Cours Orienté Objet C++

La méthode orientée objet permet de concevoir une application sous la forme d'un ensemble d'objets reliés entre eux par des relations.
Lorsque que l'on programme avec cette méthode, la première question que l'on se pose plus souvent est :
«qu'est-ce que je manipule ? »,
Au lieu de
« qu'est-ce que je fait ? ».

L'une des caractéristiques de cette méthode permet de concevoir de nouveaux objets à partir d'objets existants.
On peut donc réutiliser les objets dans plusieurs applications.
La réutilisation du code fut un argument déterminant pour vanter les avantages des langages à objets.

Pour faire la programmation orientée objet il faut maitriser les fondamentaux de l'orienté objet à savoir:

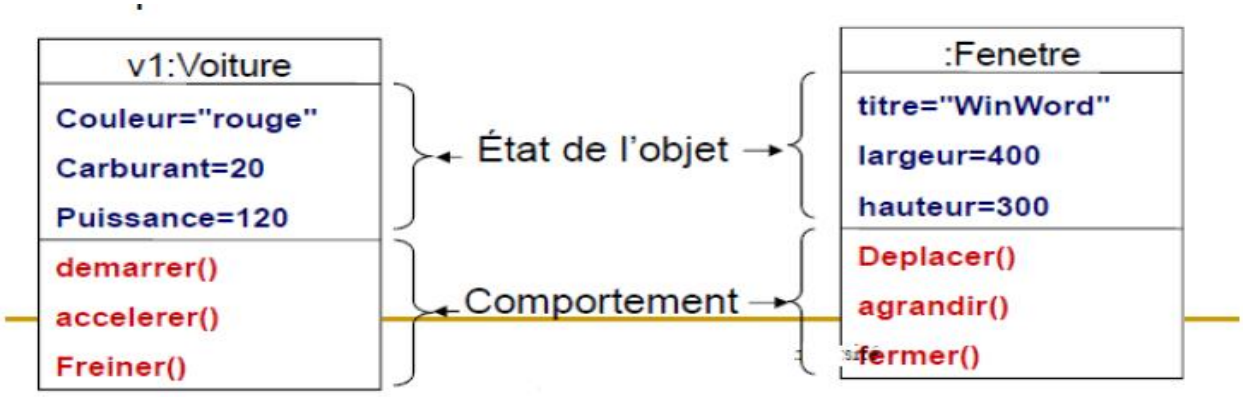
Objet et classe
Héritage
Encapsulation (Accessibilité)
Polymorphisme

Objet

Un objet est une structure informatique définie par un état et un comportement.
Objet=état + comportement

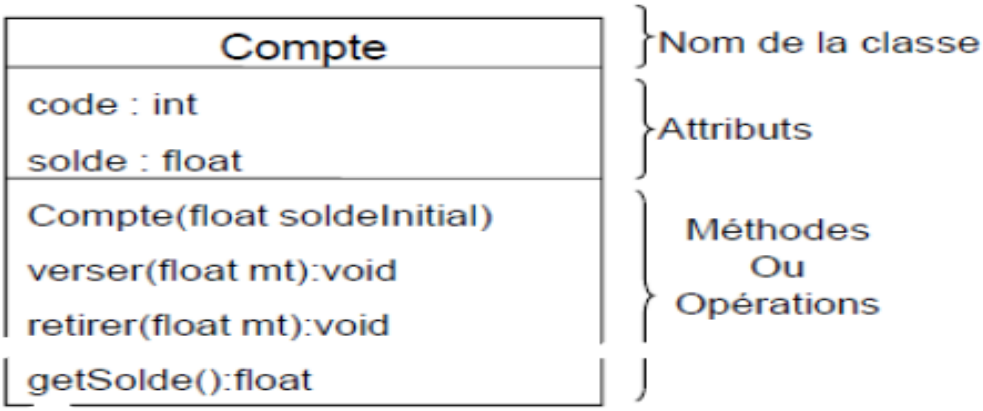
L'état regroupe les valeurs instantanées de tous les attributs de l'objet.
Le comportement regroupe toutes les compétences et décrit les actions et les réactions de l'objet. Autrement dit le comportement est défini par les opérations que l'objet peut effectuer.
L'état d'un objet peut changer dans le temps.
Généralement, c'est le comportement qui modifie l'état de l'objet
Exemple

Cours Orienté Objet C++



Classes

Les objets qui ont des caractéristiques communes sont regroupés dans une entité appelée classe. La classe décrit le domaine de définition d'un ensemble d'objets.
Chaque objet appartient à une classe
Les généralités sont contenues dans les classes et les particularités dans les objets.
Les objets informatiques sont construits à partir de leur classe par un processus qui s'appelle l'instanciation.
Tout objet est une instance d'une classe.

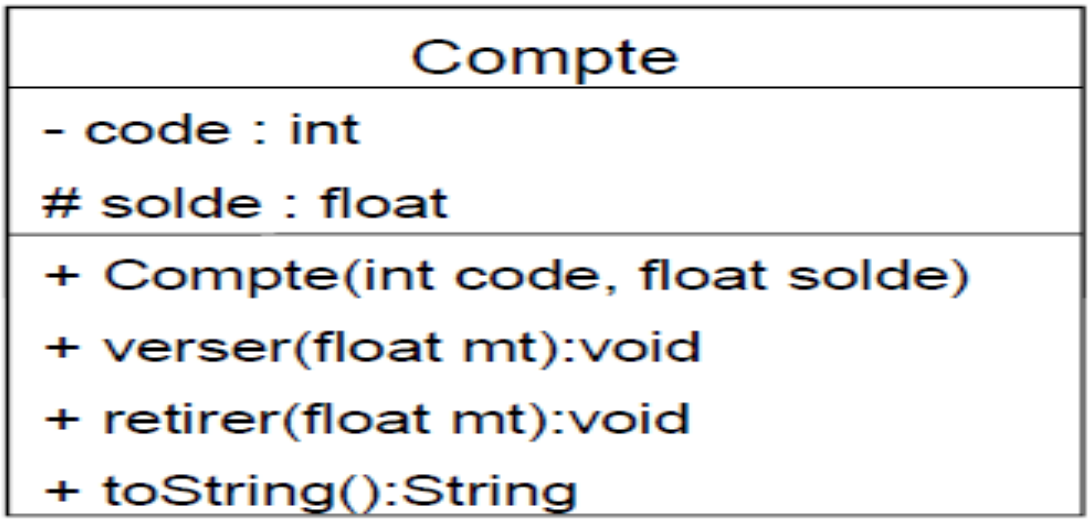


Exemple

Caractéristique d'une classe

Une classe est définit par:
Les attributs
Les méthodes
Les attributs permettent de décrire l'état de des objets de cette classe.
Chaque attribut est défini par:
Son nom
Son type
Éventuellement sa valeur initiale
Les méthodes permettent de décrire le comportement des objets de cette classe.

Une méthode représente une procédure ou une fonction qui permet d'exécuter un certain nombre d'instructions.



Parmi les méthodes d'une classe, existe deux méthodes particulières:
Une méthode qui est appelée au moment de la création d'un objet de cette classe. Cette méthode est appelée **CONSTRUCTEUR**
Une méthode qui est appelée au moment de la destruction d'un objet. Cette méthode s'appelle le **DESTRUCTEUR**

Compte();

virtual ~Compte();

Une classe est représentée par un rectangle à 3 compartiments:
Un compartiment qui contient le nom de la classe
Un compartiment qui contient la déclaration des attributs
Un compartiment qui contient les méthodes

Accessibilité aux membres d'une classe

Dans java, il existe 4 **niveaux** de **protection** :
private (-) : Un membre privé d'une classe n'est accessible qu'à l'intérieur de cette classe.

protected (#) : un membre protégé d'une classe est accessible à :
L'intérieur de cette classe
Aux classes dérivées de cette classe.
Aux classes du même package.

public (+) : accès à partir de toute entité interne ou externe à la classe
Autorisation par défaut : dans java, en l'absence des trois autorisations précédentes, l'autorisation par défaut est **package.** Cette autorisation indique que uniquement les classes du même package ont l'autorisation d'accès.

static : Un membre static d'une classe appartient à cette classe (Pas à l'objet).

Dans l'exemple de la classe Compte, chaque objet Compte possède ses propres variables code et solde. Les variables code et solde sont appelées variables d'instances.
Les objets d'une même classe peuvent partager des mêmes variables qui sont stockées au niveau de la classe. Ce genre de variables, s'appellent les variables statiques ou variables de classes.
Un attribut statique d'une classe est un attribut qui appartient à la classe et partagé par tous les objets de cette classe.
Comme un attribut une méthode peut être déclarée statique, ce qui signifie qu'elle appartient à la classe et partagée par toutes les instances de cette classe.
Dans la notation UML, les membres statiques d'une classe sont soulignés.

D'implémentation d'une classe

```
class Compte{
public:
     int code;

     public string toString(){
          return "Code="+code+"Solde"+solde;
     }

      public string verser(float mt){
           solde-=mt;
     }

      public string retirer(float mt){
          solde-=mt;
     }


private:
     float solde;

/*protected:*/

};
```

TP d'implémentation …… Constructeur …. Desctructeur ..

Getters et Setters

Les attributs privés d'une classe ne sont accessibles qu'à l'intérieur de la classe.

Pour donner la possibilité à d'autres classes d'accéder aux membres privés, il faut définir dans la classes des méthodes publiques qui permettent de :
 Lire les variables privées. Ce genre de méthodes s'appelle les **accesseurs** ou **Getters**
 Modifier les variables privés. Ce genre de méthodes s'appelle les **mutateurs** ou **Setters**

Les getters sont des méthodes qui commencent toujours par le mot **get** et finissent par le nom de l'attribut en écrivant en majuscule la lettre qui vient juste après le get. Les getters retournent toujours le même type que l'attribut correspondant.

Les setters sont des méthodes qui commencent toujours par le mot set et finissent par le nom de l'attribut en écrivant en majuscule la lettre qui vient juste après le set.

```
class Compte{
public:
     int code;

     public string toString(){
          return "Code="+code+"Solde"+solde;
     }

      public string verser(float mt){
           solde-=mt;
     }

      public string retirer(float mt){
          solde-=mt;
     }
     //Getters et Setters
     void setSolde(float num);
     float getSolde();


private:
     float solde;

/*protected:*/


};
```

Exercice 1 :

Soit un service caractérisé par son code et son libelle.
Soit un employé caractérisé par leur Matricule, Prénom, Nom, Sexe, Nombre d'enfants, Ancienneté, Salaire de base, Statut (Cadre, Maîtrise, Agent d'exécution), Prime spéciale, Prime, IPRES et son service.

- Une Indemnité représentant 5% du salaire de base est accordée à chaque employé.
- Salaire imposable = Salaire de base + Indemnités + Prime spéciale
- Caisse de sécurité sociale est égale 3% du salaire imposable
- Pour le calcul de l'impôt sur le revenu, les règles sont les suivantes :
  - Si l'ancienneté > 15 et Nombre d'enfants >=3 alors 5% du salaire imposable
  - Sinon 8% du salaire imposable
- Tout agent verse pour le régime général IPRES 8,4% du salaire imposable et 3,6% du salaire imposable pour le régime cadre s'il s'agit d'un cadre
- Retenues = Impôt + IPRES+CSS
- Net à percevoir = Salaire imposable-Retenues

Ecrire un programme qui contient les modules suivants () :

1. Créer le Diagramme de Classe

2. Créer une classe Employé

3. Créer une classe Service

4. Créer une Application Test qui permet d'ajouter un Service et un Employé puis de les Affiches ;

Exercice 2

Une cercle est défini par :
Un point qui représente son centre : centre(x,y) et un rayon.
On peut créer un cercle de deux manières :
Soit en précisant son centre et un point du cercle.
Soit en précisant son centre et son rayon
Les opérations que l'on souhaite exécuter sur un cercle sont :
getPerimetre() : retourne le périmètre du cercle getSurface() : retourne la surface du cercle.
appartient(Point p) : retourne si le point p appartient ou non à l'intérieur du cercle.
toString() : retourne une chaîne de caractères de type CERCLE(x,y,R)

1. Etablir le diagramme de classes
2. Créer les classe Point définie par:

Les attributs x et y de type int
Un constructeur qui initialise les valeurs de x et y.
Une méthode toString().

3. Créer la classe Cercle

4. Créer une application qui permet de :
   a. Créer un cercle défini par le centre c(100,100) et un point p(200,200)
   b. Créer un cercle défini par le centre c(130,100) et de rayon r=40
   c. Afficher le périmètre et le rayon des deux cercles.
   d. Afficher si le point p(120,100) appartient à l'intersection des deux cercles ou non.