

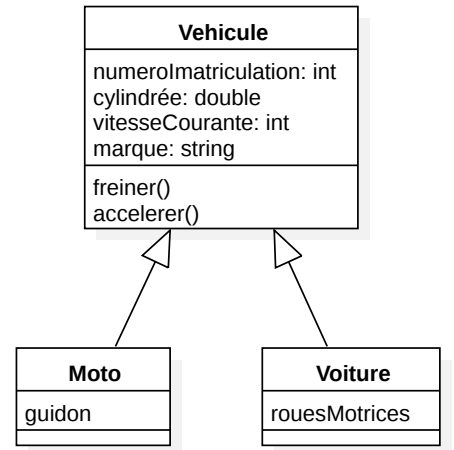


Généralisation-Héritage

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. La notion d'héritage dans les diagrammes de classes est très proche de la notion de spécialisation/généralisation du diagramme de cas d'utilisation.

L'exemple ci-contre illustre le concept d'héritage sur l'exemple des différents moyens de transport. Plusieurs niveaux d'abstraction sont donnés :

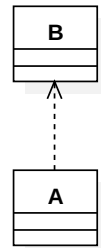
- **Vehicule** hérite de **MoyenDeTransport**
- **Moto** et **Voiture** héritent de **Vehicule** (et donc, par transitivité, héritent de **MoyenDeTransport**)



Dépendance

Une dépendance est une relation unidirectionnelle signifiant une liaison sémantique entre deux classes. Cette relation est plus faible que les associations et la généralisation. Elle indique qu'une modification de la classe cible *pourrait* entraîner la modification de la classe cible. Dans la figure ci-contre on indique une dépendance de la classe A vers la classe B. On dit dans cas qu'un objet de type A "utilise" un objet de type B.

Toutes les relations possibles entre les classes sont en fait des dépendances.



Conventions de nommage en UML

Les noms des attributs, rôles et opérations commencent toujours par une minuscule. Les noms des classes commencent systématiquement par une majuscule. Il est préférable de ne pas utiliser d'accents ni de caractères spéciaux.

Exercice 1 - échauffement

1. Développer et tester le modèle de données suivant en respectant au mieux les fiches du cours de *développement efficace*.
2. Pour chaque exemple ci-dessous, dessiner le diagramme de classes correspondant
 - Dans le cadre d'un système de vente en ligne les livres et les CDs sont des articles. Les clients peuvent commander des articles.
 - Une compagnie aérienne propose des vols et un vol peut être proposé par différentes compagnies.
 - D'après les textes de loi en 2018, en France deux personnes peuvent être mariées et une personne ne peut pas être mariée à deux autres personnes en même temps.
 - Voici deux classes écrites en C++ :

```
class Document
{
public:
    void imprimer(Imprimante &Out);
};
```

```
class Imprimante
{
public:
    string getNom() { return nom; }
};
```

Donnez le diagramme de classes correspondant.

Exercice 2 - Simulation d'un réseau ferroviaire

Vous êtes chargé de créer un prototype de simulation simplifié d'un système de réseau ferroviaire du pays. Des lignes ferroviaires comportent au moins deux gares (correspondants aux extrémités de la ligne) et chaque gare a au moins une ligne ferroviaire qui la traverse. Les gares ont un nom. Les trains utilisent les lignes ferroviaires. À chaque train on affecte une gare d'origine et une gare d'arrivée, des propriétés qui sont modifiables. Chaque ligne ferroviaire possède des capteurs qui détectent les mouvements sur les lignes et des barrières. Chaque barrière peut être ouverte ou fermée (on va supposer des variables booléennes) et à chaque barrière un capteur est associé.

1. Proposez un diagramme de classes

Exercice 3 - Système de réservation de vol

Sujet adapté d'une étude de cas de "UML 2 par la pratique" de Pascal Roques

Cette étude de cas concerne un système simplifié de réservation de vols pour une agence de voyages. Les interviews des experts métier auxquelles vous avez procédé ont permis de résumer leur connaissance du domaine sous la forme des phrases suivantes :

- Des compagnies aériennes proposent différents vols.
- Un vol est ouvert à la réservation et refermé sur ordre de la compagnie.
- Un client peut réserver un ou plusieurs vols pour un ou plusieurs passagers.
- Une réservation concerne un seul vol et un seul passager.
- Une réservation peut être annulée ou confirmée.
- Un vol a un aéroport de départ et un aéroport d'arrivée.
- Un vol a une date et une heure de départ, ainsi qu'une date et une heure d'arrivée.
- La durée d'un vol est calculée à partir des heures de départ et d'arrivée respectives.
- Chaque aéroport dessert une ou plusieurs villes.

Vous devez proposer un diagramme de classes modélisant ce système.

1. Définissez les entités (classes) du modèle.
2. Déterminez les associations entre ces entités.
3. Ajoutez les attributs et les opérations et raffinez votre modèle si besoin.

Exercice 4 - Ordres de missions

On vous demande de modéliser le fonctionnement d'un logiciel de saisie d'ordres de mission et de remboursement des frais de mission. Afin de pouvoir partir en mission d'affaires (payée donc par l'entreprise) les salariés doivent saisir un ordre de mission qui doit être validé par un responsable hiérarchique. Pour simplifier, on peut supposer qu'un salarié n'a qu'un responsable hiérarchique et que le patron de l'entreprise n'a pas de responsable.

Un ordre de mission peut concerner plusieurs trajets. Chaque trajet ayant un but précis, une date, etc. Ainsi, après avoir effectué les trajets en question, le salarié peut faire une demande de remboursement des frais de mission en saisissant les documents justificatifs nécessaires. Cette demande est sauvegardée. Ainsi le service comptabilité de l'entreprise rembourse les frais qui ont été approuvés et qui sont conformes aux règles de l'entreprise (pas de dépassement des forfaits pour l'hôtel, la restauration, etc.). Le salarié se fait rembourser également les frais de transport. Si le salarié effectue le déplacement avec un véhicule personnel, il doit saisir toutes les données nécessaires dans le logiciel (la marque, l'immatriculation, puissance fiscale). Il se peut que deux salariés soit en couple et ainsi partagent la même voiture. Le remboursement est également sauvegardée dans le système.

1. Construire le diagramme de cas d'utilisation.
2. Proposez un diagramme de classes simplifié i.e. sans forcément indiquer tous les attributs, ni toutes les méthodes :
 - (a) Identifiez les classes
 - (b) Établissez les relations entre les classes