M22- GENIE LOGICIEL ET CONCEPTION ORIENTEE OBJETS (UML)

Lotfi NAJDI

Année Universitaire 2021 / 2022

Génie Informatique

FPT Taroudant

Modélisation statique et UML

Diagramme de Classe

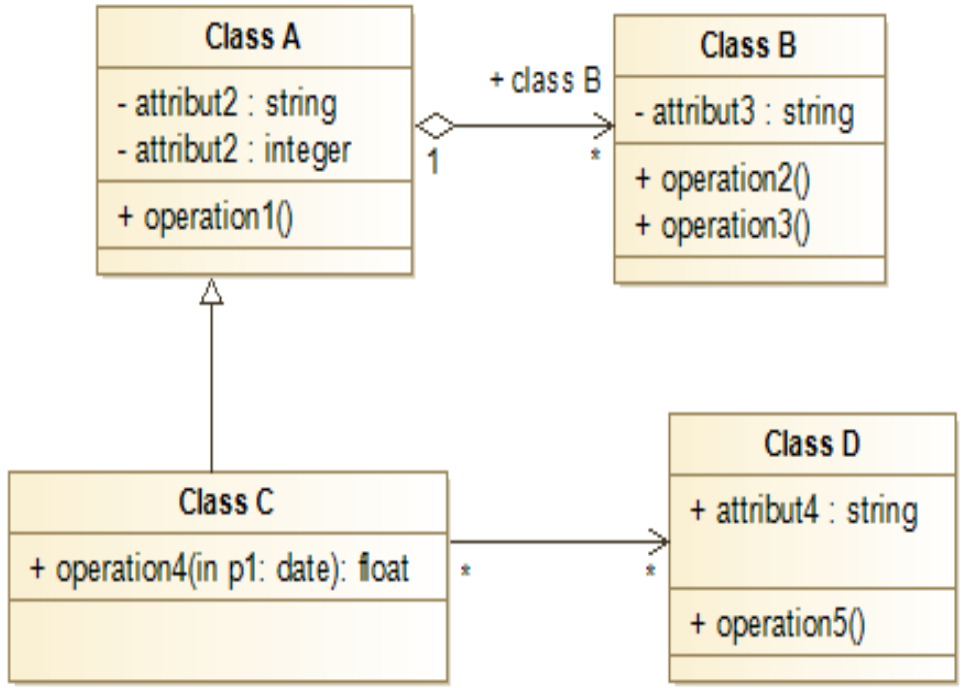
Diagramme de classes

- Diagramme le plus important et le plus utilisé dans toute approche orientée objet.
- Représentation graphique des classes (du système étudié) et les relations qui existent entre celles-ci
- Représentation statique de la structure interne du logiciel
- Diagramme riche en notations, utilisé par les outils de génération automatique de code.

Diagramme de classes

Le diagramme de classe permet de décrire la structure du système :

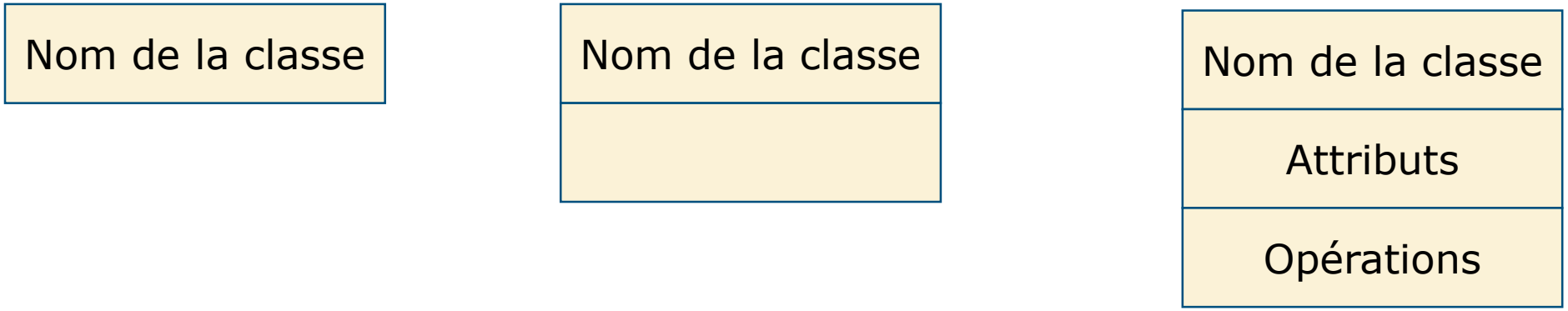
- classes
- attributs
- Opérations
- relations entre classes (associations , agrégations , compositions, généralisations..etc.)



Classe et objet

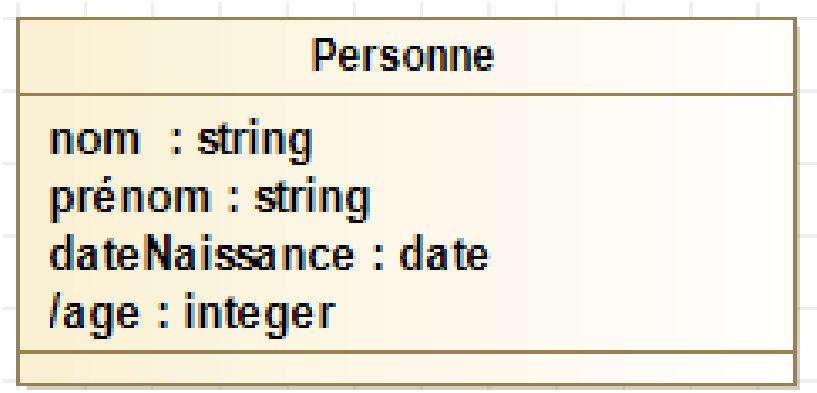
Une classe représente la description abstraite d'un ensemble d'objets possédant les mêmes caractéristiques(attributs, opérations, relations et sémantique)

Graphiquement, une classe est représentée par un rectangle, comprenant généralement son nom, ses attributs et ses opérations dans des compartiments séparés et désignés.



Les attributs

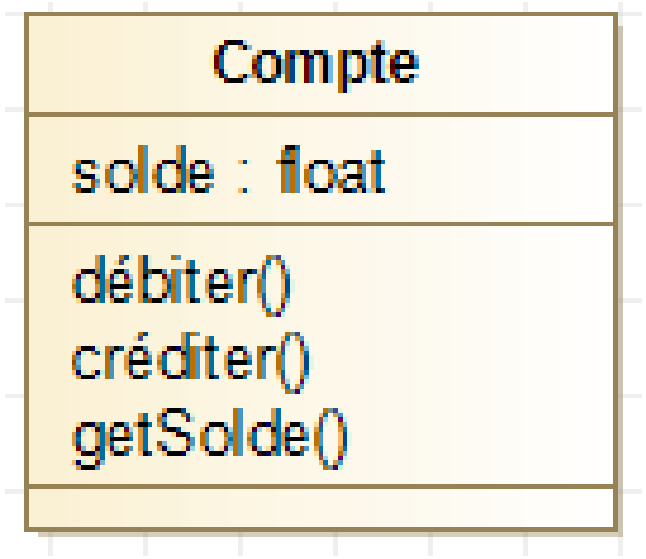
- Un attribut représente un type d'information contenu dans une classe.
- Un attribut est une propriété nommée d'une classe qui décrit l'objet modélisé.
- Les attributs sont généralement listés sous la forme: nom de l'attribut : Type



- Un attribut dérivé est un attribut qui peut être calculé à partir d'autres attributs.
- Un attribut dérivé est reprenseté par un '/' devant le de l’attribut

Les opérations

- Une opération représente un élément de comportement (un service) contenu dans une classe.
- Les opérations apparaissent dans le troisième compartiment.



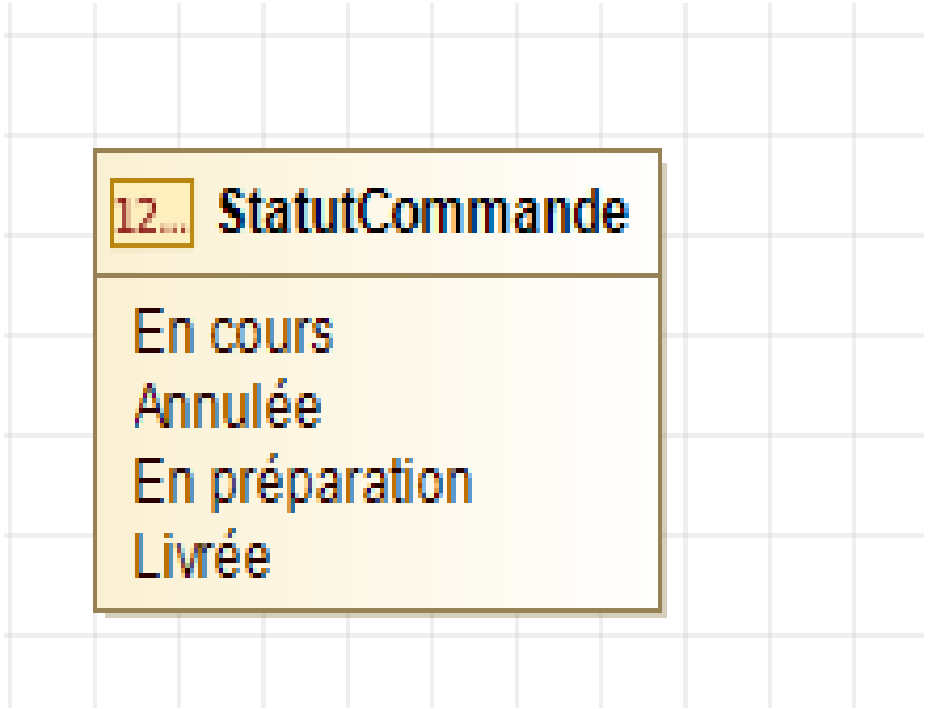
Marqueur de visibilité

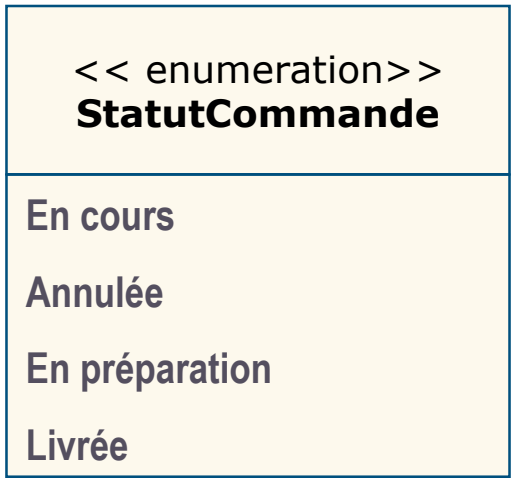
UML propose les types de visibilité suivants :

+	public	tout le monde
#	protected	les sous-classes
-	private	privées
~	package	espace de nommage

Énumérations

Dans les modèles UML, les *énumérations* sont des éléments de modèle dans les diagrammes de classes qui représentent des types de données définis par l'utilisateur. Les énumérations contiennent des ensembles d'identificateurs nommés qui représentent les valeurs de l'énumération. Ces valeurs sont appelées littéraux d'énumération.





Relation

En UML, les interconnexions (logiques ou physiques) entre les objets sont modélisées comme des relations.

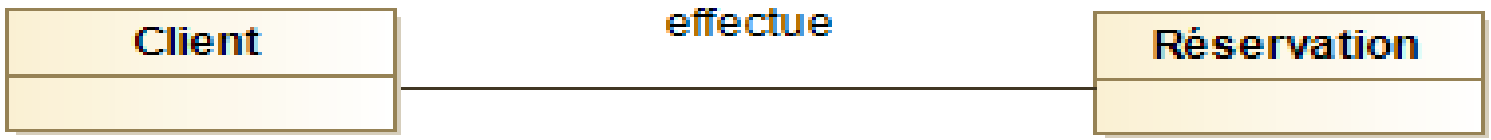
Il existe trois types de relations en UML :

- les associations
- Les généralisations
- Les dépendances

ASSOCIATION

- Une association représente une relation sémantique particulière entre des classes, qui porte un nom.
- Une association permet de naviguer entre les classes (liens possibles entre les instances des classes associée)
- Une association possède un nom constitué ,généralement, d’un verbe en présent.
- Dans la plupart des cas on a des associations entre deux classes (association binaire) , mais on peut avoir des association entre plus de deux classes (association n-aire)

ASSOCIATION



Un Client peut effectuer (acheter) des Réservations. La relation effectue est une association entre les classes Client et la classe Réservation.

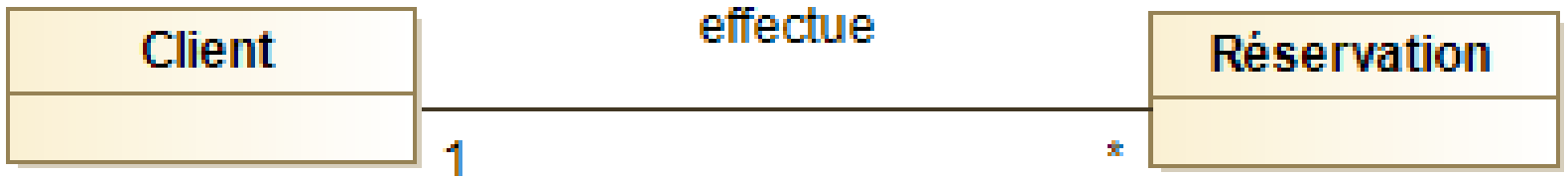
Le verbe qui désigne l'association ne privilégie pas forcement un seule sens de lecture, Donc implicitement, l'exemple précédent inclut également le fait qu'une Réservation est effectuée par un Client.

Multiplicité

- Sur chaque extrémité d’une association on peut mettre une indication de multiplicité.
- Contrainte sur le nombre d'objets (min et max) de la classe cible pouvant être associés à un seul objet donné de la classe source (la classe de l'autre terminaison de l'association).
  - 1 ou 1..1 : un et un seul
  - * ou 0..* : plusieurs
  - 1..* : au moins un
  - m. .n : de m à n (entiers > 0)

Multiplicité

La multiplicité est placée de l'autre côté de la classe utilisée comme sujet, du côté du complément d'objet direct.



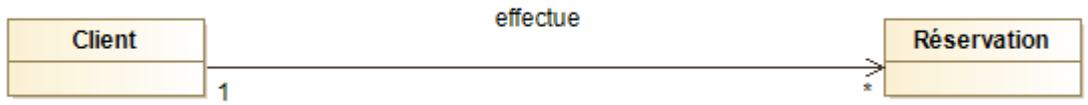
Navigabilité

Qualité d'une association qui permet le passage d'une classe à l'autre dans une direction donnée.

La navigabilité :

- indique la possibilité de traverser une association.
- est représentée graphiquement par une flèche du côté de la terminaison navigable.
- est limitée par une croix du côté de la terminaison non navigable

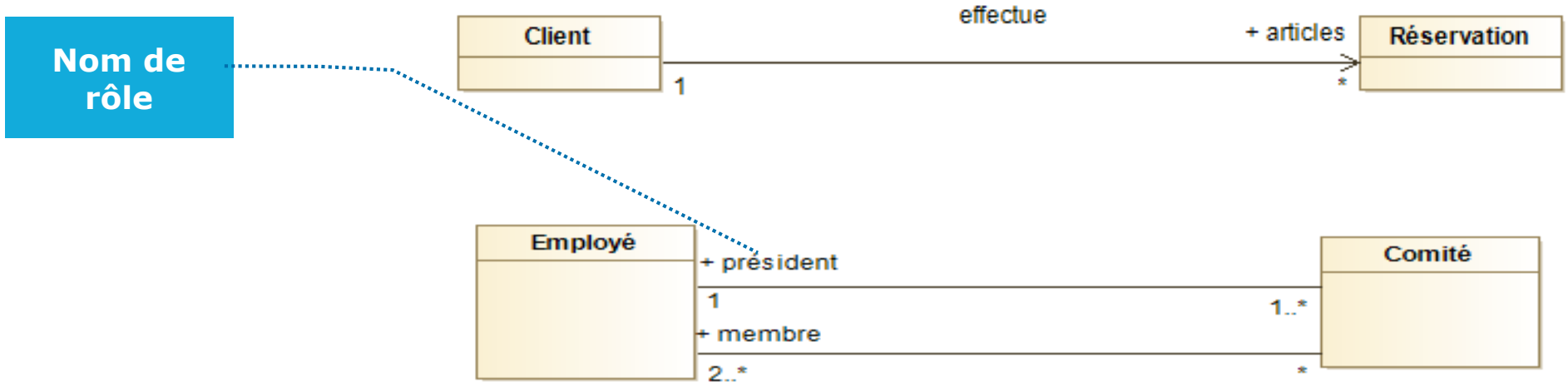
Par défaut, une association est navigable dans les deux sens.



- La terminaison du côté de la classe Client n'est pas navigable
  ➔ les instances de la classe Réservation ne stockent pas de liste d'objets du type Client.
- En revanche, la terminaison du côté de la classe Réservation est navigable
  ➔chaque instance de la classe Client contient une liste d'objets de type Réservation.

Nom de rôle

- La terminaison d’une association peut avoir un nom appelé « *nom du rôle* ».
- Le nom est situé à proximité de la terminaison.
- Une association peut avoir autant de rôle que de terminaisons
- Ce nom représente la manière dont les instances d'une classe accèdent aux instances d'une autre classe.

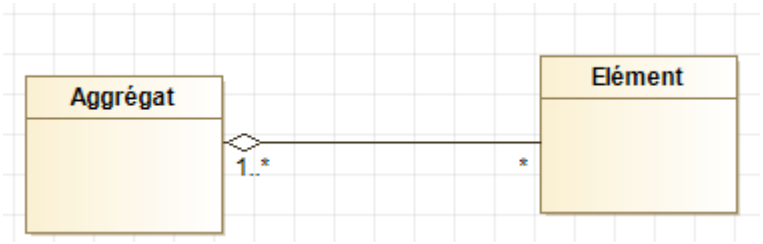


Relation d’agrégation/composition

- Une agrégation est un cas particulier d'association qui modélise la notion de contenance et de possession
- Les agrégations n'ont pas besoin d'être nommées : implicitement elles signifient la relation « contient », «se compose de ».
- une agrégation est une association avec une sémantique du type « un tout (un agrégat) vers les parties de ce tout ».
- Une relation d’inclusion (hiérarchique) structurelle ou comportementale d'un élément dans un ensemble (« contenant / contenu » , « ensemble / élément »)

Agrégation

- L'agrégat fait référence à ses parties
- La création ou destruction du l'agrégat est indépendante de la création ou destruction de ses parties
- Un objet peut faire partie de plusieurs agrégats à la fois



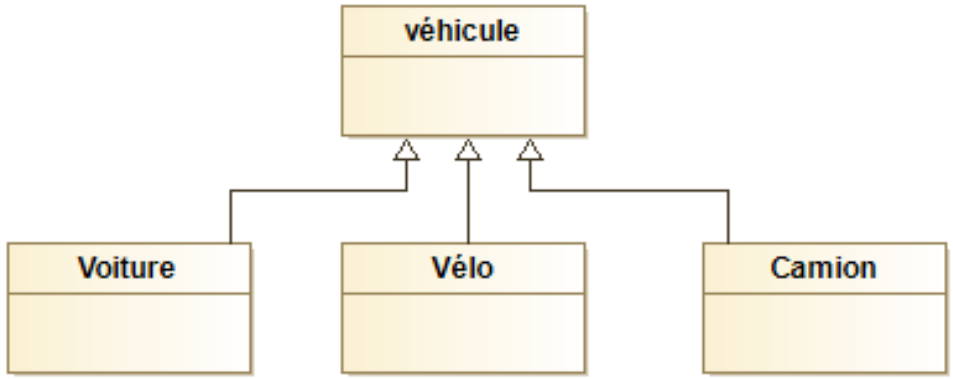
Composition

- Une composition est une agrégation particulière dans laquelle la durée de vie des parties est liée à celle du « tout ».
- Forme forte d'agrégation, dans laquelle les parties ne peuvent appartenir à plusieurs agrégats et où le cycle de vie des parties est subordonné à celui de l'agrégat.

Relation de Généralisation

Une super-classe est une classe plus générale reliée à une ou plusieurs autres classes plus spécialisées (sous-classes) par une relation de généralisation.

Les sous-classes « héritent » des propriétés de leur super-classe et peuvent comporter des propriétés spécifiques supplémentaires.



Dépendances

- Une dépendance existe entre deux éléments si des changements dans la définition d'un élément (source) peuvent entraîner des changements dans l'autre (cible).
- Concernant les classes, les dépendances existent pour diverses raisons : Une classe envoie un message à une autre , une classe mentionne une autre comme paramètre d'une opération (passage d'objets en tant qu'argument) ou bien une classe comporte une autre dans ses propres caractéristiques.
- Pour plus de détails voir Types de relation de dépendance.

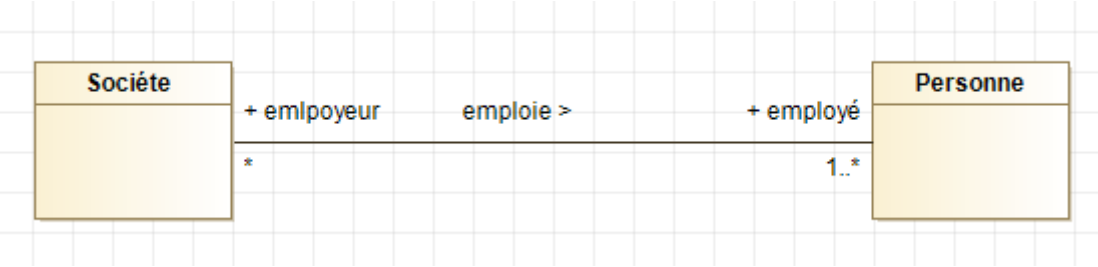
Dépendances

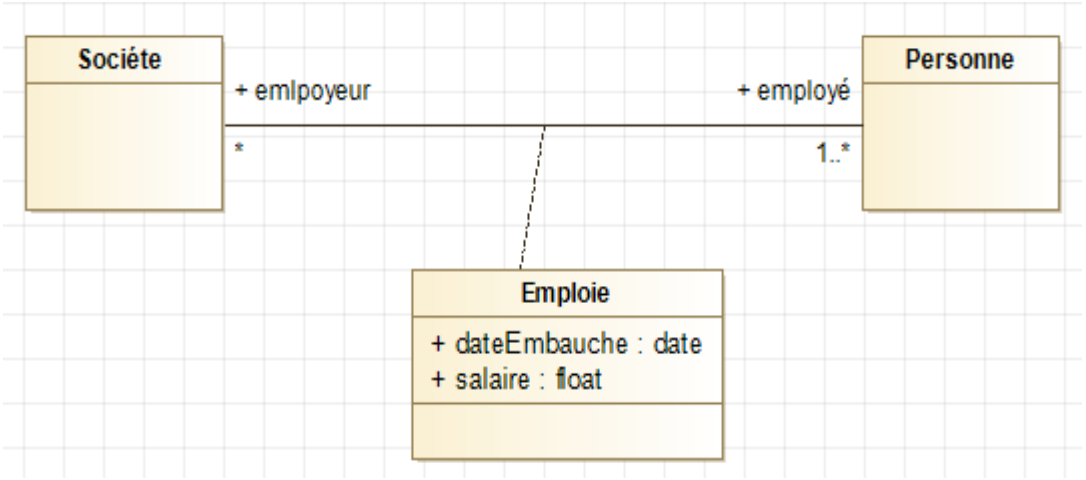
Une dépendance est représentée sous la forme d'une ligne en pointillés avec une flèche ouverte dirigée de la cible(élément dépendant) vers la source.

Une classe EmploisTemps dépend d'une classe Matière parce que la classe EmploisTemps utilise la classe Matière comme paramètre pour une opération d'ajout. Dans un diagramme de classes, une relation de dépendance est dirigée de la classe EmploisTemps vers la classe Matiere.

Classe d’association

- Une classe-association est un élément ayant à la fois les caractéristiques d’une classe et d’une association(reliée à deux ou plusieurs classes et possède également des attributs et des opérations)
- Concept UML absent dans les langages de programmation objet, d'où le besoin de transformation en classe normale avec l’ajout des variables de type référence.





PACKAGE

Un Package est un dispositif général qui permet le regroupement d'éléments en UML.

En Package permet de regrouper des classes ainsi que d'autres packages.

En conception orienté objet, les packages peuvent être utilisés pour regrouper des classes et des associations.

Les packages correspondent à des espaces de noms (deux éléments ne peuvent pas porter le même nom au sein du même package).

La structuration d’un diagramme de classe en packages doit respecter deux principes :

- *Cohérence qui* consiste à regrouper les classes qui sont proches d'un point de vue sémantique.
- *Indépendance :* s'efforce de minimiser les relations entre packages, c'est-à-dire plus concrètement les relations entre classes de packages différents.

Exercice diagramme de classe

Réaliser le diagramme de classe relative à un système simplifié de réservation de vols pour une agence de voyages. Les interviews avec experts métier auxquelles ont permis de résumer leur connaissance du domaine sous la forme des phrases suivantes :

1. Des compagnies aériennes proposent différents vols.
2. Un vol est ouvert à la réservation et refermé sur ordre de la compagnie.
3. Un client peut réserver un ou plusieurs vols, pour des passagers différents.
4. Une réservation concerne un seul vol et un seul passager.
5. Une réservation peut être annulée ou confirmée.
6. Un vol a un aéroport de départ et un aéroport d'arrivée.
7. Un vol a un jour et une heure de départ, et un jour et une heure d'arrivée.
8. Un vol peut comporter des escales dans des aéroports.
9. Une escale a une heure d'arrivée et une heure de départ.
10. Chaque aéroport dessert une ou plusieurs villes.

UML 2 par la pratique

Exercice diagramme de classe

Diagramme d’objets

Diagramme d’objets

Le diagramme d'objets permet de représenter les instances des classes, c'est-à-dire des objets.

Propose une image instantanée des instances d'un système et des relations entre ces instances.

Utilisé principalement pour présenter des exemples de la structures de données :

- Peut être crée pour illustrer le diagramme de classe et vérifier si il est complet et exact.
- Ou bien avant de créer un diagramme de classes se familiariser avec les concepts du domaine et leurs liens.

Diagramme d’objets

Symboles et notations :

- Chaque objet est représenté par un rectangle dans lequel le nom de l'objet et sa classe sont soulignés et séparés par deux points.

- Comme pour les classes, les attributs des objets sont énumérés dans un compartiment distinct. Mais contrairement aux classes, les valeurs doivent être assignées aux attributs des objets.

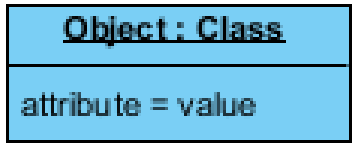


Diagramme d’objets