

## Diagramme de classe UML

Introduction : Ce TP/TD porte sur la modélisation (et pas du tout sur l'écriture de code). C'est un travail de conception qui, dans la séquence des phases de développement d'un projet, se situe vers l'amont. Ce qui vous est demandé dans chaque exercice c'est d'avoir une réflexion sur les classes à écrire, sur leurs interactions, et d'être capable de traduire vos idées dans un diagramme soigné et compréhensible.

Cette étape permet de lever certaines ambiguïtés des spécifications, et de poser d'important jalons dans le processus qui aboutira à une implémentation. Vous ferez ainsi des choix conceptuels puis techniques clairs qui serviront de guide au programmeur. Il est donc important de comprendre la syntaxe UML et de l'utiliser à bon escient.

L'héritage par exemple est représenté par une flèche ; l'agrégation l'est par un losange creux ; la composition par un losange aussi (mais il est noir) etc...

Encore faut-il bien comprendre ces concepts. En particulier la différence entre agrégation et composition. On rappelle rapidement que lorsqu'on dit qu'un objet en agrège d'autres, cela signifie qu'il stocke en interne des références vers ces objets qui existent par ailleurs. Alors qu'on dira qu'un objet A est composé par d'autres objets lorsque ces objets seront stockés dans A. On comprend bien la différence lorsqu'on réfléchit à la destruction d'un objet : on sait ce que deviennent ses constituants selon la façon dont on a décidé de les associer.

Les relations entre objets comportent également des cardinalités. Là aussi la convention doit être claire, surtout qu'il existe des pratiques différentes (UML, merise ...). Dans ce cours la cardinalité représente *un intervalle qui borne le nombre d'instances qui participent à une relation étant donné les autres extrémités fixées de manière quelconque*.

Outils Vous pouvez installer différents logiciels pour travailler chez vous, par exemple **Bouml** ou **Staruml**. Certains d'entre eux possèdent même des fonctionnalités qui proposent de traduire les diagrammes en des squelettes de code (ce que nous ne ferons pas car nous voulons maîtriser cette étape).

Pour cette séance nous allons travailler à la fois sur papier et sur un outil en ligne pour nous assurer de la qualité uniforme du rendu : <https://app.diagrams.net/>. Vous choisirez "Device" comme emplacement où sauvegarder vos modèles, puis "Create new Diagram", et enfin "Class Diagram" en précisant le nom que vous voulez donner à votre fichier selon l'exercice que vous faites.

Notez que dans le panel de gauche vous avez un item **UML** qui contient les représentations dont vous pourrez avoir besoin.

Un conseil : réfléchissez d'abord avec un papier et un stylo. N'hésitez pas à discuter avec votre voisin.

**Exercice 1** Une banque compte plusieurs agences réparties sur le territoire français. Une banque est caractérisée par son nom commercial, le nom de son directeur général, son capital global et l'adresse de son siège social. Une agence a un numéro d'agence et une adresse. Chaque agence emploie plusieurs conseillers, qui se caractérisent par leur nom, prénom, salaire et date d'embauche. Les conseillers peuvent demander leur mutation d'une agence à l'autre, mais chaque conseiller ne peut travailler que dans une seule agence. Les conseillers d'une agence ne font que gérer des clients. Un client ne peut avoir des comptes que dans une seule agence de la banque et se voit attribuer un unique conseiller de l'agence. Il a un nom, prénom et adresse. Les comptes ont un solde, un taux d'intérêt (éventuellement nul) et rapportent des intérêts annuellement.

1. Créez le diagramme de classes pour modéliser les relations liant banque, agence, conseiller, client et compte.
2. En fait la banque emploie différents salariés : conseillers, directeur général, vigiles, comptables et agents de guichet. Modifiez votre diagramme pour prendre en compte cette structure regroupant les différents salariés tous caractérisés par un nom, un prénom, un salaire et une date d'embauche.

**Exercice 2** Vous allez modéliser un mini réseau social.

Le réseau social est composé d'utilisateurs. Un utilisateur est caractérisé par son nom, son prénom, une adresse mail, sa date de naissance et son statut (étudiant, salarié, ...). Il peut faire partie ou non d'un ou plusieurs groupes. Un groupe est caractérisé par son nom, sa date de création et un administrateur. Les utilisateurs communiquent entre eux via des conversations composées de messages. Une conversation vide n'a pas de sens. Un message est composé d'un texte et d'un écrivain (un utilisateur). Il y a deux types de conversations, la conversation de groupe qui implique tous les membres d'un groupe et la conversation privée qui ne concerne qu'un ensemble ciblé d'utilisateurs. Les conversations peuvent être surveillées par des modérateurs qui peuvent décider de les bloquer. Un modérateur surveille plusieurs conversations mais une conversation n'est surveillée que par au plus un seul modérateur. Un modérateur est caractérisé par un nom et prénom.

Créez proprement un diagramme de classes pour votre réseau social.

**Exercice 3** On souhaite concevoir une application gérant une université. La structure d'une université est la suivante:

- chaque université est structurée par des UFRs;
- chaque UFR regroupe des enseignants dont l'un est directeur;
- chaque UFR est administrée par un responsable administratif, qui peut être partagé par plusieurs UFRs;
- chaque UFR est dotée de personnels administratifs et techniques;
- les enseignants peuvent intervenir dans plusieurs matières;
- une matière peut être enseignée par plusieurs enseignants dans différentes salles (amphi, salle de TP, salle de TD);
- les étudiants sont notés dans chaque matière.

Créez un diagramme de classes correspondant à votre solution.