

# Concepts et langages orientés objets – Définitions

---

## Les définitions du cours

**Définition** *Modélisation*

La modélisation est une simplification d'un problème (de la réalité) qui permet de résoudre un problème.

**Définition** *Modèle*

Un modèle est une représentation simplifiée pertinente de la réalité pour résoudre un problème.

**Définition** *L'objet*

Un objet est une entité informatique regroupant les données et les traitements d'une entité des problèmes.

**Définition** *L'encapsulation*

Les données de l'objet sont encapsulées dans une « couche de procédure ». Nous sommes donc obligés de passer par les procédures de l'objet pour accéder aux données de celui-ci.

**Définition** *L'interface*

L'interface est l'ensemble de méthodes que l'on peut invoquer par l'envoi de messages.

**Définition** *Le protocole*

Le protocole est l'ensemble des signatures de méthodes de l'objet avec leur sémantique en français.

**Définition** *La signature*

Une signature est composée d'un nom, d'un type ainsi que de la position de ses arguments.  
Par exemple : `print(int), print(int, double), print(double, int)`.

**Définition** *La classe*

Déclaration de structures de données et de méthodes d'objets identiques.

**Définition** *L'instanciation*

La création d'un objet à partir d'une classe s'appelle l'instanciation.

La seule différence entre deux instances de la classe est le contenu de leurs variables (attributs).

**Définition** *Information Hidding*

Cela caractérise le fait qu'un utilisateur d'objet connaisse son protocole mais pas son implémentation qui lui est cachée.

**Définition** *Instanciation d'une classe*

C'est l'opération qui consiste à créer un nouvel objet à partir (qui satisfait les déclarations de la classe) des déclarations de la classe.

**Définition** *Constructeur*

Un constructeur est un élément du langage Java qui permet de créer des instances et de les initialiser.

**Définition** *Le polymorphisme*

Capacité d'un langage (objet) à permettre de multiples définitions d'une méthode de même signature dans des objets différents (une seule définition par objet).

**Définition** *Le « binding »*

Lien entre une référence symbolique et le code exécutable.

**Définition** *La classe abstraite*

Elle représente un concept abstrait du domaine. Une classe abstraite ne peut pas être instanciée car une partie de son protocole doit être spécialisé dans une classe concrète pour que la description soit complète.

**Définition** *Le package*

Il représente un groupement d'éléments de modélisation.

Le package permet d'éviter des collisions entre les noms des éléments de modélisation. Les éléments à l'intérieur d'un package sont protégés vis-à-vis de l'extérieur à moins de les rendre explicitement visibles.

**Définition** *Une association*

Déclaration que les instances des classes associées peuvent communiquer.

**Définition** *Un rôle*

Façon dont un objet est « vu » par un autre.

On le note du côté de l'association où le rôle est « joué ».

**Définition** *La responsabilité*

- Effectuer un calcul.
- Stocker / Retourner une valeur.
- Connaître un autre objet.

**Définition** *L'agrégation*

Elle représente l'association entre un objet « composé » et ses « composants ».

**Définition** *Classe d'association*

Elle permet de représenter une information qui émerge parce que deux classes sont associées.

**Définition** *RootSet*

Ensemble des variables accessibles aux programmes à un instant donné : variables globales, variables locales des objets qui s'exécutent.

**Définition** *Objets vivants*

Se sont les objets qui sont atteignables transitivement depuis le RootSet.

---