :

le destructeur intervient lors de la destruction d'un objet

Notion de fonction d'accès :

Elle permet de faire des calculs dans les requêtes, d'utiliser des formules conditionnelles dans les états, de déterminer le nombre d'enregistrements sélectionnés. Détermine le nombre d'enregistrements contenus dans un jeu d'enregistrements (domaine) spécifié.

Pointeur :

Un pointeur est un objet qui contient l'adresse mémoire d'une donnée ou d'une fonction.

Méthode :

Une méthode permet de définir une suite d'opérations à exécuter dans un ordre séquentiel donné.

En fonction de l'entête de la méthode on peut définir si elle renvoie une valeur ou non par exemple void qui ne renvoi aucune valeur

On définit également dans l'entête les paramètres (int, string, float,...).

Notion d'héritage :

C'est un principe propre à la programmation orientée objet, permettant de créer une nouvelle classe à partir d'une classe existante. Cela a comme avantage majeur de ne pas avoir à repartir de zéro lorsque l'on veut spécialiser une classe existante.

Classe abstraite :

En C++, une classe avec une fonction virtuelle pure est appelée une classe abstraite. Exemple d'une fonction virtuelle : *virtual void myFunction() = 0;*

Notion de liste :

C'est un objet algorithmique servant à stocker plusieurs variables au cours d'un programme. Les variables stockées peuvent être de différents types : variables numériques, chacun des caractères, booléens listes, etc. Les différents types de variables peuvent être mélangés au sein d'une même liste.

Notion d'arbre :

Un arbre est un ensemble de nœuds reliés entre eux par des "arêtes" dirigées ou non dirigées. L'un des nœuds est désigné comme "nœud racine" et les autres nœuds sont appelés nœuds enfants ou nœuds feuilles du nœud racine. En général, chaque nœud peut avoir autant d'enfants mais un seul nœud parent.

Structure : la pile

C'est une structure de données qui ne permet que deux opérations : empiler un élément, qui consiste à ajouter un élément en haut de la pile, dépiler un élément, qui consiste à retirer le dernier élément empilé et à lire son contenu.

Notion de file :

C'est un type abstrait basé sur le principe "premier entré, premier sorti" ou PEPS, désigné en anglais par l'acronyme FIFO ("first in, first out") : les premiers éléments ajoutés à la file seront les premiers à en être retirés.

Ce qui a été traité :

- ☒ objet
- ☒ -classe
- ☒ -constructeur / destructeur
- ☒ -fonctions d'accès
- ☒ -méthodes
- ☒ -héritage
- ☒ -classe abstraite

-
- ☒ -pointeurs
 - ☒ Listes
 - ☒ Piles
 - ☒ Files
 - ☒ Arbres