



Diagramme

Classes et des Objets

Classes et objets

Un objet est un membre d'un système orienté objet.

C'est une entité atomique qui possède :

- une identité qui le caractérise de façon non ambiguë,
- un état représenté par le contenu de ses attributs et les liens qu'il a avec les autres objets,
- un comportement qui regroupe les compétences d'un objet et décrit ses actions et ses réactions (messages reçus , messages envoyés).

Les objets communiquent entre eux par des messages.

La persistance des objets est la capacité pour un objet de sauvegarder son état dans un système de stockage de l'information.

Classes et objets

Un objet a une vie :

- il naît,
- il vit,
- il meurt.

Il est symbolisé par :

Nom objet

ou

Nom objet : Classe

ou

: Classe

Le stéréotype de la classe peut surmonter le nom

<<StéréotypeClasse>>
Nom objet : Classe

Classes et objets

Une classe est un regroupement d'objets ayant les mêmes propriétés ou attributs, les mêmes comportements, les mêmes relations avec les autres objets.

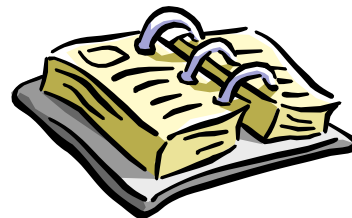
Un objet est une instance d'une classe.

Une classe nous permet de mieux gérer la complexité car c'est une définition abstraite mettant en lumière les points communs entre les objets qu'elle regroupe.

Le nom d'une classe provient directement du domaine. Il commence par une lettre majuscule :

Commandes, Clients, LigneCommandes

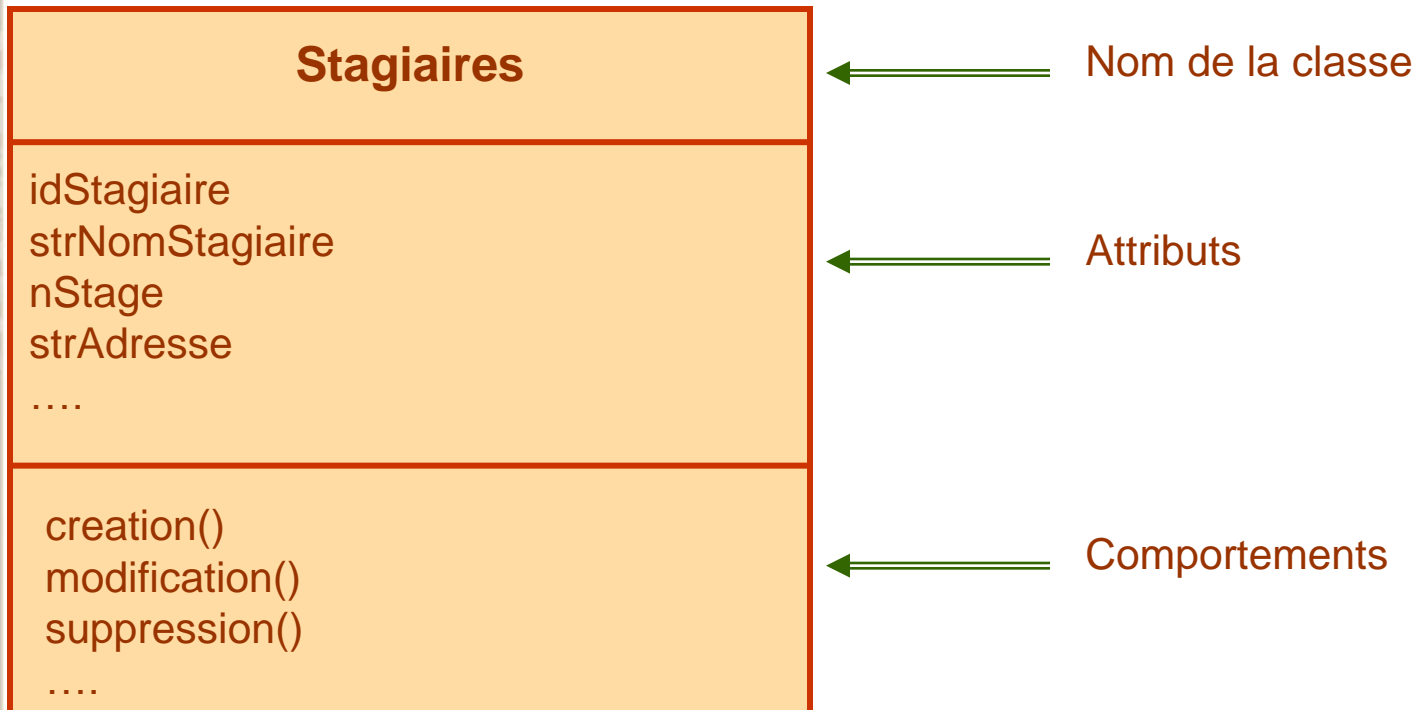
Classes et objets



Combien de classes voyez-vous ?

Classes et objets

Une classe est représentée par :



Un attribut est représenté par :

- un nom qui est une chaîne de caractères.
- type :
- visibilité :
 - public :

l'élément est visible par toutes les classes.
le nom de l'attribut est précédé du symbole +
 - privé
l'élément n'est visible que par sa propre classe.
Le nom de l'attribut est précédé du symbole -

Classes et objets les opérations

Une méthode a un nom qui est une chaîne de caractères

La syntaxe pour définir une méthode est la suivante :

][visibilité] nom [(listeParamètres)]

- visibilité : même possibilité que pour les attributs
 - + publique
 - privé
- listeParamètres
 - syntaxe d 'un paramètre
 - o nomParamètre : typeParamètre

Une classe "entity" modélise les informations persistantes et les comportements associés.

Les caractéristiques d'une telle classe sont :

- Elle reflète une réalité,
- Elle a besoin des tâches internes du système,
- La valeur de ses attributs est souvent fournie par un acteur,
- Son comportement est indépendant des intervenants.

Classes et objets

Les relations

Pourquoi ?

- Tout système comporte de nombreuses classes et objets,
- Les objets contribuent au comportement du système en collaborant avec d'autres objets; cette collaboration est représentée par des relations.

Il existe deux grands types de relations

- L'association
- L'agrégation

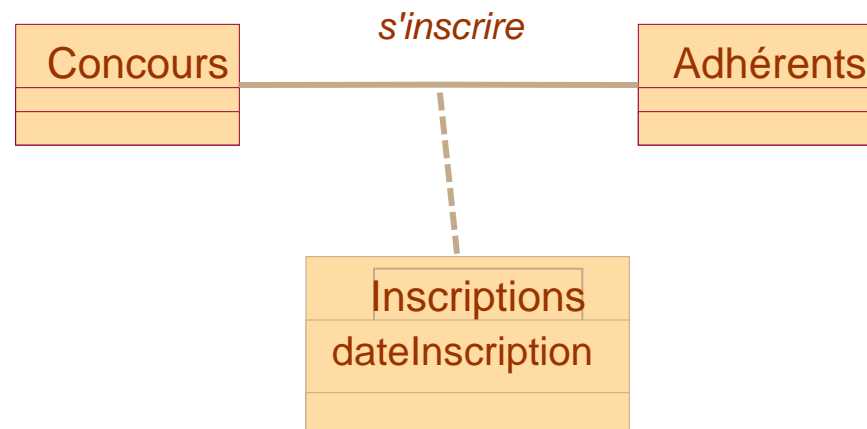
Classes et objets

Les relations : association

Association = connexion sémantique bidirectionnelle entre deux classes



Association binaire non porteuse d'attributs

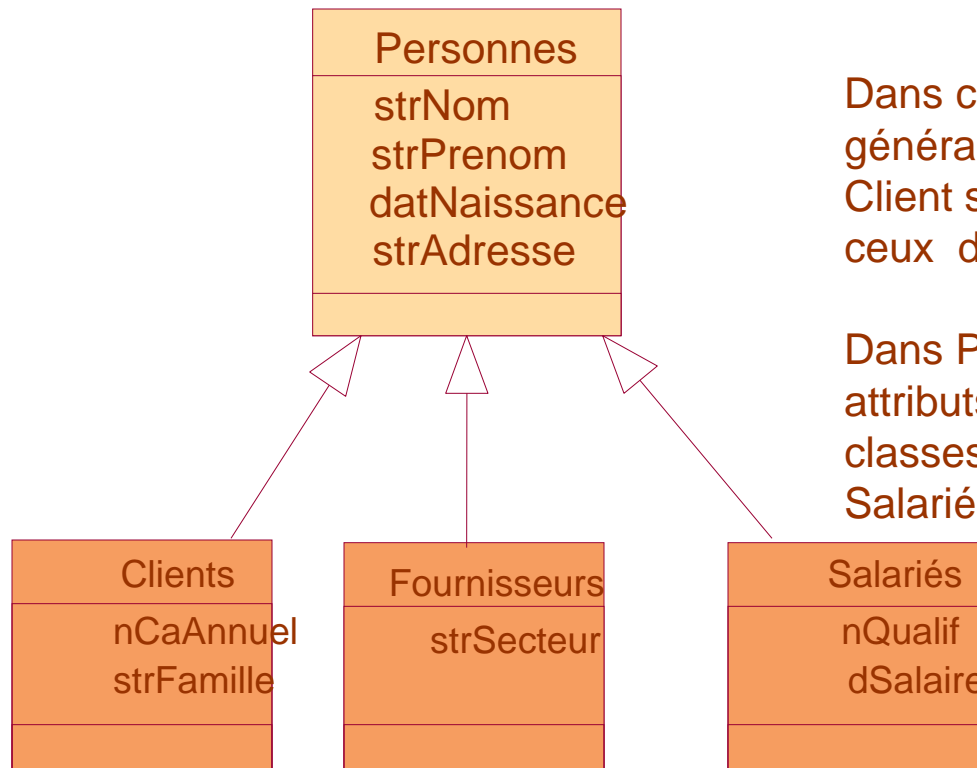


Classe association : Association porteuse d'attributs

Classes et objets

Les relations: généralisation

La généralisation est une relation entre un élément général (la super classe) et un élément plus spécifique.



Dans cet exemple de généralisation, les attributs de Client sont ceux de Personnes plus ceux de Client.

Dans Personnes, on retrouve les attributs qui sont communs aux classes Clients, Fournisseurs, Salariés.

Classes et objets

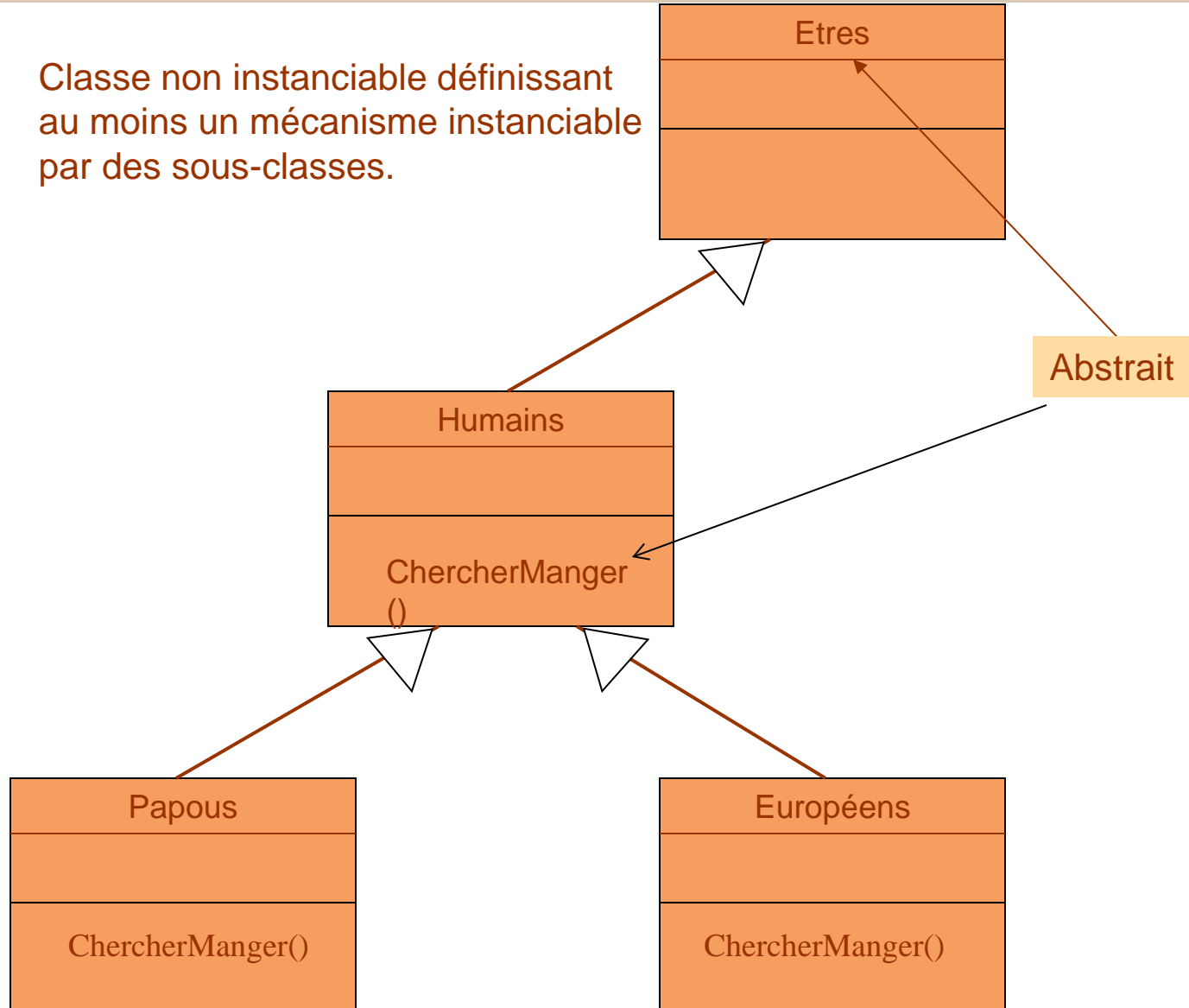
Les relations: héritage

- C'est la technique qui permet de réaliser la généralisation / spécialisation.
- Il permet d'établir une hiérarchie de classes dont la classe de plus haut niveau est la racine.
- Les sous-classes héritent des attributs et opérations de la super-classe.
- Il peut y avoir :
 - Héritage simple (une super-classe).
 - Héritage multiple (plusieurs super-classes).
- Généralisation :
 - Mise en commun d'attributs et méthodes de classes
 - On crée des super-classes
- Spécialisation
 - On détecte des particularités d'un ensemble d'objets
 - On crée des sous-classes

Classes et objets

Les classes abstraites

Classe non instanciable définissant au moins un mécanisme instanciable par des sous-classes.

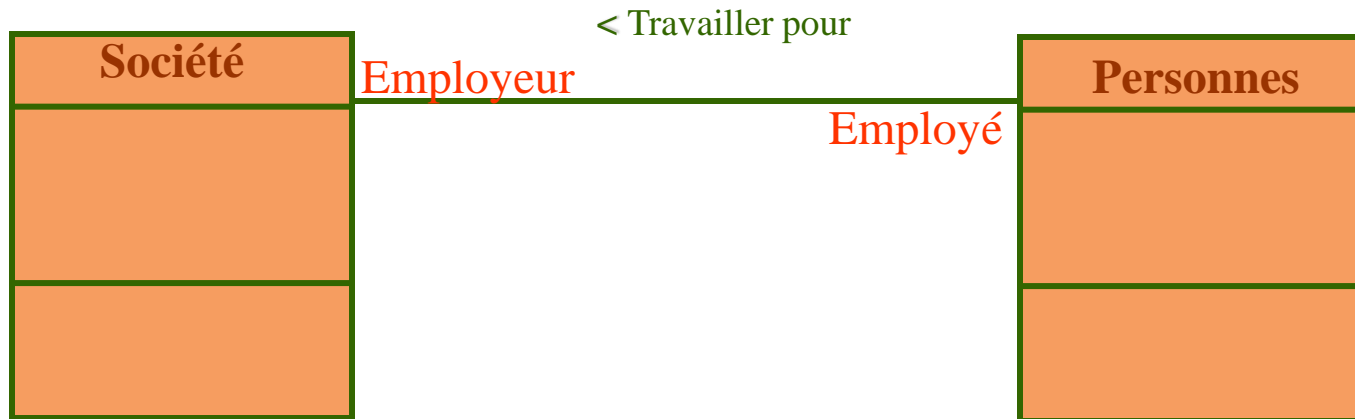


Classes et objets

Les relations : les rôles

L'extrémité d'une association est appelée rôle.

Le rôle décrit comment une classe voit l'autre classe au travers d'une association.



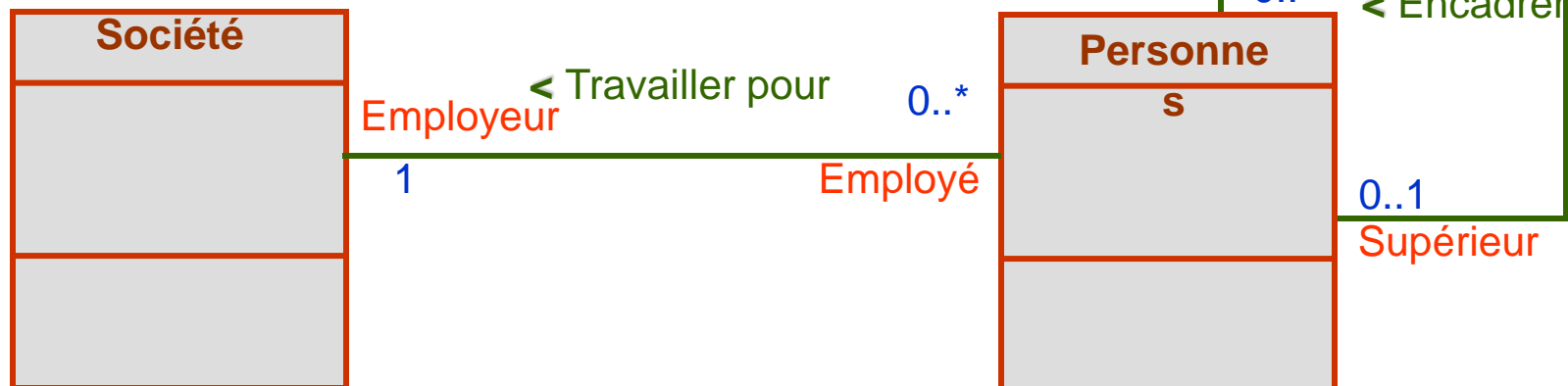
Classes et objets

Les relations : les cardinalités

Chaque rôle d'une association porte une indication de multiplicité indiquant combien d'objets de la classe considérée peuvent être liés à un objet de l'autre classe

(c'est la même notion que les cardinalités merise sauf que l'indication se fait à l'inverse).

1	Un et un seul
0..1	0 ou un
*	De 0 à plusieurs
0..*	
1..*	De un à plusieurs
M..N	De m à n occurrences



Classes et objets

Exemple

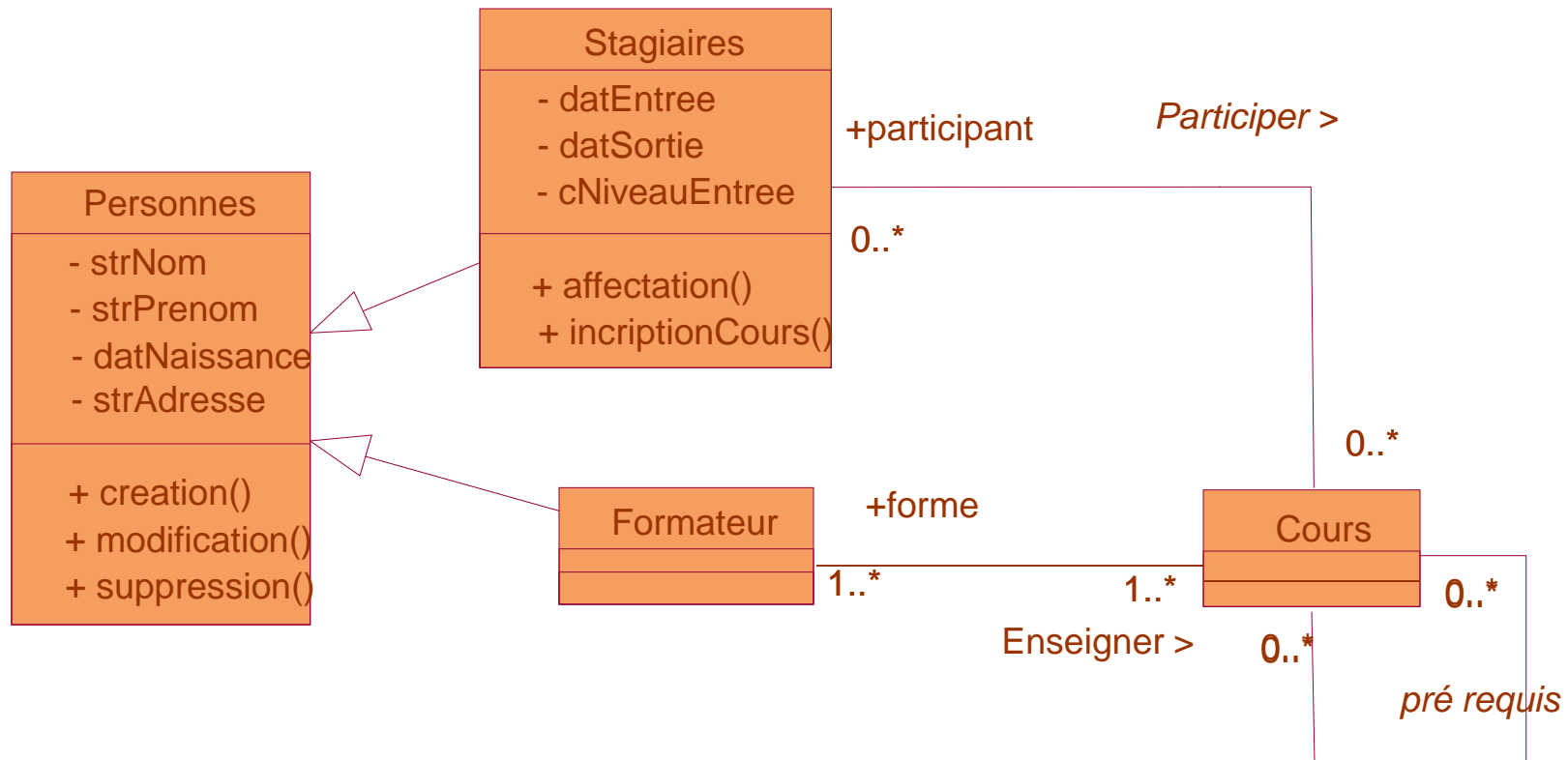


Diagramme de cas d'utilisation

Exemple

Cahier des charges :

Un concessionnaire automobile possède un stock de véhicules qu'il tient à jour.

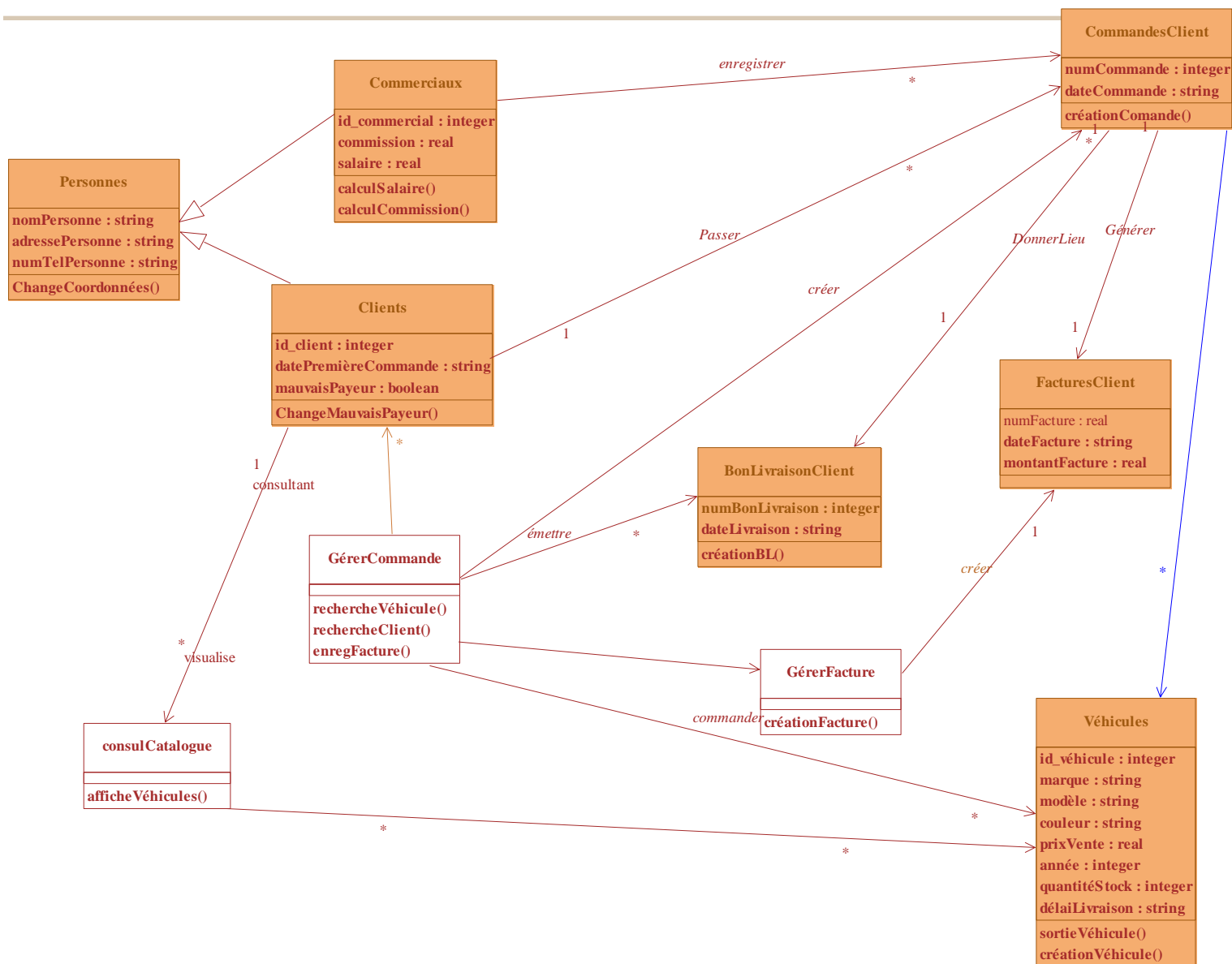
Il a plusieurs commerciaux qui sont chargés d'enregistrer les informations sur des prospects (clients potentiels) qu'ils ont contacté ainsi que les informations concernant les clients réels. Ils enregistrent les commandes. Cet enregistrement permet l'émission d'une facture avec enregistrement du règlement et édition du document.

Le concessionnaire peut également enregistrer une commande pour un client déjà connu.

Les clients disposent d'un catalogue qu'ils peuvent consulter pour les aider dans leur choix de véhicule.

Diagramme de classe

Exemple



Classes et objets

Paquetage

- Un paquetage ou package est un mécanisme servant à organiser des éléments en groupes.

- Les packages peuvent être mis en relation

