

VOCABULAIRE DE LA COO

La conception orientée objet (COO) est une approche de programmation qui utilise des objets pour modéliser des données et des comportements dans le développement de logiciels. C'est une manière de structurer un programme en regroupant des attributs (données) et des méthodes (fonctions ou procédures) en unités appelées classes.

1. Entité

Dans la COO, une entité est tout élément du monde réel ou conceptuel ayant une existence distincte et identifiable.

Une « entité » est un concept, un objet physique ou abstrait qui a une existence propre. Dans la conception orientée objet, une entité peut être considérée comme un élément du monde réel avec une identité distincte. Par exemple, une voiture, une personne, ou un livre sont des entités dans leurs contextes respectifs.

Exemple pratique :

- Entité : Livre
- Description : Un livre est une entité physique caractérisée par un titre, un auteur, et un numéro ISBN. Par exemple, un exemplaire du livre « Le Petit Prince » est une entité distincte avec ses propres attributs uniques.

Exemple avancé :

- Entité : Compte Bancaire
- Description : Un compte bancaire est une entité abstraite représentant la relation financière entre un client et sa banque. Chaque compte est unique et identifié par un numéro de compte. Il possède des attributs comme le solde actuel, le titulaire du compte, et l'historique des transactions.

2. Classe

Une classe est une sorte de modèle ou de plan qui définit les caractéristiques et les comportements de quelque chose. En programmation, elle sert à définir les attributs (caractéristiques) et les méthodes (comportements) que les objets créés à partir de cette classe auront. Par exemple, la classe Voiture peut inclure des attributs comme la marque, le modèle, la couleur, et l'année, ainsi que des méthodes pour démarrer, arrêter, accélérer, et freiner.

Autrement, Une classe est un blueprint, un modèle pour créer des objets. Elle est le fondement de la COO, servant de modèle pour créer des objets. Elle décrit un groupe d'objets avec des attributs communs (propriétés) et des méthodes communes (actions ou comportements). La classe définit quelles données sont stockées et comment les objets interagissent avec le monde extérieur.

Une classe est comme une "recette" pour créer des objets. Par exemple, une classe

`Voiture` peut définir des attributs tels que la marque, le modèle, et la couleur, et des méthodes comme démarrer() ou arrêter().

Exemple pratique :

- Classe : Livre
- Attributs : titre, auteur, ISBN, annéeDePublication
- Description : La classe `Livre` sert de modèle pour créer des objets livre, avec des informations générales comme le titre et l'auteur.

Exemple avancé :

- Classe : CompteBancaire
- Attributs : numéroDeCompte, solde, titulaire
- Méthodes : déposer(montant), retirer(montant), afficherSolde()
- Description : La classe `CompteBancaire` définit un modèle pour les comptes bancaires. Elle inclut des méthodes pour gérer le compte, comme effectuer un dépôt ou un retrait, et pour consulter le solde.

Classe

CompteBancaire
numéroDeCompte solde titulaire
déposer retirer afficherSolde

3. Attribut

Un « attribut » est une caractéristique d'une classe. C'est une donnée membre d'une classe qui représente une propriété spécifique de l'entité modélisée par la classe. Par exemple, les attributs d'une classe `Personne` pourraient inclure `nom`, `âge`, et `adresse`.

Les attributs sont les propriétés stockées dans une classe. Ils représentent les caractéristiques de l'entité modélisée par la classe.

Exemple pratique :

- Attributs de la classe Livre : titre (Les Misérables), auteur (Victor Hugo), ISBN (123456789), annéeDePublication (1862).
- Description : Ces attributs définissent les propriétés spécifiques d'un livre.

Exemple avancé :

- Attribut détaillé : solde dans CompteBancaire
- Description : Le `solde` représente la quantité d'argent disponible dans un compte bancaire. Il varie avec les transactions, comme les dépôts et les retraits.

4. Objet

Un objet est une instance spécifique d'une classe. Il possède des valeurs concrètes pour les attributs définis par sa classe.

Exemple pratique :

- Objet : Une instance spécifique du livre 'Les Misérables'.
- Création : `lesMisérables = Livre(Les Misérables, Victor Hugo, 123456789, 1862)`
- Description : Cet objet a des valeurs spécifiques pour les attributs définis dans la classe `Livre`.

Exemple avancé :

- Objet : `monCompteBancaire`
- Création : `monCompteBancaire = CompteBancaire(123456, 1000, Jean Dupont)`
- Description : Cet objet représente un compte bancaire spécifique pour Jean Dupont, avec un numéro de compte 123456 et un solde initial de 1000.

Classe vs objet – Différence entre eux

Voici la différence importante entre classe et objet :

Classe	Objet
Une classe est un modèle pour créer des objets dans un programme.	L'objet est une instance d'une classe.
Une classe est une entité logique	L'objet est une entité physique
Une classe n'alloue pas d'espace mémoire lors de sa création.	L'objet alloue de l'espace mémoire à chaque fois qu'il est créé.
Vous ne pouvez déclarer une classe qu'une seule fois.	Vous pouvez créer plusieurs objets à l'aide d'une classe.
Exemple : Voiture.	Exemple : Jaguar, BMW, Tesla, etc.
La classe génère des objets	Les objets donnent vie à la classe.
Les classes ne peuvent pas être manipulées car elles ne sont pas disponibles en mémoire.	Ils peuvent être manipulés.
Il n'a aucune valeur associée aux champs.	Chaque objet a ses propres valeurs, associées aux champs.
Vous pouvez créer une classe en utilisant le mot-clé « class ».	Vous pouvez créer un objet en utilisant le mot-clé « nouveau » en Java

Objet

CompteBancaire :123456
1000
Jean Dupont

5. Méthode

Les méthodes sont des fonctions associées à une classe. Elles définissent les comportements ou actions que les objets de la classe peuvent exécuter.

Exemple pratique :

- Méthode de la classe Livre : `afficherInfos()`
- Description : Cette méthode pourrait afficher les informations d'un livre, telles que son titre et son auteur.

Exemple avancé :

- Méthode détaillée : déposer(montant) dans CompteBancaire
- Description : La méthode `déposer` permet d'ajouter un certain montant au solde du compte. Si `monCompteBancaire.déposer(200)` est exécuté, le solde du compte est augmenté de 200 euros.

Classe

CompteBancaire
numéroDeCompte
solde
titulaire
déposer
retirer
afficherSolde

6. Encapsulation

L'encapsulation est un principe de la COO qui consiste à cacher les détails d'implémentation d'une classe et à exposer uniquement les méthodes nécessaires pour l'utilisation de l'objet.

Exemple pratique :

- Encapsulation : Cacher le calcul des notes dans une classe `Evaluation`.
- Description : Seules les méthodes pour ajouter une note ou obtenir la note moyenne sont exposées, tandis que le calcul interne de la note moyenne est caché.

Exemple avancé :

- Encapsulation : Accès au solde dans CompteBancaire
- Description : L'attribut `solde` est privé et ne peut être modifié directement. Les méthodes publiques `déposer` et `retirer` sont les seules manières de modifier le solde, assurant ainsi le contrôle et la validation des opérations.

7. Héritage

L'héritage permet à une classe de reprendre (ou d'hériter) les attributs et méthodes d'une autre classe. Cela facilite la réutilisation du code et l'extension des fonctionnalités.

Exemple pratique :

- Héritage : Classe `Roman` héritant de la classe `Livre`.
- Description : La classe `Roman` pourrait ajouter des attributs spécifiques aux romans, comme le genre, tout en héritant des attributs généraux de la classe `Livre`.

Exemple avancé :

- Héritage : `CompteEpargne` hérite de `CompteBancaire`
- Description : `CompteEpargne` est une spécialisation de `CompteBancaire` qui ajoute un attribut `tauxDIntérêt`. Il hérite des méthodes de base tout en pouvant ajouter de nouvelles méthodes ou en modifier certaines (par exemple, calculer les intérêts).

8. Polymorphisme

Le polymorphisme est la capacité des méthodes à effectuer différentes actions en fonction de l'objet sur lequel elles opèrent, même si ces objets proviennent de classes différentes.

Exemple pratique :

- Polymorphisme : Méthode `lireExtrait()`
- Description : La méthode `lireExtrait()` peut présenter un extrait de manière différente selon qu'elle agit sur un roman, un manuel scolaire, ou une bande dessinée, en utilisant le polymorphisme pour adapter son comportement au type de livre.

Exemple avancé :

- Polymorphisme : méthode `calculerIntérêt()` dans `CompteBancaire` et `CompteEpargne`
- Description : Bien que `CompteBancaire` puisse avoir une méthode `calculerIntérêt` de base (qui pourrait ne rien faire ou renvoyer zéro), `CompteEpargne` implémente cette méthode pour calculer et ajouter des intérêts au solde en fonction de son `tauxDIntérêt`.

Exemples de compréhension

Exemple 1 : Film

1. Entité

Imaginez un roman médiéval où les entités seraient '**Chevalier**', '**Château**', et '**Dragon**'. Chacune de ces entités aurait ses propres caractéristiques et comportements.

2. Classe

La classe '**Personnage**' pourrait être définie avec des attributs comme '**nom**', '**force**', et '**allégeance**', et des méthodes comme '**combattre()**' ou '**parler()**'.

3. Objet

Un objet serait **Sir Lancelot**, une instance spécifique de la classe '**Chevalier**'. **Sir Lancelot** aurait des attributs comme un nom spécifique et un niveau de force.

4. Attribut

Dans l'objet **Sir Lancelot**, '**nom**' pourrait être "**Lancelot du Lac**", et '**force**' pourrait avoir une valeur de **8 sur 10**.

5. Héritage

La classe '**Chevalier**' hérite de la classe '**Personnage**' mais ajoute un attribut '**chevalerie**' et une méthode '**jouter()**'.

6. Polymorphisme

La méthode '**combattre()**' pourrait être redéfinie dans la classe '**Chevalier**' pour inclure le combat à cheval, ce qui montre le polymorphisme – la même méthode '**combattre()**' a un comportement différent pour un '**Chevalier**' que pour un '**Dragon**', par exemple.

7. Taxonomie

Dans notre histoire, la taxonomie serait la classification des différents types de personnages : tous les '**Chevaliers**', '**Magiciens**', et '**Dragons**' seraient des sous-classes de la classe plus générale '**Personnage**'.

8. Constructeur

Un constructeur pour Sir Lancelot pourrait ressembler à cela : ``SirLancelot = Chevalier("Lancelot du Lac", 8, "Roi Arthur")``, qui initialise l'objet avec son nom, sa force, et son allégeance.

9. Destructeur

Lorsque l'histoire se termine et que **Sir Lancelot** est écrit hors de l'histoire, le destructeur pour l'objet Sir Lancelot pourrait être appelé pour "nettoyer" après lui.

10. Signature

Une signature de méthode pourrait être ``combattre(self, adversaire)``, qui définit comment la méthode ``combattre`` est appelée et qu'elle nécessite un objet ``adversaire``.

11. Symbole

Dans le récit, les symboles pourraient être les noms des personnages ou des objets importants, comme **Excalibur**, qui représente plus qu'une simple **épée**, mais un pouvoir et une légitimité royale.

12. Emblème

Un emblème pourrait être un blason ou un étendard qui représente une maison ou un royaume dans l'histoire, symbolisant l'honneur et l'appartenance.

Exemple 2 : Conte de fées

1. Entité

Dans un conte de fées, une entité pourrait être une **"Fée"**. L'entité Fée serait un concept général avec des attributs communs comme des ailes, la capacité de voler et de lancer des sorts. En littérature, une entité serait comme un archétype, un élément récurrent avec des caractéristiques spécifiques.

2. Classe

Une classe serait comme le genre littéraire. Par exemple, le genre **"Roman d'aventure"** est une classe qui a des caractéristiques communes telles que des **héros**, des **voyages**, des **quêtes**, etc. Chaque livre du genre partage ces caractéristiques mais les détails varient d'une histoire à l'autre.

3. Objet

En utilisant l'exemple du genre **"Roman d'aventure"**, un objet serait un livre spécifique comme **"L'île au trésor"** de **Robert Louis Stevenson**. C'est une instance concrète du genre avec ses propres personnages, son intrigue et ses paramètres.

4. Attribut

Les attributs d'un **"Roman d'aventure"** pourraient inclure le nom du héros, le lieu de l'aventure, et l'objet de la quête. Dans **"L'île au trésor"**, ces attributs sont respectivement **Jim Hawkins**, **l'île au trésor** et le **trésor lui-même**.

5. Héritage

Prenons l'exemple de la mythologie. La classe **"Dieu"** peut avoir des enfants qui héritent de certains attributs et pouvoirs. Dans la mythologie grecque, **Athéna** est une **déesse** qui hérite de certains attributs de son père **Zeus**.

6. Polymorphisme

Dans les **fables**, un personnage comme un **shape-shifter** peut prendre différentes **formes** - un **loup**, un **humain**, ou une **brise**. Cela illustre le polymorphisme, où le personnage a une méthode **'transformer'** qui a différents comportements selon la forme qu'il prend.

7. Taxonomie

Dans une encyclopédie de créatures mythiques, vous pourriez avoir une taxonomie qui classe les créatures par caractéristiques comme les **créatures volantes**, les **créatures marines**, et les créatures terrestres. C'est une classification hiérarchique basée sur des attributs communs.

8. Constructeur

Lorsqu'un conteur crée un nouveau **personnage de sorcière**, il peut choisir **son nom, ses pouvoirs et ses faiblesses** au début de l'histoire. Cela ressemble à l'utilisation d'un constructeur en POO pour initialiser un nouvel objet avec des valeurs spécifiques.

9. Destructeur

Lorsqu'**une sorcière** est vaincue à la fin de l'histoire et que sa présence est effacée du récit, c'est comme si un destructeur était appelé pour terminer son arc narratif et libérer son rôle dans l'histoire.

10. Signature

Si **une sorcière** a une capacité spéciale appelée **`ensorceller(cible)`**, la signature de cette capacité inclurait le **nom 'ensorceller'** et l'argument **'cible'**. C'est la **"signature"** de son action magique.

11. Symbole

Dans la littérature, un symbole est un élément qui représente des idées plus grandes. Par exemple, **le phénix** symbolise souvent la renaissance et le renouveau. En POO, un symbole est un identifiant qui représente un objet ou une valeur.

12. Emblème

Un emblème est un objet représentatif ou une image avec un symbolisme significatif. Dans **"Le Seigneur des Anneaux"**, l'Anneau Unique est un emblème de pouvoir et de corruption.