UML

Les niveaux

^

Haut

niveau

I python

I Web; js

I C

I L’assembleur qui compacte le binaire sous des paquets de 8 (1octet)

Bas

niveau

I Le binaire

UML : Unified Model Langage. Un langage pour les unifier tous

**1 - Programmation orientée objets & les classes**

La programmation orientée objet (POO), ou programmation par objet, est un [paradigme](https://fr.wikipedia.org/wiki/Paradigme_(programmation)" \o "Paradigme (programmation)) de [programmation informatique](https://fr.wikipedia.org/wiki/Programmation_informatique" \o "Programmation informatique). Il consiste en la définition et l'interaction de briques logicielles appelées *[objets](https://fr.wikipedia.org/wiki/Objet_(informatique)" \o "Objet (informatique))* ; un objet représente un concept, une idée ou toute entité du monde physique, comme une voiture, une personne ou encore une page d'un livre. Il possède une structure interne et un comportement, et il sait interagir avec ses pairs. Il s'agit donc de représenter ces objets et leurs relations ; l'interaction entre les objets via leurs relations permet de concevoir et réaliser les fonctionnalités attendues, de mieux résoudre le ou les problèmes.

**1.3 UML**

On écrit

**1.4 Syntaxe**

En python on utilise le mot clé Classe pour définir une classe

Le nom de la classe commence par une majuscule

Une classe est considérée comme un bloc et on se termine par «:», le corps de la classe est indenté

Class MaClasse :

Le corps de ma classe

On peut déclarer cet objet afin de l’utiliser

MonObjet = MaClasse()

On l’affiche

print(MonObjet)

**1.4.1 Les méthodes**

Les méthodes sont des fonctions déclarées à l’intérieur de la classe. Méthode est juste un nom pour dire «cette fonction est dans une classe». On va maintenant créer une méthode très simple

**1.4.5 Protection des données**

**1.4.5.1 Principe de l’encapsulation**

Masquer au maximum les champs de la classe pour restreindre leurs accès

**1.4.5.2 Objectif**

Protéget contrôler les valeurs des attributs

Nom UML Python Accesibilité

Public + Aucun (par défaut)

Protégé # \_ avant le nom Classe dérivées

Privé - \_\_ avant le nom Seules les méthodes situés dans la class

Getters :

Offre un accès de lecture

Aucun paramètre d’entrée (autre que self)

Setters :

Un paramètre d’entrée(en plus de self)

Composition : aucun des deux objets ne peut exister sans l’autre (fondation/bâtiment)

Aggrégation : un des deux objets ne peut pas exister sans l’autre(orchestre/musicien)

Association : les deux objets peuvent exister sans l’autre(voiture/personne)