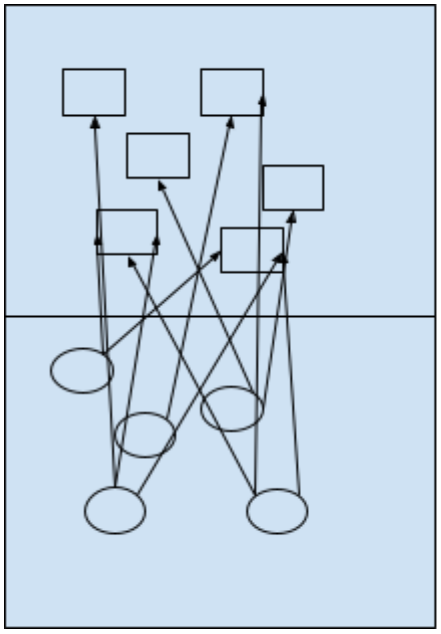
La Programmation Orientée Objet
Java et PHP

1. **Introduction**

Dans la programmation structurée, un algorithme (ou un programme) se présentait comme un ensemble d'objets de données sur lesquels s'exécutent des actions (procédures ou des fonctions). L'algorithme se divisait en deux parties : une partie déclaration et une partie traitement, les deux parties sont séparées.

La programmation structurée présente des inconvénients :

- le manque de lisibilité et de visibilité des codes
- la difficulté de la réutilisabilité des codes
- la difficulté de la maintenance des codes



La programmation orientée objet tente de modéliser les objets du monde réel tels qu'ils sont perçus sous forme d'un état et d'un comportement.

Etat : Ensemble d'attributs, de propriétés, de caractéristiques qui définissent intrinsèquement un objet du monde réel.

Comportement : Ensemble de méthodes, d'actions, de procédures et de fonctions qui agissent sur l'état et le font évoluer.

Exemples :

Humain

Etat : nom, prénom, âge, taille, date de naissance, etc

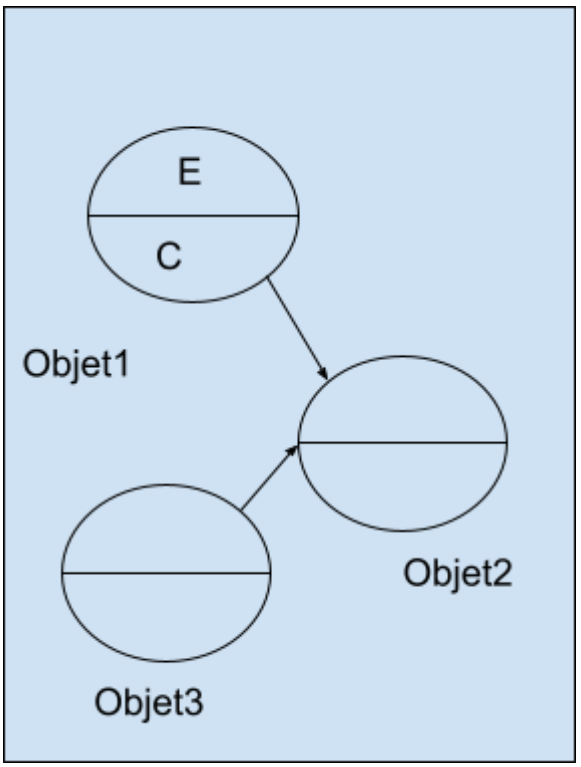
Comportement : Naitre, Manger, grandir, voyager, etc … mourir (disparaître)

Compte bancaire

Etat : nom, prénom, solde, type de compte etc

Comportement : ouvrir, déposer, retirer, etc … fermer

Un algorithme (ou un programme) orienté objet n'est autre qu'un ensemble d'objet qui peuvent communiquer entre eux.



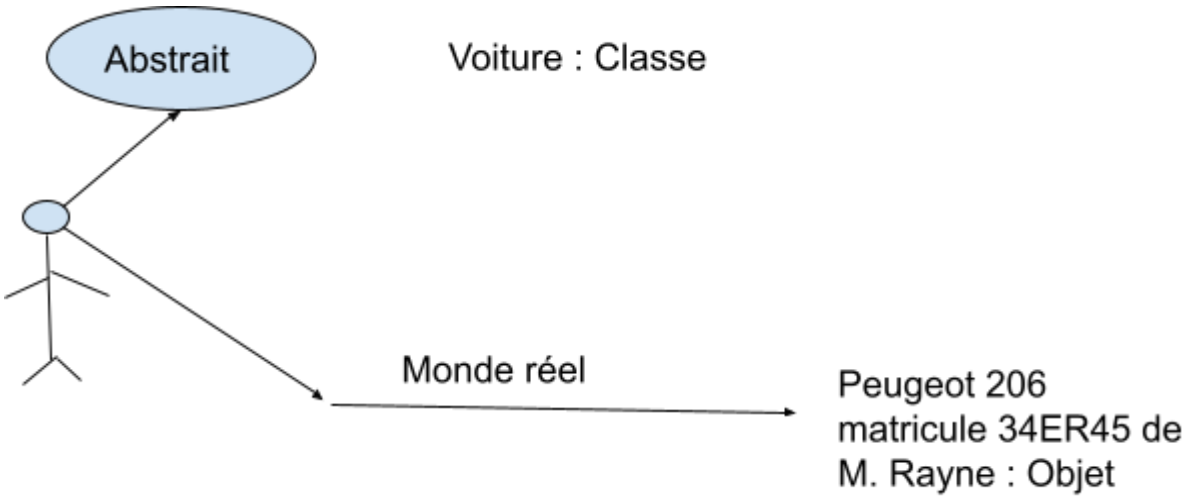
Programme Orienté Objet

2. Les concepts de base

a. La classe et l'objet

Une classe est un modèle, une catégorie, un supertype qui possède un nom et est composé d'un état et un comportement.

Un objet est une instance de la classe, une sorte de la classe, une déclinaison d'un objet du monde réel de la classe.



Syntaxe de déclaration d'une classe en Java et PHP :

```
class NomClasse {
    //attributs
    //méthodes
}
```

Principe d'instanciation de la classe : création d'un objet de la classe dans le monde réel.

Java	PHP
NomClasse nomObjet = new NomClasse ();	$nomObjet = new NomClasse () ;

L'accès aux attributs se fait en Java avec le point de composition (.) et en PHP avec (->).

Java	PHP
nomObjet.attribut nomObjet.methode ()	$nomObjet->attribut $nomObjet->methode ()

b. La visibilité

Les membres de la classe (attribut ou méthode) peut être accessibles selon trois types d'accès :

- privé ou private : tout membre déclaré en privé est visible exclusivement dans sa classe.
- public ou public : tout membre déclaré en public est visible dans sa classe et à l'extérieur de celle-ci.
- protégé ou protected : tout membre déclaré en protégé est visible dans sa classe et dans toutes les classes héritières (principe d'héritage).

Principe d'encapsulation : En règle générale, tous les attributs sont déclarés en privé sauf exception, toutes les méthodes sont déclarées en public sauf exception.

Comment accéder aux attributs privés d'une classe à partir d'une autre classe ?

La réponse ; on crée pour chaque attribut deux méthodes en public qui sont : get et set. Get pour récupérer la valeur de l'attribut et set pour modifier la valeur attribut : getter and setter.

Exemple en Java :

```
class Date {
    private int jour, mois, annee ;
    public void saisir () {
        …
    }
    public void afficher () {
    …
```

```
        }
        // getter sur le jour
        public int getJour () {
                return jour ;
        }
        //setter sur le jour
        public void setJour (int j ) {
                jour  = j;
        }
    }
```

Exemple en PHP :

```
    class Date {
        private $jour, $mois, $annee ;
        public function saisir () {
                …
        }
        public function afficher () {
        …
        }
        // getter sur le jour
        public function getJour () {
                return $jour ;
        }
        //setter sur le jour
        public function setJour ( $j ) {
                $jour  = $j;
        }
    }
```

c. **La référence this**

Chaque objet incorpore en son sein une référence (un pointeur) nommé par défaut this. Cette référence symbolise l'auto-référencement (le moi).

Syntaxe d'accès avec this :

Java	PHP
this.attribut this.méthode ()	$this->attribut $this->méthode ()

Exemple en Java :

```
class Date {
        private int jour, mois, annee ;
        public void saisir () {
                …
        }
        public void afficher () {
        …
```

```
        }
        // getter sur le jour
        public int getJour () {
                return jour ;
        }
        //setter sur le jour
        public void setJour (int jour ) {
                jour  = jour; //ambiguité de nom : entre attribut et argument.
                this.jour = jour ;
        }
}
```

La référence this lève l'ambiguïté entre les noms des attributs et les arguments ou les variables locales des méthodes.

d. Les constructeurs et les destructeurs

Les constructeurs sont des méthodes particulières qui s'exécutent automatiquement dès l'instanciation d'une classe. Ils permettent d'initialiser les attributs de la classe.

En Java, ils ont le même nom que la classe. En PHP, ils sont nommés __construct ().

Exemples :

Java	PHP

class **Date** {
private int jour, mois, annee ;
public **Date** ( ) {
this.jour = 0 ;
this.mois = 0 ;
this.annee = 0 ;
}
}
	class **Date** {
	private $jour, $mois, $annee ;
	public function **__construct** ( ) {
	$this->jour = 0 ;
	$this->mois = 0 ;
	$this->annee = 0 ;
	}
	}

Les constructeurs en Java comme en PHP peuvent recevoir des arguments.

Il n'y a pas de destructeurs en Java, car java dispose d'un programme ramasse-miettes qui permet de récupérer tous les espaces mémoires non utilisés.

En PHP, les destructeurs sont nommés __destroy( ) : leur fonction principale est la récupération des espaces mémoires non utilisés (unset).

_____ FIN DU COURS 16/09/2023 _____