

UML

Sommaire

Sommaire

- ❑ Présentation
- ❑ Cas d'utilisation
- ❑ Diagramme d'activités
- ❑ Diagramme de classes
- ❑ Diagramme de séquences

Présentation

Présentation - Définition

- ❑ Qu'est ce qu'UML ?
 - **Unified Modeling Language**, un langage graphique de modélisation des données et des traitements.
 - Présente 13 diagrammes interagissant dans les processus d'analyse et de conception d'un projet de développement informatique.

Présentation - Définition

- ❑ But :
 - visualiser de façon claire une application avant son développement.
 - rôle de documentation : comprendre le fonctionnement d'une application.

Présentation - Différents diagrammes

- ❑ Diagrammes de structure
 - **diagramme de classes**
 - diagramme d'objets
 - diagramme de composants
 - diagramme de structure composite
 - diagramme d'ensemble
 - diagramme de déploiement
 - diagramme de profil

Présentation - Différents diagrammes

- ❑ Les diagrammes de comportement
 - **diagramme de cas d'utilisation**
 - **diagramme d'activité**
 - diagramme d'état-transition
 - **diagramme de séquence**
 - diagramme de communication
 - diagramme de temps
 - diagramme d'aperçu d'interaction

Cas d'utilisation

Cas d'utilisation

❑ Introduction

- Intervient dans la phase d'analyse des besoins du client
- But : établir le cahier des charges fonctionnel
- Trois questions :
 - Définir les **utilisations principales** du système : A quoi sert-il ?
 - Définir l'**environnement** du système : Qui va l'utiliser et interagir avec lui ?
 - Définir la **limite** du système : Où s'arrête sa responsabilité ?

Cas d'utilisation - Introduction

- Éléments de description :
 - **Description textuelle** des cas d'utilisation
 - **Diagramme** de cas d'utilisation

Cas d'utilisation

❑ Diagramme de cas d'utilisation

○ Éléments de base :

○ Acteur : Entité qui interagit avec le système

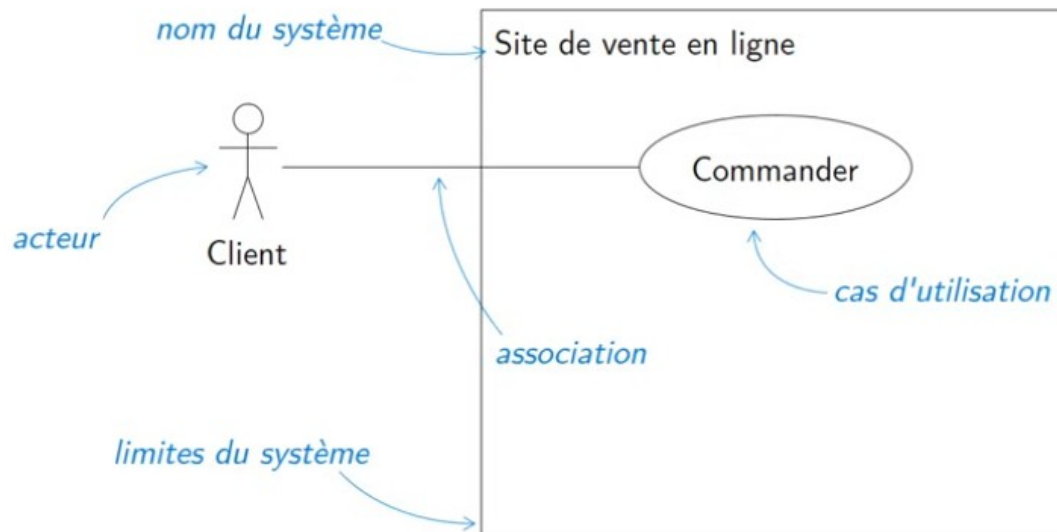
- Personne, logiciel, **extérieur au système** décrit
- Représente un **rôle** (plusieurs rôles possibles pour une même entité)
- Identifié par un **nom de rôle**

○ Cas d'utilisation : Fonctionnalité principale visible de l'extérieur

- Action **déclenchée** par un acteur
- Identifié par une action (verbe à l'infinitif)

>> Vision du système centrée sur l'utilisateur ≠ comportement interne du système

Cas d'utilisation - Diagramme



○ Association :

- lien entre l'**acteur** et le **cas d'utilisation**
- Représente la possibilité de l'acteur de **déclencher** le cas

Cas d'utilisation - Exercice 1

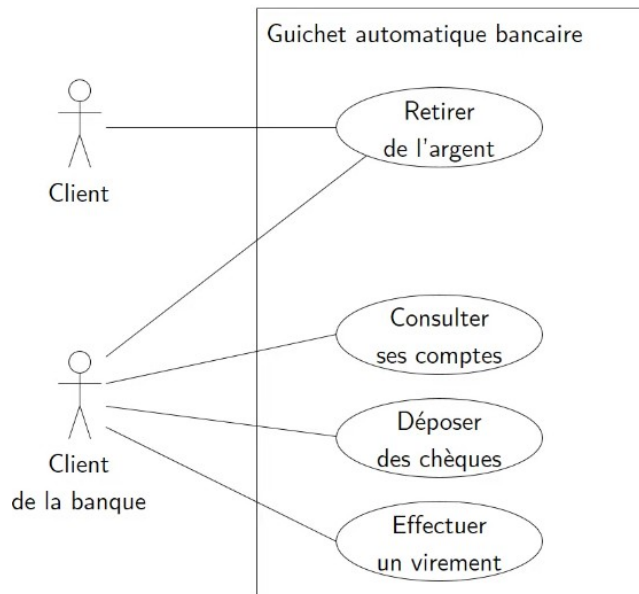
□ Identification des acteurs et de cas d'utilisation simples

Considérons une station-service de distribution d'essence. Les clients se servent de l'essence et le pompiste remplit les cuves.

- Question 1 : Le client se sert de l'essence de la façon suivante : il prend un pistolet accroché à une pompe et appuie sur la gâchette pour prendre de l'essence. Qui est l'acteur du système ? Est-ce le client, le pistolet ou la gâchette ?
- Question 2 : Jojo, dont le métier est pompiste, peut se servir de l'essence pour sa voiture. Pour modéliser cette activité de Jojo, doit-on définir un nouvel acteur ? Comment modélise-t-on ça ?
- Question 3 : Lorsque Jojo vient avec son camion citerne pour remplir les réservoirs des pompes, est-il considéré comme un nouvel acteur ? Comment modélise-t-on cela ?
- Question 4 : Certains pompistes sont aussi qualifiés pour opérer des opérations de maintenance en plus des opérations habituelles des pompistes telles que le remplissage des réservoirs. Ils sont donc réparateurs en plus d'être pompistes. Comment modéliser cela ?

Cas d'utilisation - Diagramme

- Généralisation des rôles
 - Cas d'une même fonctionnalité utilisé par plusieurs acteurs



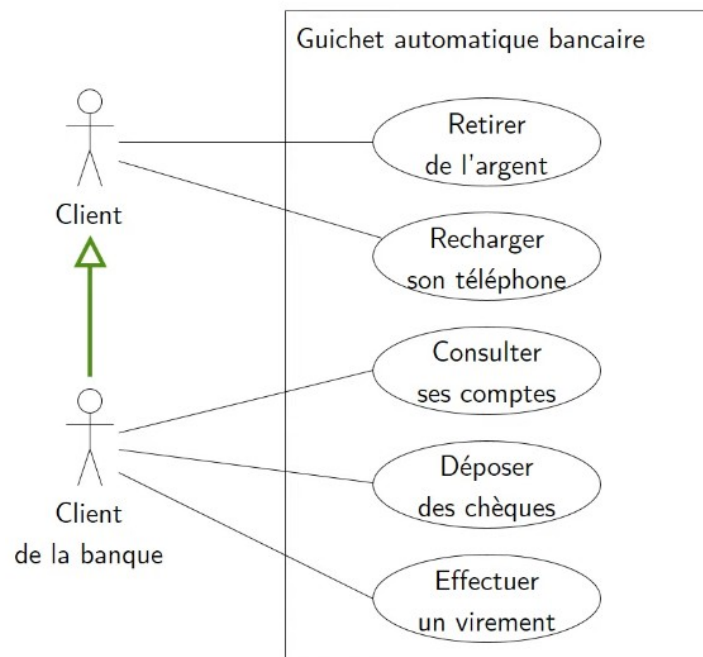
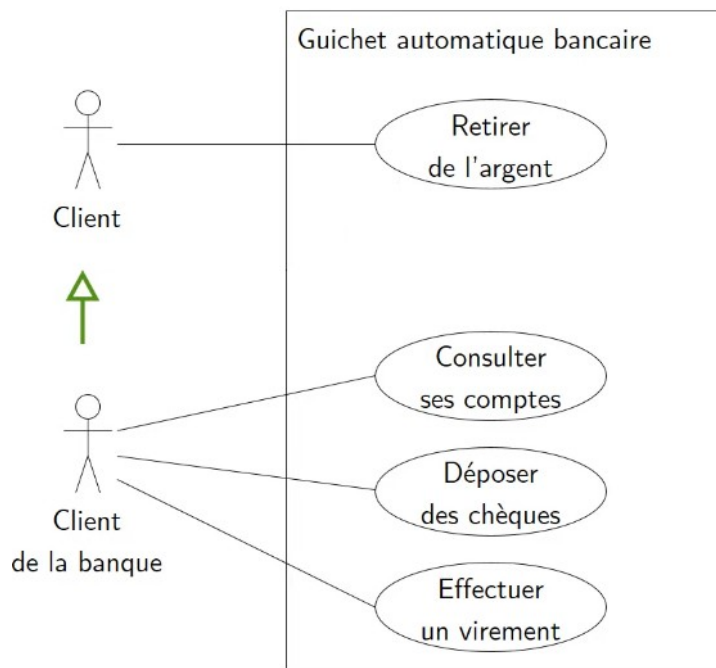
Cas d'utilisation - Diagramme

- Généralisation des rôles
 - Cas d'une même fonctionnalité utilisé par plusieurs acteurs
 - Situation : le client de la banque peut faire tout ce que fait le client
 - Problématique : le diagramme peut devenir illisible lors d'ajout d'association entre acteurs et cas d'utilisation
 - Solution : Faire apparaître une relation d'héritage entre les acteurs.
Dans notre exemple, un client de la banque peut retirer de l'argent, consulter ses comptes, déposer des chèques et effectuer un virement.

Le client de la banque devient donc un cas particulier du client.

Cas d'utilisation - Diagramme

- Généralisation des rôles
 - Cas d'une même fonctionnalité utilisé par plusieurs acteurs



Cas d'utilisation - Exercice 1

- ❑ Identification des acteurs et de cas d'utilisation simples (suite)

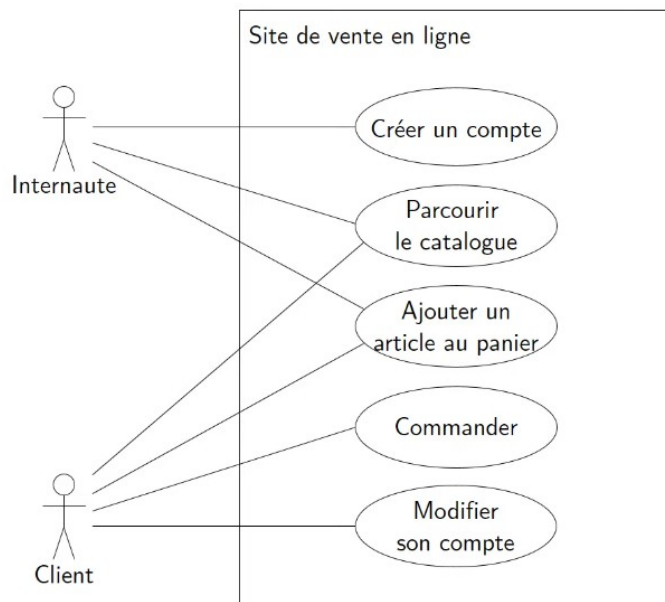
Considérons une station-service de distribution d'essence. Les clients se servent de l'essence et le pompiste remplit les cuves.

- Question 4 : Certains pompistes sont aussi qualifiés pour opérer des opérations de maintenance en plus des opérations habituelles des pompistes telles que le remplissage des réservoirs. Ils sont donc réparateurs en plus d'être pompistes. Comment modéliser cela ?

Cas d'utilisation - Diagramme

□ Généralisation des rôles

- Cas de plusieurs acteurs utilisant les mêmes fonctionnalités mais ayant des cas spécifiques



Cas d'utilisation - Diagramme

❑ Généralisation des rôles

- Cas de plusieurs acteurs utilisant les mêmes fonctionnalités mais ayant des cas spécifiques

- Problématique : le diagramme peut devenir illisible lors d'ajout d'association entre acteurs et cas d'utilisation
- Solution : utiliser l'héritage d'acteurs en créant un acteur intermédiaire qui va rassembler les cas communs et dont va hériter chacun des acteurs de départ.

Dans l'exemple, internaute et client peuvent tous les deux parcourir le catalogue et ajouter un article au panier.

L'internaute peut créer un compte pour devenir un client.

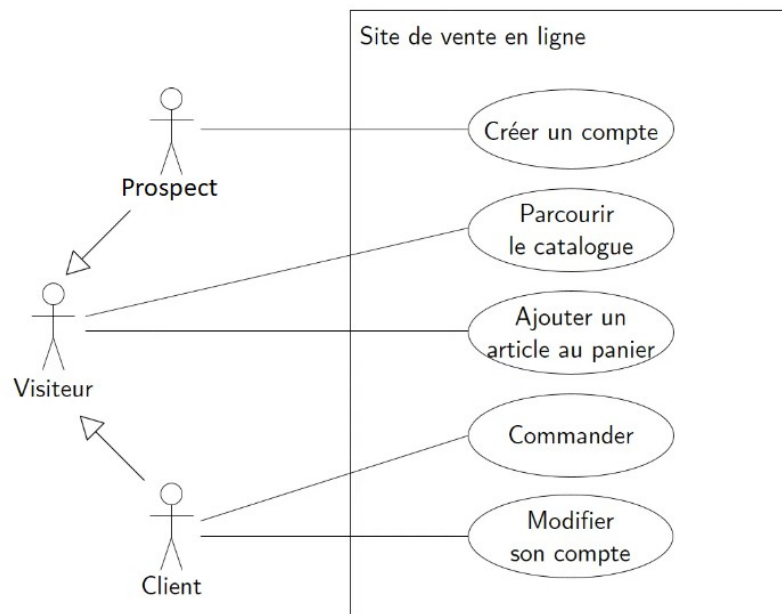
Un client peut modifier son compte et passer une commande.

Le client n'hérite pas d'internaute car il ne peut pas créer de compte, il en a déjà un.

Cas d'utilisation - Diagramme

□ Généralisation des rôles

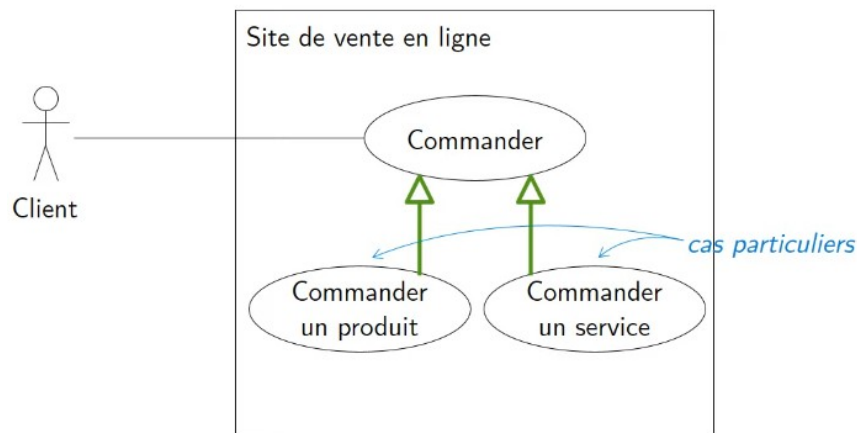
- Cas de plusieurs acteurs utilisant les mêmes fonctionnalités mais ayant des cas spécifiques



Cas d'utilisation - Diagramme

❑ Relation entre cas d'utilisation

- Héritage : un cas d'utilisation regroupe plusieurs cas particuliers



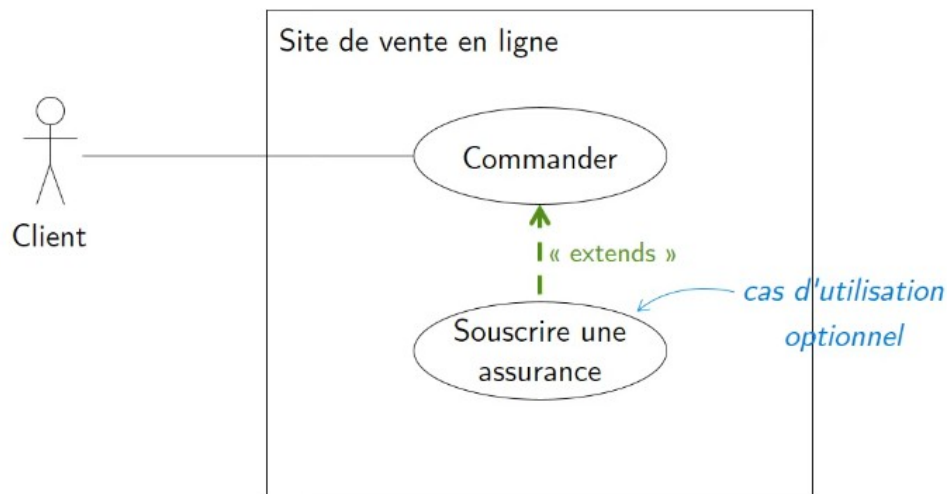
Dans cet exemple, commander un produit et commander un service ont des scénarios communs mais aussi différents notamment pour la livraison qui n'existe pas pour un service.

On a donc un cas générique « commander » dont héritent « commander un service » et « commander un produit » qui sont des cas particuliers.

Cas d'utilisation - Diagramme

❑ Relation entre cas d'utilisation

- Extension : un cas d'utilisation est une partie optionnelle d'un autre cas.

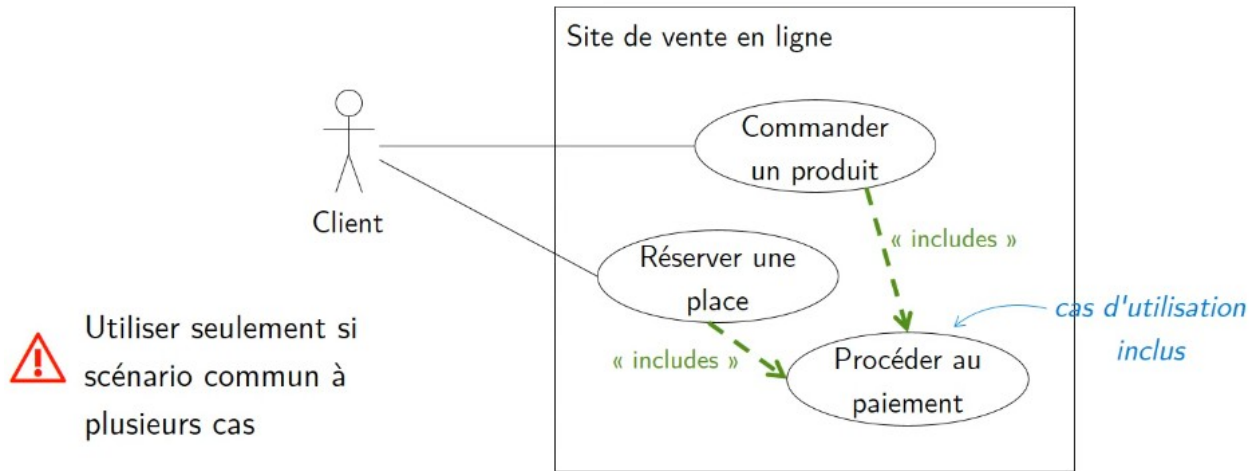


Ici, on peut commander en prenant une assurance ou non. On dit que « souscrire une assurance » étend de « commander ». « Souscrire une assurance » ne peut pas vivre indépendamment de « commander ».

Cas d'utilisation - Diagramme

❑ Relation entre cas d'utilisation

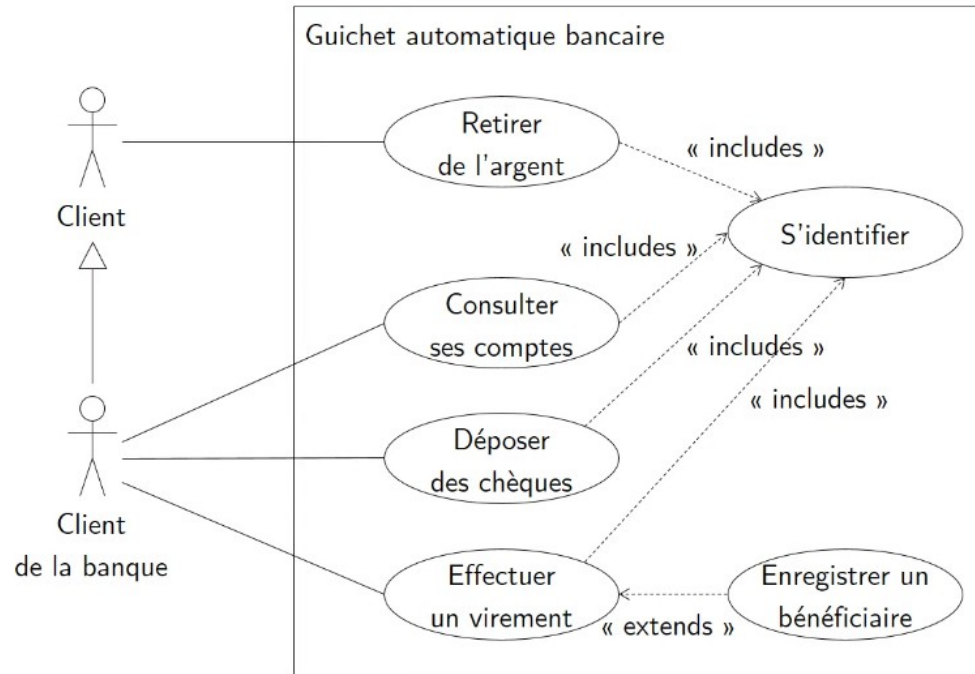
- Inclusion : un cas d'utilisation est une partie obligatoire d'un autre cas.



Ici, Commander un produit et réserver une place sont deux scénarios différents mais incluent un même procédé pour régler la commande. On dit qu'ils incluent tous les deux le scénario « Procéder au paiement ».

Cas d'utilisation - Diagramme

■ Relation entre cas d'utilisation



Cas d'utilisation – Exercice 2

- Considérons les cas d'utilisation suivants : * Expédition commande * Expédition complète * Expédition partielle * Gestion de commande * Passer commande * Passer commande urgente * Identification utilisateur

La gestion de la commande désigne le processus complet, du passage à l'expédition. Il peut toutefois arriver qu'une commande passée ne soit pas envoyée. Passer une commande urgente est un cas particulier de passer une commande. Pour passer une commande, il faut nécessairement valider l'utilisateur.

Proposez un diagramme de cas d'utilisation sans représenter les acteurs.

Cas d'utilisation – Exercice 3

- On cherche à modéliser un distributeur automatique de billets (DAB). Ce distributeur sera utilisé par des clients qui veulent pouvoir choisir une opération parmi le retrait d'argent (rapide ou normal) et la consultation du solde de leur compte. Pour chaque opération, il faudra s'être identifié.

Le distributeur devra permettre d'éditer des tickets pour chaque opération si l'utilisateur le souhaite. Un système central extérieur permettra de vérifier le solde des comptes dans le cas d'un retrait important, et de fournir les informations nécessaires dans le cas d'une consultation.

Proposez un diagramme de cas d'utilisation pour modéliser les fonctionnalités attendues du système DAB.

Cas d'utilisation – Exercice 4

- Une bibliothèque universitaire souhaite automatiser sa gestion. Cette bibliothèque est gérée par un gestionnaire chargé des inscriptions et des relances des lecteurs quand ceux-ci n'ont pas rendu leurs ouvrages au-delà du délai autorisé. Les bibliothécaires sont chargés de gérer les emprunts et la restitution des ouvrages ainsi que l'acquisition de nouveaux ouvrages.

Il existe trois catégories d'abonné. Tout d'abord les étudiants qui doivent seulement s'acquitter d'une somme forfaitaire pour une année afin d'avoir droit à tous les services de la bibliothèque. L'accès à la bibliothèque est libre pour tous les enseignants. Enfin, il est possible d'autoriser des étudiants d'une autre université à s'inscrire exceptionnellement comme abonné moyennant le versement d'une cotisation. Le nombre d'abonné externe est limité chaque année à environ 10 % des inscrits.

Un nouveau service de consultation du catalogue général des ouvrages doit être mis en place.

Les ouvrages, souvent acquis en plusieurs exemplaires, sont rangés dans des rayons de la bibliothèque.

Chaque exemplaire est repéré par une référence gérée dans le catalogue et le code du rayon où il est rangé.

Chaque abonné ne peut emprunter plus de trois ouvrages. Le délai d'emprunt d'un ouvrage est de trois semaines, il peut cependant être prolongé exceptionnellement à cinq semaines.

Élaborer le diagramme des cas d'utilisation

Cas d'utilisation - Diagramme

❑ Conseils :

- Restez lisible :
 - Pas plus de 6 ou 8 cas par diagramme
 - Au besoin, faire plusieurs diagrammes (si cas disjoints entre acteurs ou pour détailler un cas)
 - Relations entre cas seulement si nécessaires
- Pour les détails, privilégiez la description textuelle.

Cas d'utilisation - Diagramme

❑ Conclusion

- **Utiles** pour la discussion avec le client car **intuitifs et concis**
- **Insuffisants** pour l'équipe de développement
- Nécessité d'**une description détaillée** des scénarios représentés par chacun des cas :
 - Description textuelle en **langue naturelle structurée**
 - **Explication du vocabulaire** utilisé dans les diagrammes

Cas d'utilisation

❑ Scénario d'utilisation

○ But :

- Pour chaque cas d'utilisation, décrire l'ensemble des scénarios qu'il représente.

○ Champs à renseigner :

- Nom du cas d'utilisation
- Acteur(s)
- Contexte (comment le scénario est déclenché ?)
- Données en entrée et pré-conditions
- Données en sortie et pré-conditions
- Scénario principal pour ce cas d'utilisation : étapes à suivre pour réaliser ce cas
- Variantes, cas d'erreur : déviations du scénarios principal, scénarios alternatifs, scénarios d'erreur.

Cas d'utilisation – Scénario d'utilisation

- ❑ Exemple : commande d'un client sur un site de vente en ligne
 - Nom : commander
 - Acteur : client
 - Contexte : le cas d'utilisation commence à partir du moment où le client clique sur « commander »
 - Données d'entrée : produits sélectionnés par le client

Cas d'utilisation – Scénario d'utilisation

- ❑ Exemple : commande d'un client sur un site de vente en ligne
 - Scénario principal :
 1. Le système demande au client de saisir son identifiant et son mot de passe.
 2. Le client saisit son identifiant et son mot de passe valides.
 3. Le système demande au client de choisir une adresse de livraison parmi sa liste d'adresses ou d'en saisir une nouvelle.
 4. Le client choisit une adresse de livraison et valide.
 5. Le système demande au client de choisir un mode d'expédition parmi une liste prédéfinie (Colissimo ou point relais).
 6. Le client choisit un mode d'expédition et valide.

Cas d'utilisation – Scénario d'utilisation

- ❑ Exemple : commande d'un client sur un site de vente en ligne
 - Scénario principal (suite) :
 7. Le système affiche un récapitulatif de la commande, indique le montant total avec le coût de livraison et demande au client de choisir un mode de paiement parmi une liste prédéfinie (Paypal, carte bancaire, Google Pay)
 8. Le client choisit un mode de paiement et valide.
 9. Le système demande au client de saisir les informations de paiement.
 10. Le client saisit les informations de paiement et valide.
 11. Le système informe le client que la transaction a bien été effectuée et un e-mail récapitulatif de la commande est envoyé au client.

Cas d'utilisation – Scénario d'utilisation

- Exemple : commande d'un client sur un site de vente en ligne
 - **Scénario d'erreur** : Client inconnu
 - 3a. Le client n'est pas connu du système. Le système affiche un message d'erreur.
Retour à l'étape 1.
 - **Scénario alternatif** : Nouvelle adresse de livraison
 - 4a. Le client saisit une nouvelle adresse de livraison et valide
Le scénario reprend à l'étape 5
 - **Scénario alternatif** : Modifications des choix de livraison
 - 8a. Le client demande à modifier son adresse de livraison.
Retour à l'étape 3.
 - 8b. Le client demande à modifier le mode de livraison.
Retour à l'étape 5.
 - **Scénario d'erreur** : Transaction impossible
 - 11a. Le système informe le client que ses informations de paiement sont incorrectes.
Retour à l'étape 9

Cas d'utilisation - Exercice 5

- Cet exercice concerne un système simplifié de caisse enregistreuse de super- marché. Le déroulement normal d'utilisation de la caisse est le suivant :

Un client arrive à la caisse avec des articles à payer

Le caissier enregistre le numéro d'identification (CPU) de chaque article, ainsi que la quantité si elle est supérieure à 1.

La caisse affiche le prix de chaque article et son libellé.

Lorsque tous les achats sont enregistrés, le caissier signale la fin de la vente.

La caisse affiche le total des achats.

Le client choisit son mode de paiement :

- numéraire : le caissier encaisse l'argent reçu, la caisse indique l'argent à rendre au client.
- chèque : le caissier vérifie la solvabilité du client en transmettant une requête à un centre d'autorisation via la caisse.
- carte de crédit : un terminal bancaire fait partie de la caisse. Il transmet une demande d'autorisation à un centre d'autorisation en fonction du type de la carte.

Cas d'utilisation - Exercice 5

La caisse enregistre la vente et imprime un ticket.

Le caissier donne le ticket de caisse au client.

Après la saisie des articles, le client peut présenter au caissier des coupons de réduction pour certains articles. Lorsque le paiement est terminé, la caisse transmet les informations sur le nombre d'articles vendus au système de gestion des stocks.

Tous les matins, le responsable du magasin initialise les caisses pour la journée.

- ❑ Construire un diagramme de cas d'utilisation détaillé de la caisse enregistreuse.
- ❑ Décrire textuellement le cas d'utilisation principal « Traiter le passage en caisse » sans détailler les cas d'utilisation spécialisés correspondant aux différents cas de règlement.

Diagramme d'activité

Diagramme d'activité

■ Introduction

- Décrit un certain cas d'utilisation à un niveau plus détaillé.
- Mise en évidence de l'organisation des activités
- Représente à la fois les comportements conditionnels et parallèles
- Indique ce qu'il se passe mais pas qui fait quoi : aucune classe n'est désignée.
- Algorithme visuel

Diagramme d'activité

❑ Réalisation du diagramme

Pour illustrer ce cours, nous allons prendre un exemple de description textuelle d'un cas d'utilisation :

- Cas d'utilisation : Créer un utilisateur
- Acteur : administrateur
- Contexte : l'administrateur clique sur le bouton « créer »

Diagramme d'activité - réalisation

- Scénario principal :
 1. L'application demande à l'administrateur de saisir un login
 2. L'administrateur saisit un login
 3. L'application demande à l'administrateur de saisir un mot de passe
 4. L'administrateur saisit un mot de passe et valide.
 5. L'application informe que l'utilisateur a bien été créé

Diagramme d'activité - réalisation

- **Scénario d'erreur : Login incorrect**
 - 3a. L'administrateur n'a pas saisi de login. L'application en informe l'administrateur.
Aller au point 1.
 - 3b. L'administrateur n'a pas saisi un login correct. Il doit comporter seulement des chiffres et des lettres. L'application en informe l'administrateur.
Aller au point 1.
- **Scénario d'erreur : Mot de passe incorrect**
 - 5a. L'administrateur n'a pas saisi de mot de passe. L'application en informe l'administrateur.
Aller au point 1.
 - 5b. L'administrateur a saisi un mot de passe incorrect. Il doit contenir au moins une majuscule, 8 caractères, un caractère spécial et un chiffre. L'application en informe l'administrateur.
Aller au point 1.

Diagramme d'activité - réalisation

- **Scénario d'erreur** : Échec d'enregistrement de l'utilisateur
 - 6a. Un utilisateur avec le même login existe déjà.
L'application en informe l'administrateur.
Retour au point 1.

Diagramme d'activité - réalisation

□ Étapes de réalisation du diagramme

○ Identification des rôles :

○ Qui intervient et qui fait quoi ?

Dans notre exemple, c'est l'administrateur et l'application :

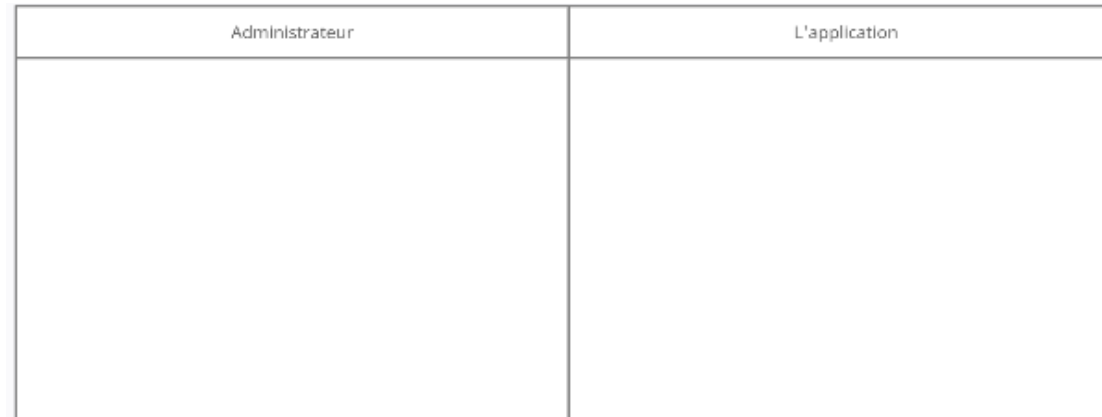


Diagramme d'activité - réalisation

□ Étapes de réalisation du diagramme

- Placement des actions principales :
 - Les actions sont les points listés dans le scénario principal, d'erreur et alternatif de la description textuelle.
 - Il faut dans un premier temps, représenter les actions du scénario principal :

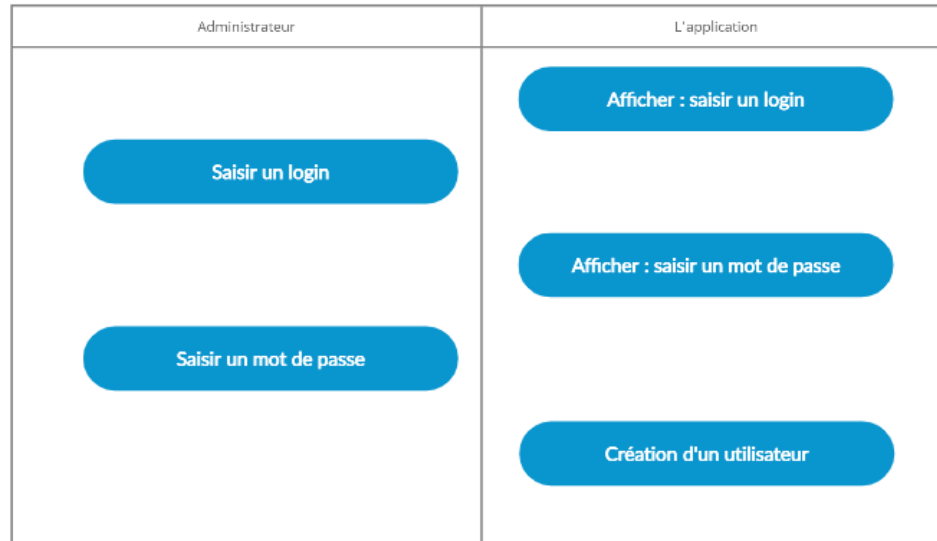


Diagramme d'activité - réalisation

- ❑ Étapes de réalisation du diagramme
 - Transition séquentielle :
 - Permet de préciser de façon visuelle l'enchaînement des actions

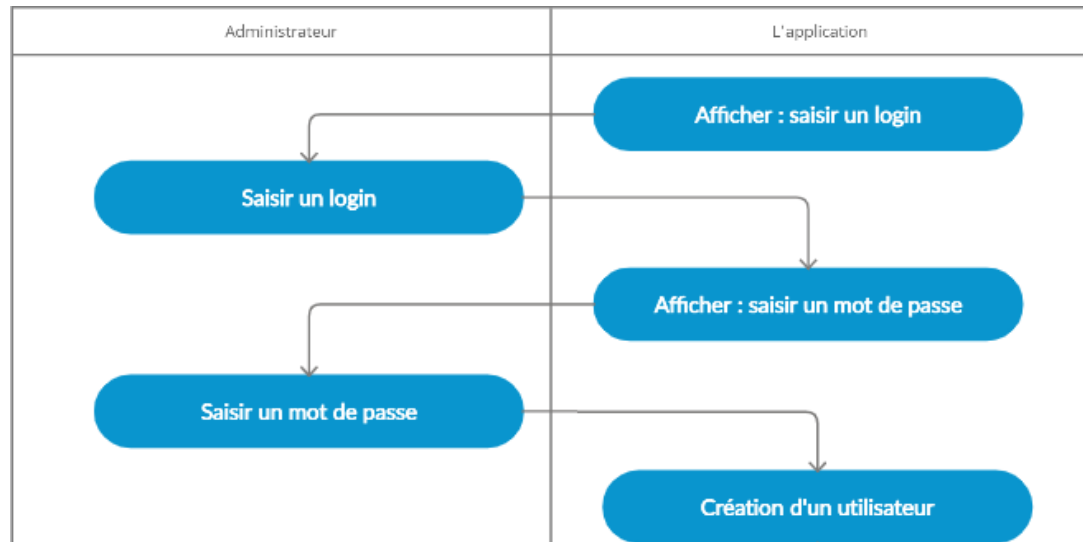


Diagramme d'activité - réalisation

□ Étapes de réalisation du diagramme

○ Branchement :

- Permet de définir une transition entrante gardée par une condition et plusieurs transitions sortantes mutuellement exclusives.

Dans notre exemple, on va pouvoir ajouter les actions et transitions du premier scénario d'erreur.

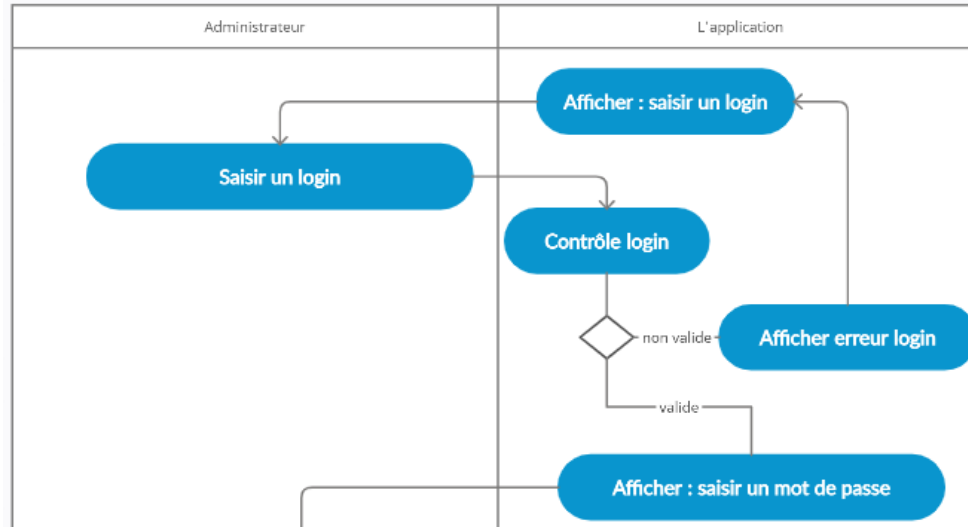
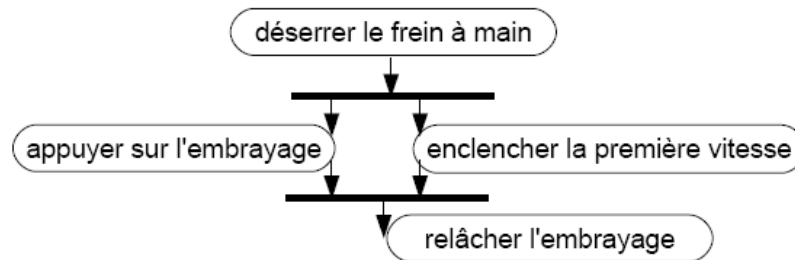


Diagramme d'activité - réalisation

□ Étapes de réalisation du diagramme

○ Comportement parallèle

- La barre de synchronisation permet d'ouvrir et de fermer des branches parallèles au sein du flot d'exécution
- Les transitions partant d'une barre ont lieu en même temps
- La barre n'est franchie qu'après réalisation de toutes les transitions qui s'y rattachent



○ Fusion

- Plusieurs transitions entrantes et une seule sortante
- Marque la fin d'un comportement

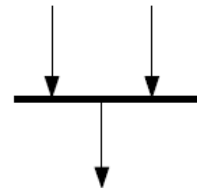
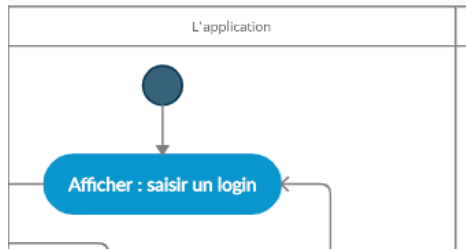


Diagramme d'activité - réalisation

□ Étapes de réalisation du diagramme

○ Nœud initial :

- Permet de définir quand débute l'activité. Une activité peut avoir plusieurs nœuds initiaux. Un nœud initial possède un arc sortant et pas d'arc entrant :



○ Nœud final :

- Marque l'état final d'une activité :



Diagramme d'activité – Exercice 6

- ❑ Finir le diagramme d'activité montré en exemple dans le cours.

Diagramme d'activité – Exercice 7

- ❑ Soit le scénario d'utilisation suivant :

Commande d'un client sur un site de vente en ligne

- Nom : commander
- Acteur : client
- Contexte : le cas d'utilisation commence à partir du moment où le client clique sur « commander »
- Données d'entrée : produits sélectionnés par le client

Diagramme d'activité – Exercice 7



- Scénario principal :

1. Le système demande au client de saisir son identifiant et son mot de passe.
2. Le client saisit son identifiant et son mot de passe valides.
3. Le système demande au client de choisir une adresse de livraison parmi sa liste d'adresses ou d'en saisir une nouvelle.
4. Le client choisit une adresse de livraison et valide.
5. Le système demande au client de choisir un mode d'expédition parmi une liste prédéfinie (Colissimo ou point relais).
6. Le client choisit un mode d'expédition et valide.

Diagramme d'activité – Exercice 7



- Scénario principal (suite) :

7. Le système affiche un récapitulatif de la commande, indique le montant total avec le coût de livraison et demande au client de choisir un mode de paiement parmi une liste prédéfinie (Paypal, carte bancaire, Google Pay)
8. Le client choisit un mode de paiement et valide.
9. Le système demande au client de saisir les informations de paiement.
10. Le client saisit les informations de paiement et valide.
11. Le système informe le client que la transaction a bien été effectuée et un e-mail récapitulatif de la commande est envoyé au client.

Diagramme d'activité – Exercice 7

- **Scénario d'erreur** : Client inconnu
 - 3a. Le client n'est pas connu du système. Le système affiche un message d'erreur.
Retour à l'étape 1.
- **Scénario alternatif** : Nouvelle adresse de livraison
 - 4a. Le client saisit une nouvelle adresse de livraison et valide
Le scénario reprend à l'étape 5
- **Scénario alternatif** : Modifications des choix de livraison
 - 8a. Le client demande à modifier son adresse de livraison.
Retour à l'étape 3.
 - 8b. Le client demande à modifier le mode de livraison.
Retour à l'étape 5.
- **Scénario d'erreur** : Transaction impossible
 - 11a. Le système informe le client que ses informations de paiement sont incorrectes.
Retour à l'étape 9

Diagramme d'activité – Exercice 7

- ❑ Réaliser le diagramme d'activité à partir du scénario d'utilisation

Diagramme de classes

Diagramme de classes

Objet

- Entité hermétique décrit par :
 - Identité (adresse mémoire)
 - des données (attributs)
 - des traitements (méthodes)

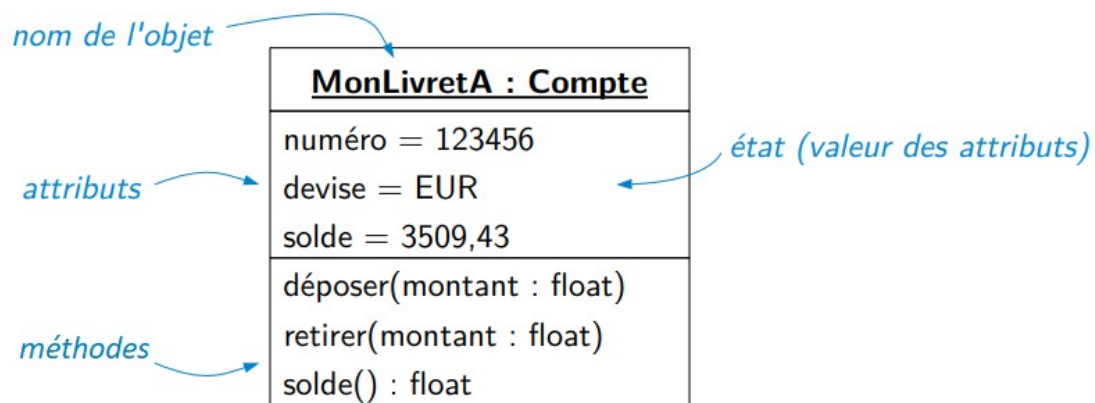


Diagramme de classes

- Classe : moule de fabrication d'objets. Ainsi, on dit que l'objet est une instance de la classe.
 - Convention de nommage : en gras.

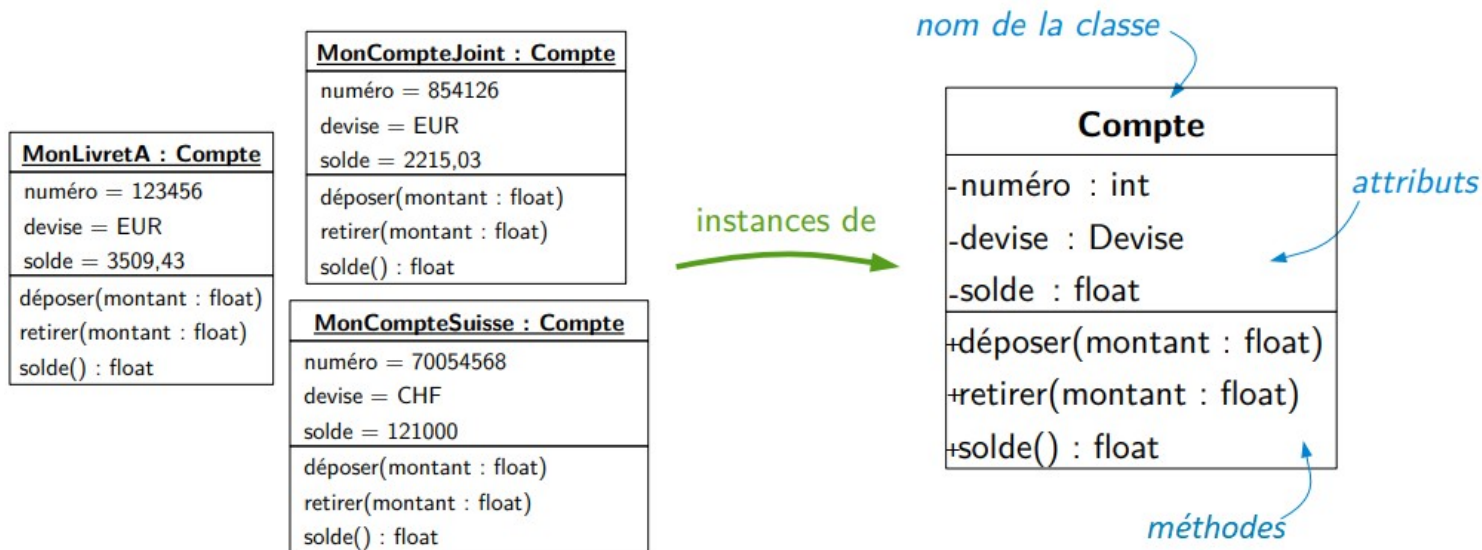


Diagramme de classes - classe

○ Attributs

- Caractéristique partagée par tous les objets de la classe
- Associe à chaque objet une valeur
- Type associé simple (int, bool...), primitif (Date) ou énuméré
- Convention de nommage : « nom_attribut : type »

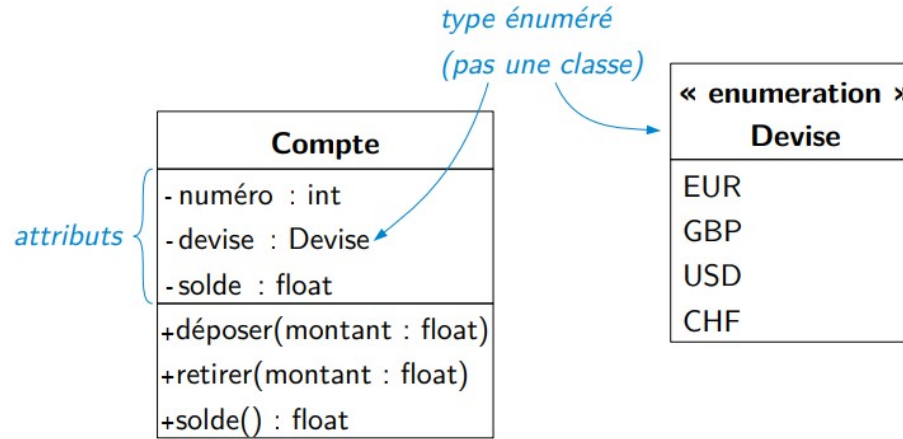


Diagramme de classes - classe

- Attributs statiques
 - Un attribut statique existe dès que sa classe est chargée en mémoire, en dehors et indépendamment de toute instantiation.
 - Convention de nommage : « nom_attribut : type »

Diagramme de classes - classe

- Valeur des attributs : État de l'objet
 - Objets différents (identités différentes) peuvent avoir mêmes attributs

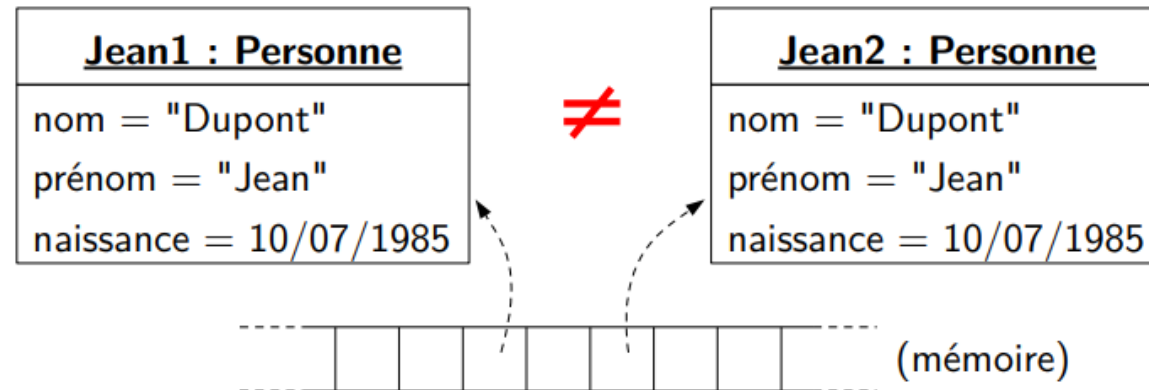


Diagramme de classes - classe

○ Méthodes

○ Différents types :

- Méthode normale : appelée par une instance d'objet
- Méthode abstraite : méthode non implémentée
- Méthode statique : méthode pouvant être appelée sans instance d'objet.

○ Conventions de nommage :

- Méthode normale : sans style
- Méthode abstraite : *italique*
- Méthode statique : souligné

Diagramme de classes - classe

- Méthodes
 - Service qui peut être demandé à tout objet de la classe
 - Comportement commun à tous les objets de la classe

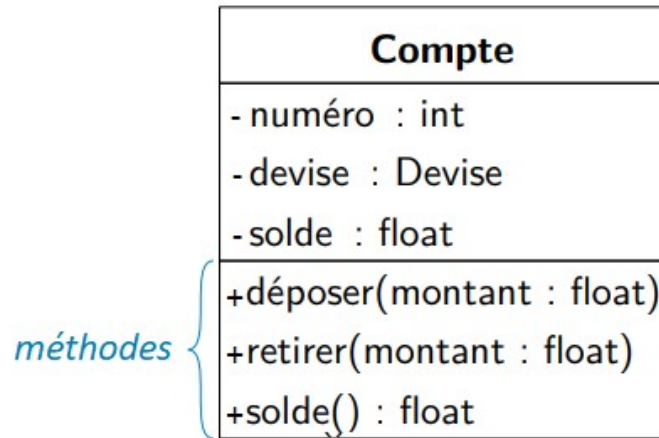


Diagramme de classes - méthode



- Accesseurs

Les accesseurs permettent de modifier ou de lire les attributs de la classe

- Constructeur

Méthode appelée à la construction de l'objet. Elle permet d'initialiser les valeurs des attributs et d'appeler éventuellement des méthodes.

- Destructeur

Méthode appelée pour détruire l'objet.



Les accesseurs, constructeurs et destructeurs sont implicites et donc absents du diagramme de classes.

Diagramme de classes - classe

○ Visibilité

- Convention de nommage : Signe à gauche des membres (attributs et méthodes)
 - - privé : accessibilité seulement dans la classe
 - # protégé : accessibilité dans la classe et les classes dérivées
 - + public : accessibilité depuis n'importe quel script.
- Par défaut, les attributs d'une classe sont privés ou protégés.
Les constantes de classe sont généralement publiques.

Diagramme de classes

Relations entre classes

- Association entre classes : Relation binaire (en général)
- Rôle : Nomme l'extrémité d'une association, permet d'accéder aux objets liés par l'association à un objet donné
- Multiplicité : Contraint le nombre d'objets liés par l'association

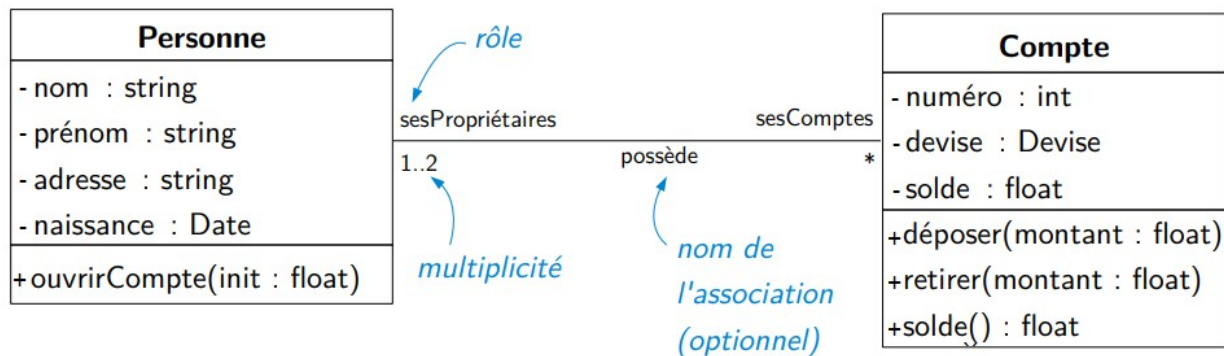


Diagramme de classes - relations

❑ Association et attribut

- **Rappel** : un attribut est de type simple, primitif ou énuméré

On ne met pas d'**attribut dont le type est une classe** du diagramme

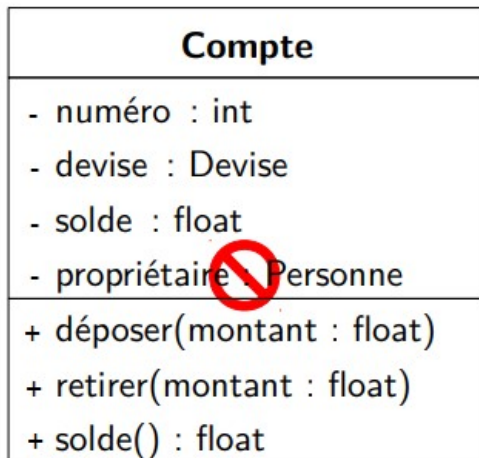


Diagramme de classes - relations

- **Solution** : on met une association vers la classe.

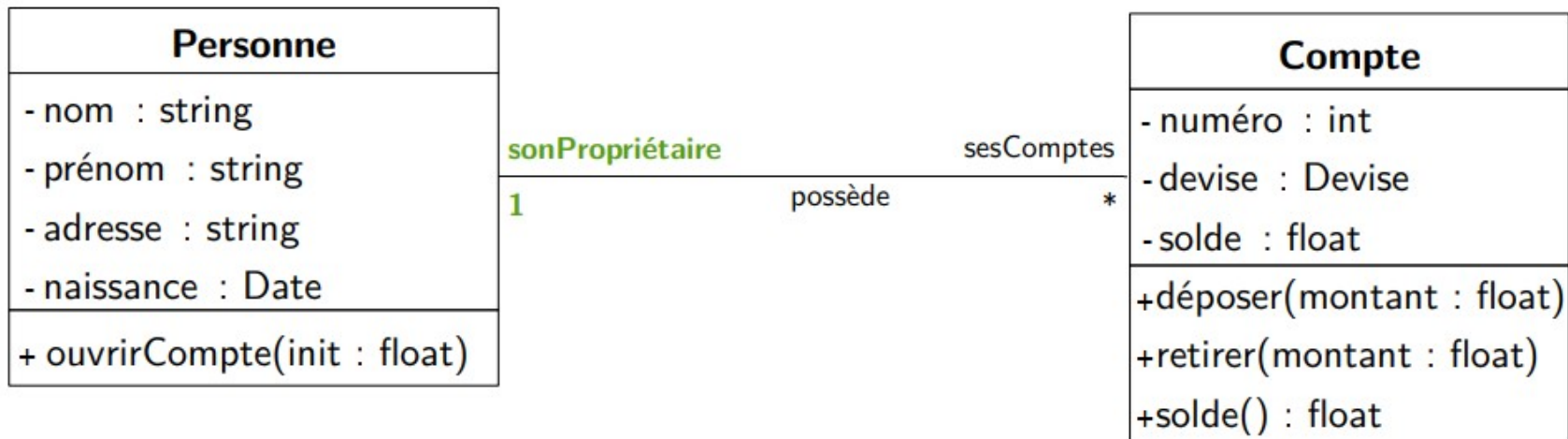


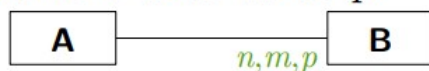
Diagramme de classes - relations

■ Multiplicités

Exactement n



Exactement n ou m ou p



Entre n et m



Au moins n



Plusieurs (0 ou plus)

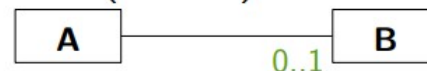


En pratique

Exactement 1



Au plus 1 (0 ou 1)



Au moins 1 (jamais 0)



0 ou plus



Diagramme de classes - relations

❑ Associations réflexives

Une association réflexive est une association qui associe une classe avec elle-même

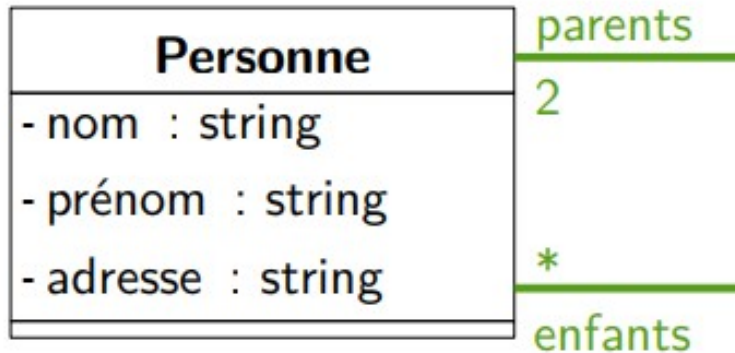


Diagramme de classes - relations

Associations multiples

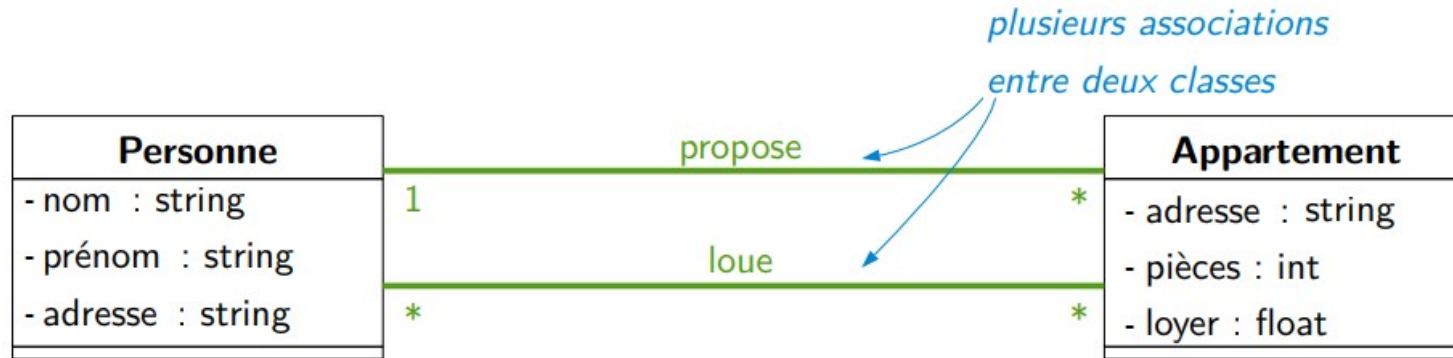


Diagramme de classes - relations

❑ Navigabilité

Orientation d'une association

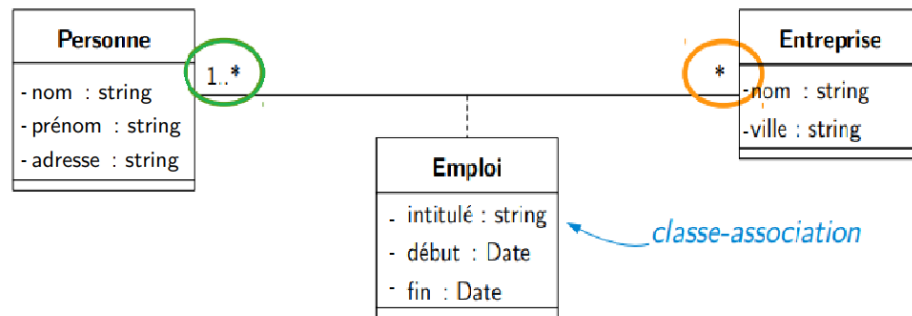
- Restreint l'accessibilité des objets
- Depuis un A, on a accès aux objets de B qui lui sont associés, mais pas l'inverse



Diagramme de classes - relations

❑ Classe-association

Permet de paramétrer une association entre deux classes par une classe.



Instance unique de la classe-association pour chaque lien entre objets

Diagramme de classes - relations

❑ Classe-association

Équivalence en termes de classes et d'associations :

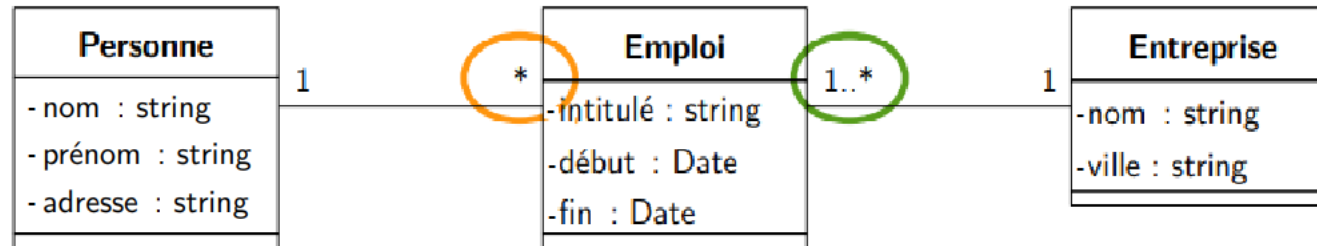


Diagramme de classes - relations

□ Agrégation faible

Agrégation par référence

- Le composite fait référence à ses composants
- La création ou destruction du composite est indépendante de la
- Création ou destruction de ses composants
- Un objet peut faire partie de plusieurs composites à la fois

Exemple

- Une liste de lecture est composée d'un ensemble de morceaux
- Un morceau peut appartenir à plusieurs listes de lecture
- Supprimer la liste ne supprime pas les morceaux

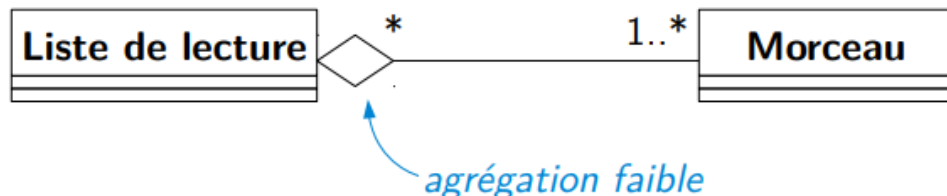


Diagramme de classes - relations

□ Composition

Agrégation par valeur

- Le composite contient ses composants
- La création ou destruction du composite entraîne la création ou
- Destruction de ses composants
- Un objet ne fait partie que d'un composite à la fois

Exemple

- Un morceau n'appartient qu'à un album
- La suppression de l'album entraîne la suppression de tous ses morceaux



Diagramme de classes

■ Hiérarchie de classe

- Principe : Regrouper les classes partageant des attributs et des méthodes et les organiser en arborescence

CompteCourant
- numéro : int
- devise : Devise
- solde : float
- découvertAutorisé : float
- fraisDécouvert : float
- déposer(montant : float)
- retirer(montant : float)
- solde() : float

CompteÉpargne
- numéro : int
- devise : Devise
- solde : float
- plafond : float
- taux : float
- déposer(montant : float)
- retirer(montant : float)
- solde() : float
- calculerIntérêts() : float

Diagramme de classes – hiérarchie de classe

- Spécialisation : raffinement d'une classe en une sous-classe
- Généralisation : abstraction d'un ensemble de classes en super-classe

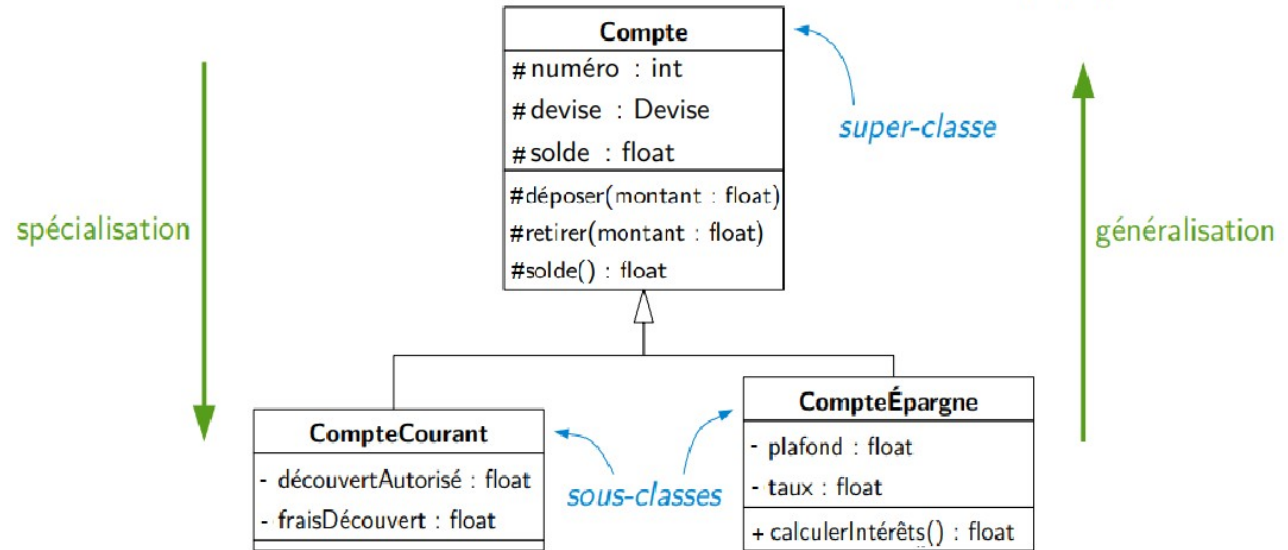


Diagramme de classes – hiérarchie de classe

- Héritage : Construction d'une classe à partir d'une classe plus haute dans la hiérarchie (partage des attributs, opérations, contraintes...)

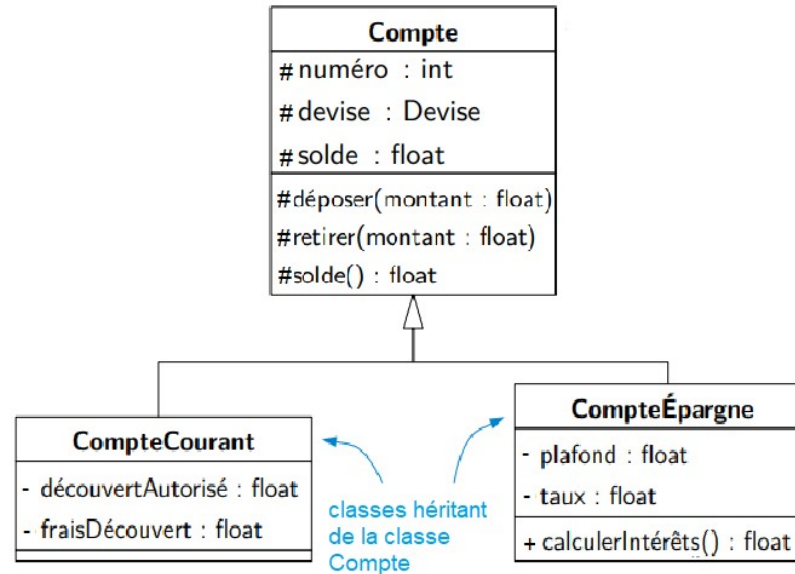


Diagramme de classes – hiérarchie de classe

□ Polymorphisme :

- Il peut exister dans une même hiérarchie de classes des méthodes surchargées.
- Elles n'effectuent pas le même traitement suivant le niveau de la hiérarchie ;
- Ce qui permet d'associer à un même message une méthode différente selon la classe d'appartenance de l'objet.

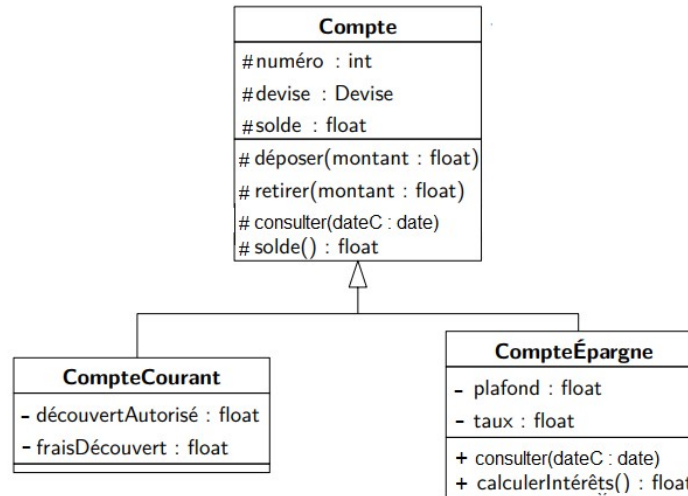


Diagramme de classes – hiérarchie de classe

□ Classe abstraite

- Classe non instanciable, seulement une base pour classes héritées
- Peut contenir des méthodes abstraites ou non.
- Une méthode abstraite est une méthode non implémentée.
- Une classe qui contient une méthode abstraite est elle-même abstraite.
- Les classes utilisant une classe abstraite devront implémentées les méthodes abstraites.
- Règle de nommage : en italique ou avec le mot « abstract »

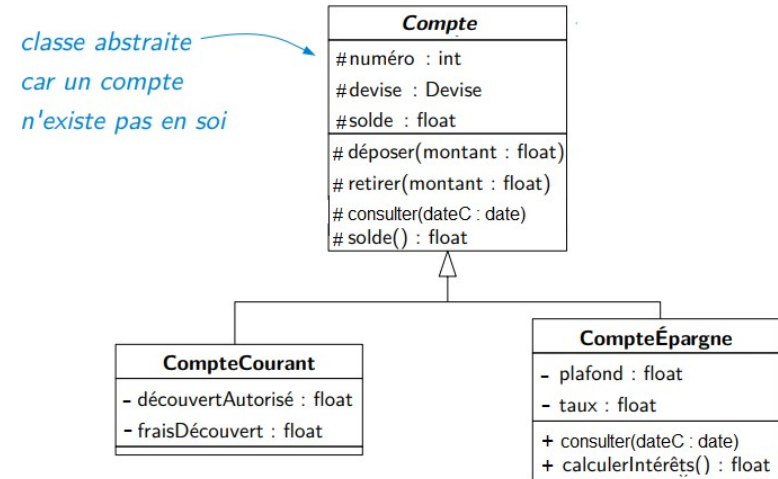


Diagramme de classes – hiérarchie de classe

❑ Interface

- Éléments de modèle qui définissent des ensembles de méthodes que des classes doivent implémenter.
- Règle de nommage : en gras avec le mot « interface »

Diagramme de classes – Exercice 8

Une entreprise de fabrication et de distribution de matériels possède une usine et plusieurs lieux de stockage/expédition.

Un produit est caractérisé par un numéro, un libellé, un prix unitaire.

Chaque produit peut être stocké dans un ou plusieurs dépôts. Un dépôt est caractérisé par un numéro. Dans chaque dépôt on connaît la quantité en stock de chaque produit et la quantité disponible (la différence représente la quantité réservée pour des commandes déjà validées mais non livrées).

Un client est déterminé par son numéro, son nom, son adresse, le total de son chiffre d'affaires, le taux de réduction. Chaque client est livré à partir d'un dépôt privilégié, ou à partir d'un dépôt de secours en cas de défaillance du premier.

A un client peuvent être associées une ou plusieurs commande(s), chacune étant caractérisée par un numéro et une date. Une ligne comporte un code produit, une quantité commandée, un délai de livraison et un code livraison indiquant si la livraison est intervenue.

A chaque commande peuvent être associées une ou plusieurs factures, une facture étant élaborée dès qu'une livraison est intervenue. Une facture est caractérisée par un numéro de facture, une date, un montant.

Une facture peut concerner plusieurs produits. Chaque ligne comprend la quantité facturée et le montant correspondant.

Proposez un diagramme de classes UML.

Diagramme de classes – Exercice 9

Une compagnie aérienne veut mettre en œuvre un système pour gérer ses différents vols.

Un VOL est un parcours aérien caractérisé par un NUMERO, une VILLE-DEPART, une VILLE-ARRIVEE, une HEURE-DEPART, une HEURE-ARRIVEE, une DISTANCE, une FREQUENCE.

Lorsqu'un VOL est programmé pour une DATE déterminée il constitue un DEPART. Un VOL n'est programmé qu'une seule fois dans une journée à l'heure prévue.

Un certain nombre de PASSAGERS peut être enregistré pour un DEPART. Un PASSAGER est caractérisé par son NOM, son ADRESSE et son NO-TELEPHONE.

Un AVION est affecté à chaque DEPART. Un AVION est caractérisé par un NUMERO, un TYPE, une CAPACITE. Un AVION utilise une certaine QUANTITE DE CARBURANT pour accomplir le trajet. Cette dernière dépend des conditions atmosphériques, donc de la DATE.

Un certain nombre de PERSONNELS est affecté à chaque DEPART. On distingue les personnels non-navigants des personnels navigants. Parmi ces derniers, on distingue le(s) pilote(s). Un membre du personnel est caractérisé par son NOM, son ADRESSE, son NO-TELEPHONE.

Diagramme de classes – Exercice 9

- ❑ On désire avoir les informations suivantes (entre autres) :
 - Passagers enregistrés pour un départ,
 - Personnels affectés à un départ pour chacune des trois catégories,
 - Départs programmés pour un vol donné,
 - Départs assignés à un avion donné pour la semaine à venir,
 - Caractéristiques du vol correspondant à un départ

Proposez un diagramme de classe UML (tous les éléments figurant en majuscule dans l'énoncé doivent être pris en compte), en exploitant au maximum la relation d'héritage.

- ❑ Un VOL peut en fait être constitué de plusieurs tronçons. Par ailleurs, on souhaite pouvoir établir pour chaque VILLE les vols au départ et les vols à l'arrivée.
Suggérez une amélioration du modèle précédent pour prendre en compte ces deux aspects.

Diagramme de classes – Exercice 10

- Il s'agit d'établir le schéma conceptuel d'une base de données pour la gestion des formations d'un institut privé. Un cours est caractérisé par un numéro de cours, un libellé, une durée en heures et un type . Un cours peut faire l'objet dans l'année de plusieurs sessions identiques. Une session est caractérisée par un numéro, une date de début et un prix. Une session est le plus souvent assurée par plusieurs animateurs et est placée sous la responsabilité d'un animateur principal. Un animateur peut intervenir dans plusieurs sessions au cours de l'année. On désire mémoriser le nombre d'heures effectué par un animateur pour chaque session. Un animateur est caractérisé par un numéro, un nom et une adresse.

Chaque session est suivie par un certain nombre de participants. Un participant est une personne indépendante ou un employé d'une entreprise cliente. Un participant est caractérisé par un numéro, un nom et une adresse. Dans le cas d'un employé, on enregistre le nom et l'adresse de l'entreprise. On désire pouvoir gérer d'une manière séparée (pour la facturation notamment) les personnes indépendantes d'une part, et les employés d'autre part. Si nécessaire, on fera les hypothèses sémantiques complémentaires qui pourraient s'imposer.

Établir un diagramme de classes UML.

Diagramme de séquences

Diagramme de séquences - présentation

- Objectif : Représenter les communications avec et au sein du logiciel
 - Représentation temporelle des interactions entre les objets
 - Chronologie des messages échangés entre les objets et avec les acteurs

Décrire la réalisation des cas d'utilisation sur le système décrit par le diagramme de classes

- Point de vue interne sur le fonctionnement du système
- Description au niveau de l'instance (état du système à un instant)
- Description de scénarios particuliers
- Représentation des échanges de messages
- Entre les acteurs et le système, entre les objets du système
- De façon chronologique

Diagramme de séquences - Représentation

□ Éléments du diagramme de séquence

- Acteurs
- Objets (instances)
- Messages (cas d'utilisation, appels de méthodes)

Principes de base : Représentation graphique de la chronologie des échanges de messages avec le système ou au sein du système

- « Vie » de chaque entité représentée verticalement
- Échanges de messages représentés horizontalement

Diagramme de séquences - Représentation

- ❑ Objectif : Description de la réalisation d'un cas d'utilisation sur le système décrit par le diagramme de classes

Problème : Communication entre les acteurs et le système vu comme un ensemble d'objets

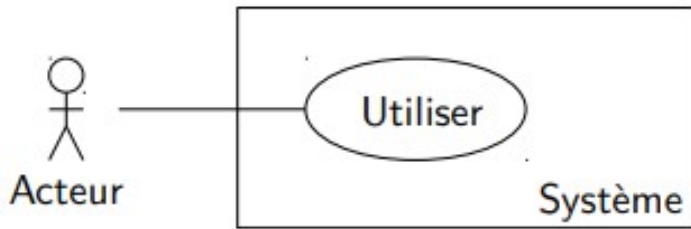


Diagramme de cas d'utilisation

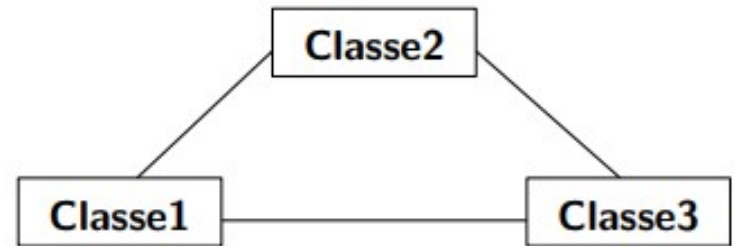


Diagramme de classes du système

Diagramme de séquences - Représentation

Communication entre acteurs et système via une interface (texte, web, physique)

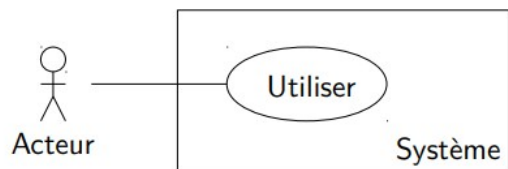


Diagramme de cas d'utilisation

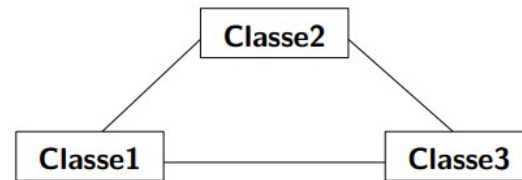


Diagramme de classes du système

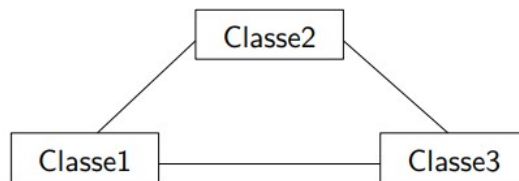


Diagramme de séquences - Représentation

- ❑ Solution : Création de classes de contrôle et de classes d'interface qui :
 - gèrent les interactions avec les acteurs
 - encapsulent le résultat des méthodes

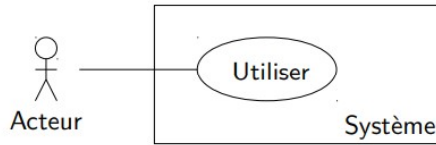


Diagramme de cas d'utilisation

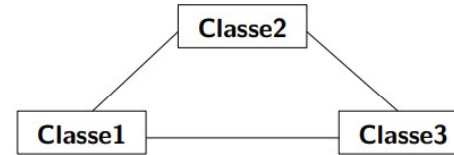


Diagramme de classes du système

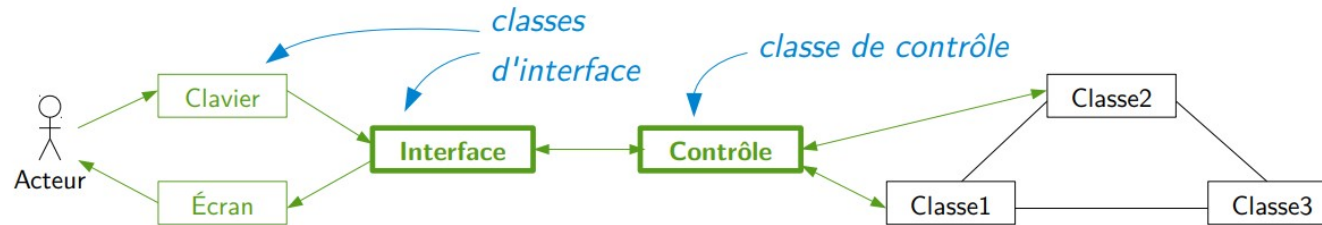


Diagramme de séquences - Représentation

Exemple

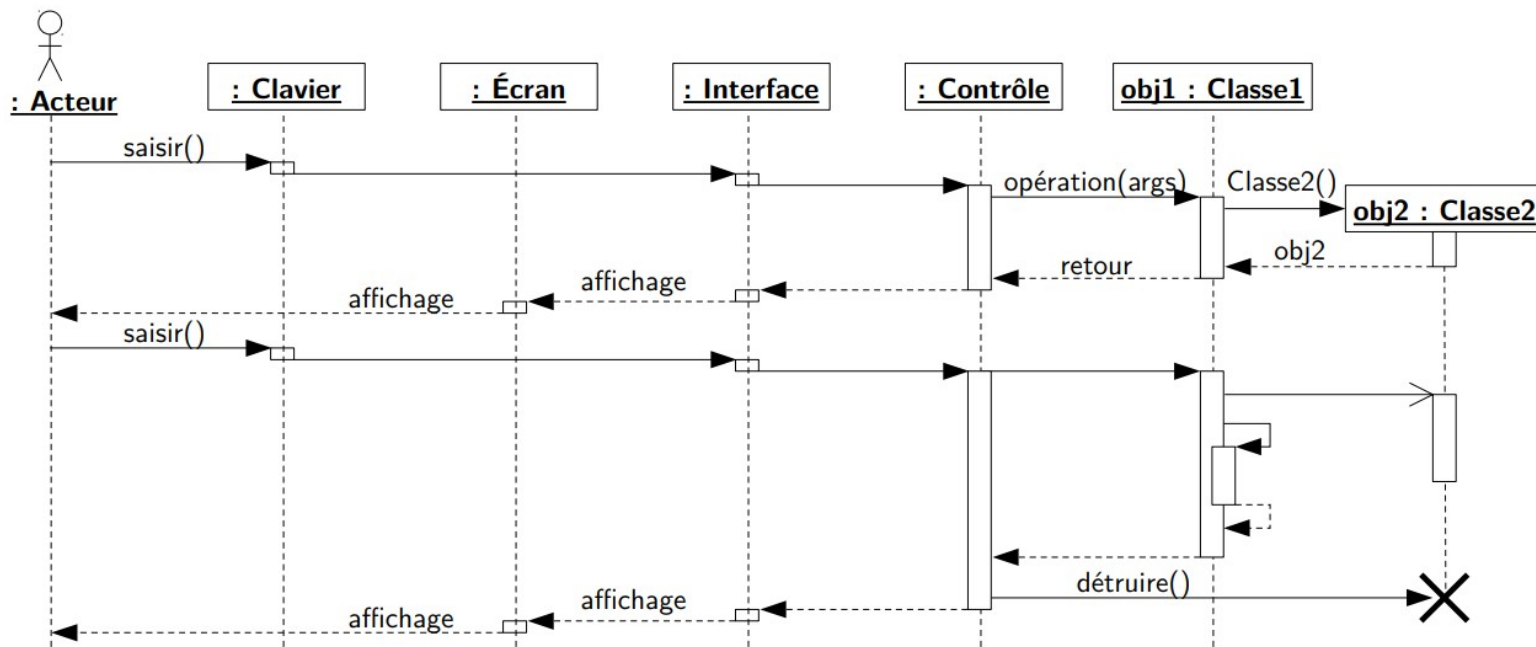


Diagramme de séquences - Représentation

■ Exemple

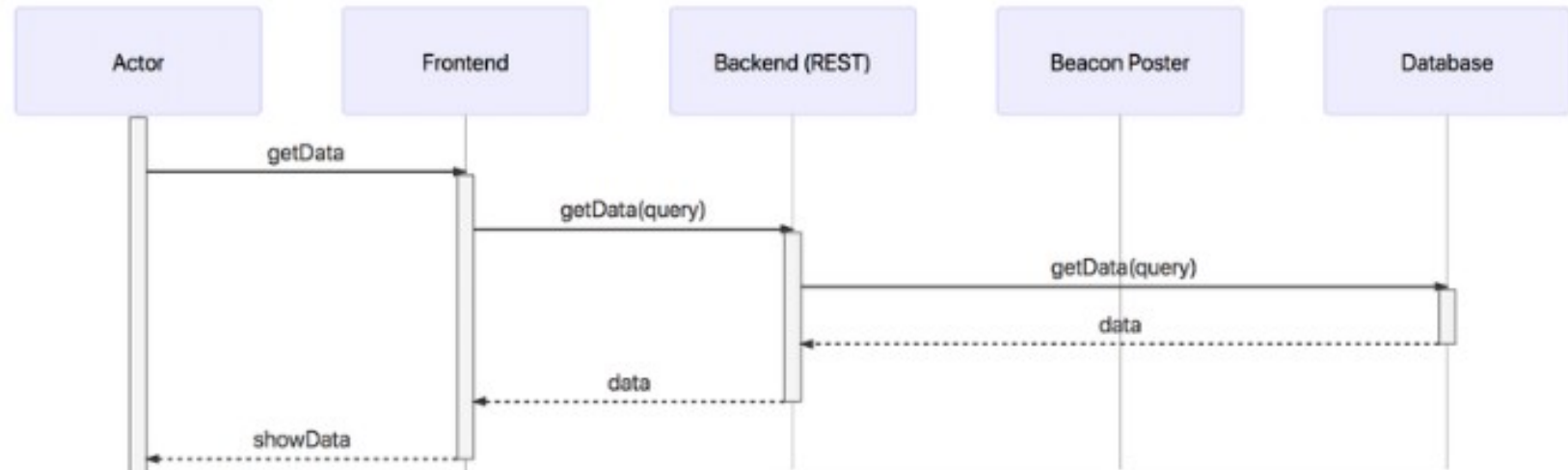


Diagramme de séquences - Représentation

❑ Éléments de base

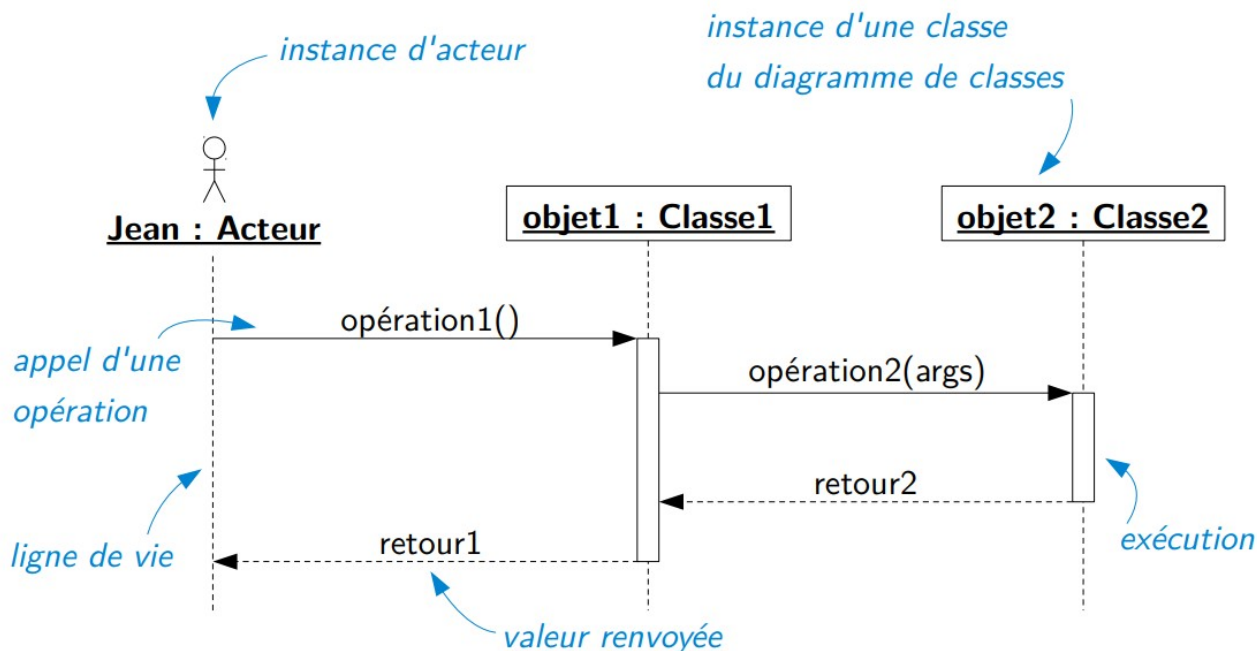
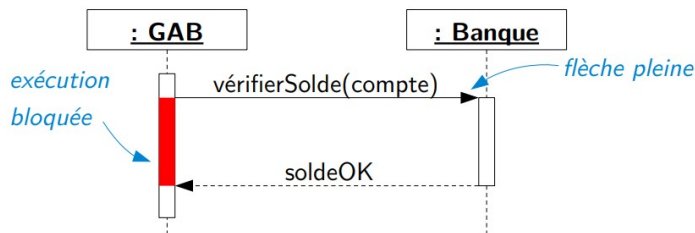


Diagramme de séquences - Représentation

□ Type de messages

- Message synchrone : Émetteur bloqué en attente du retour



- Message asynchrone : Émetteur non bloqué, continue son exécution

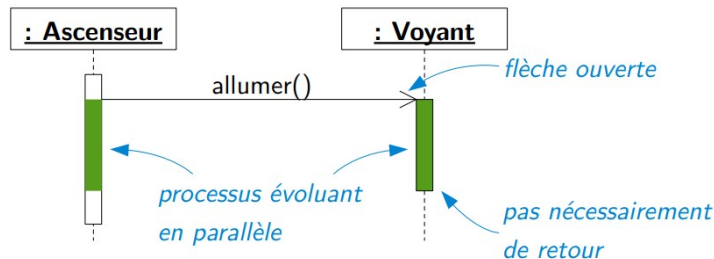


Diagramme de séquences - Représentation

❑ Création et destruction d'objets

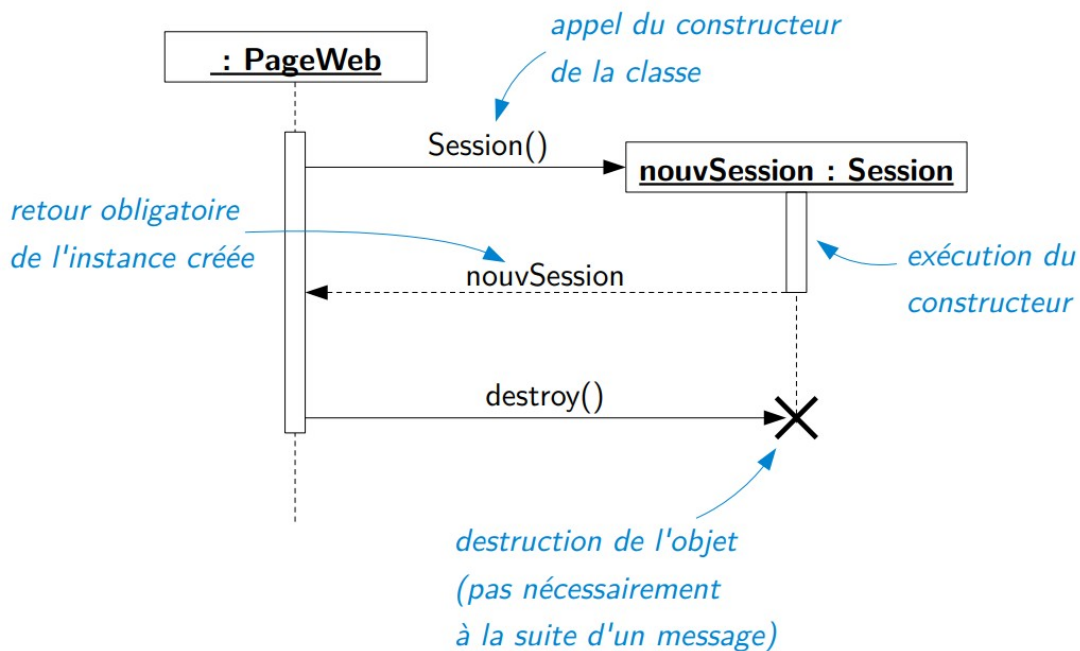


Diagramme de séquences - Représentation

❑ Message réflexif

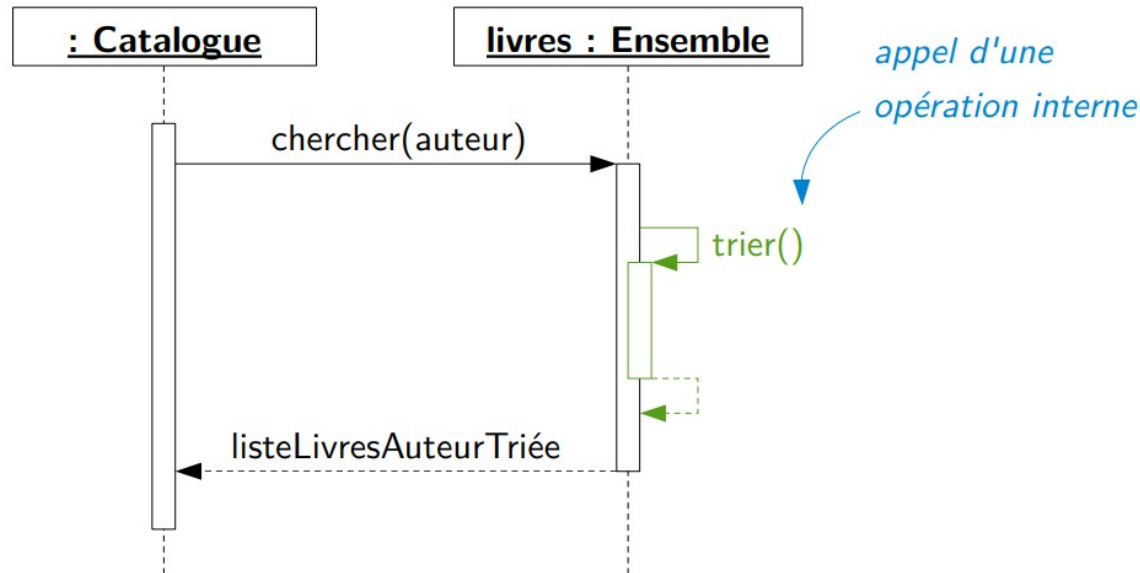


Diagramme de séquences - Représentation

❑ Alternative

- Principe : Condition à l'envoi d'un message
- Notation : Deux diagrammes

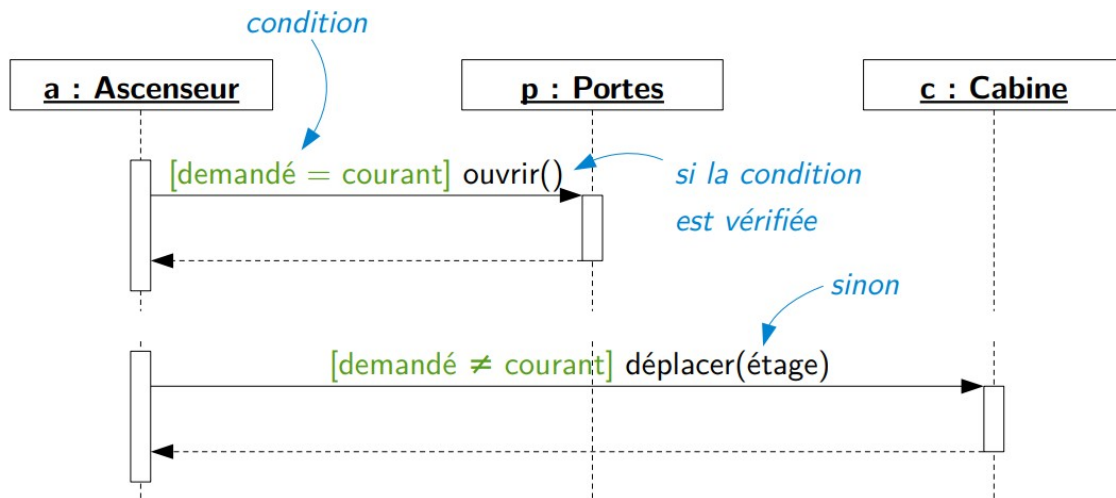


Diagramme de séquences - Représentation

❑ Alternative

○ Notation : Bloc alt

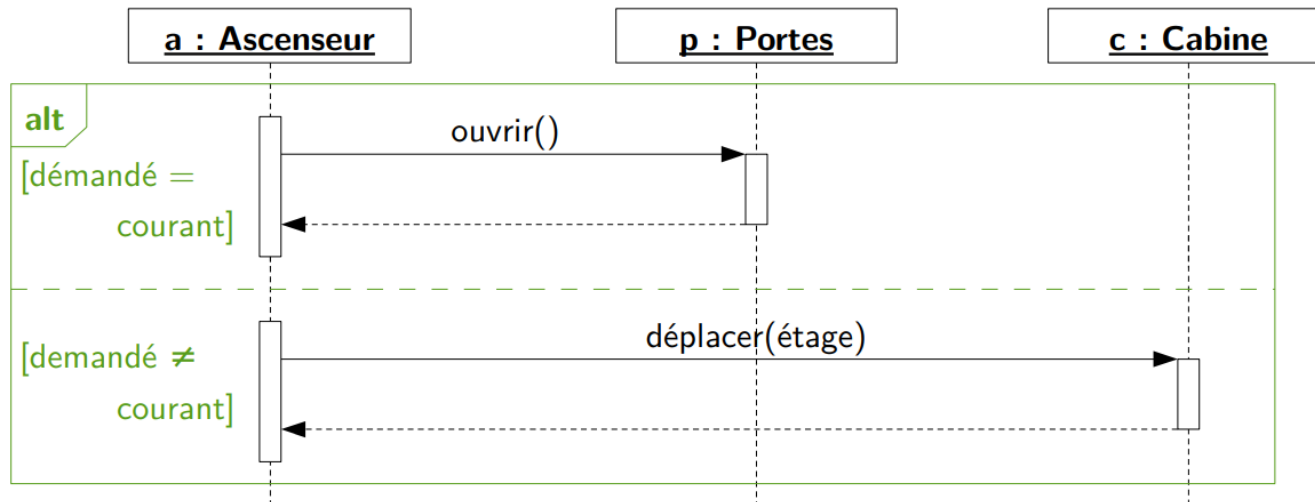


Diagramme de séquences - Représentation

□ Boucle

- Principe : Répéter un enchaînement de messages
- Notation : Note

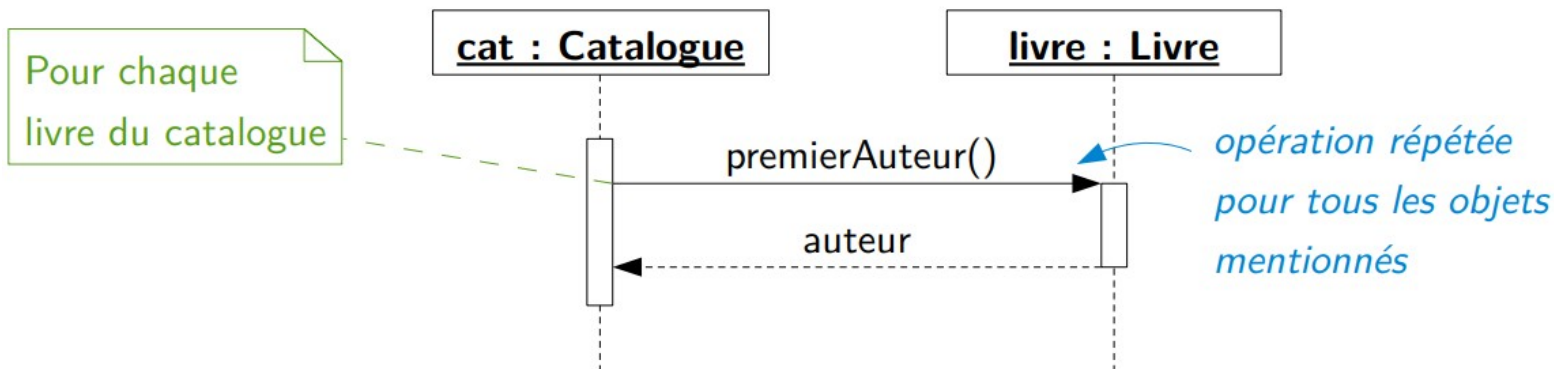


Diagramme de séquences - Représentation

❑ Boucle

- Principe : Répéter un enchaînement de messages
- Notation : Bloc de boucle loop

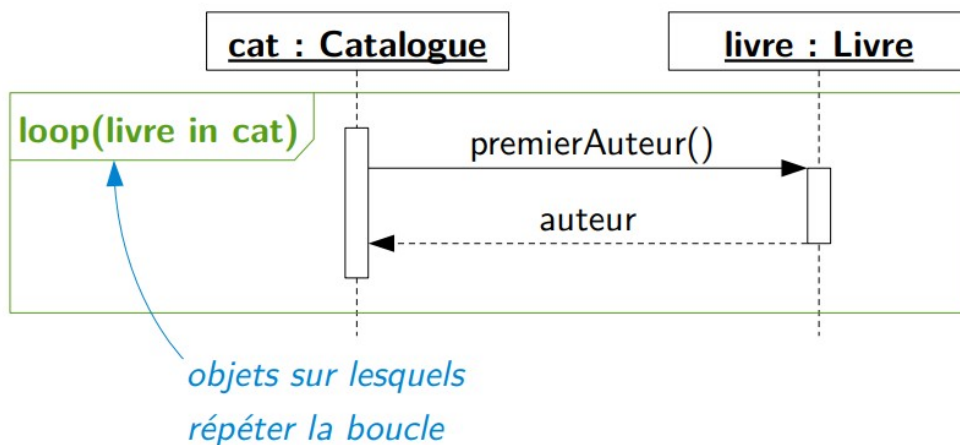


Diagramme de séquences - Représentation

❑ Référence à un autre diagramme

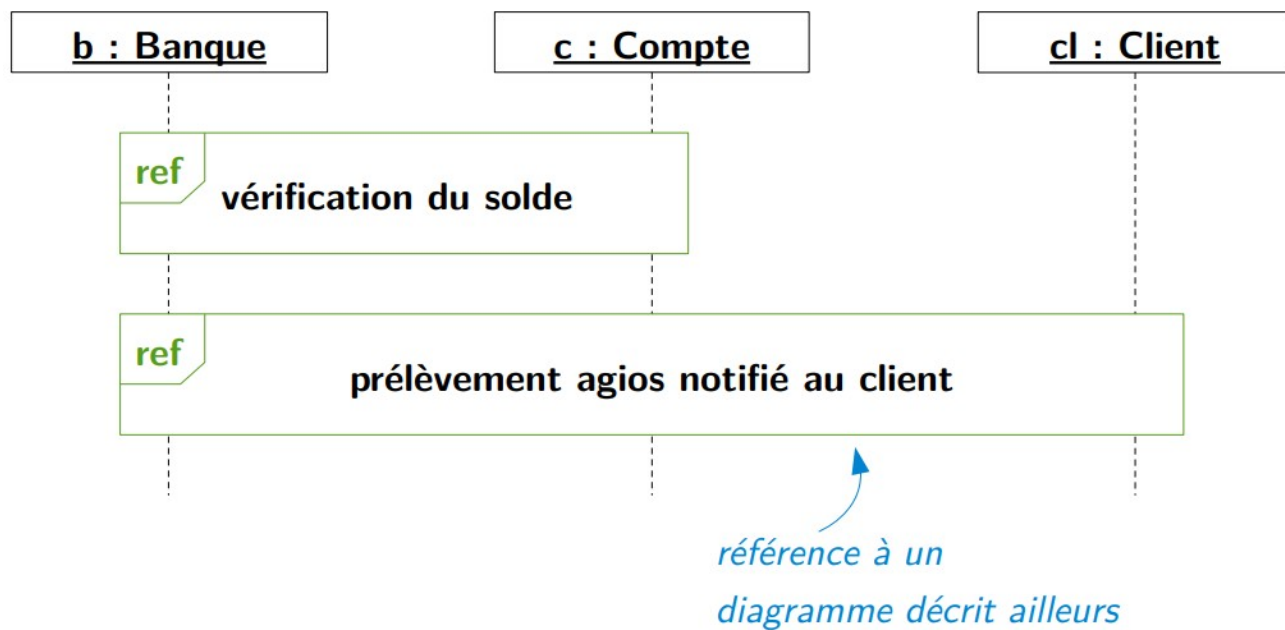
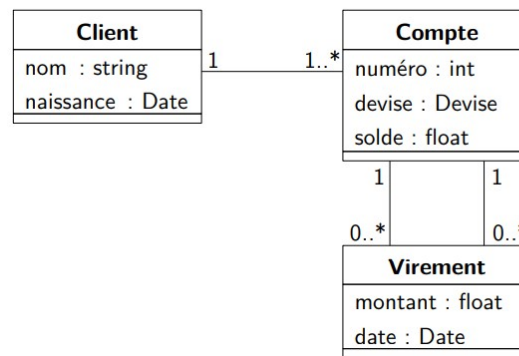
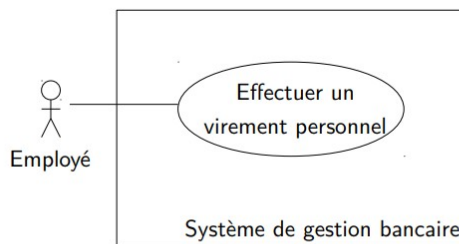


Diagramme de séquences - Représentation

Exemple



Réalisation du
cas d'utilisation
Effectuer un
virement personnel
(niveau conception)

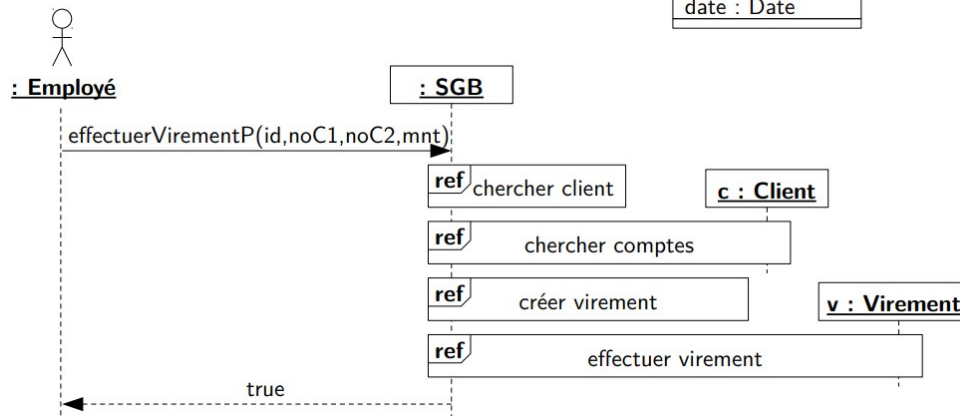


Diagramme de séquences - Représentation

Exemple

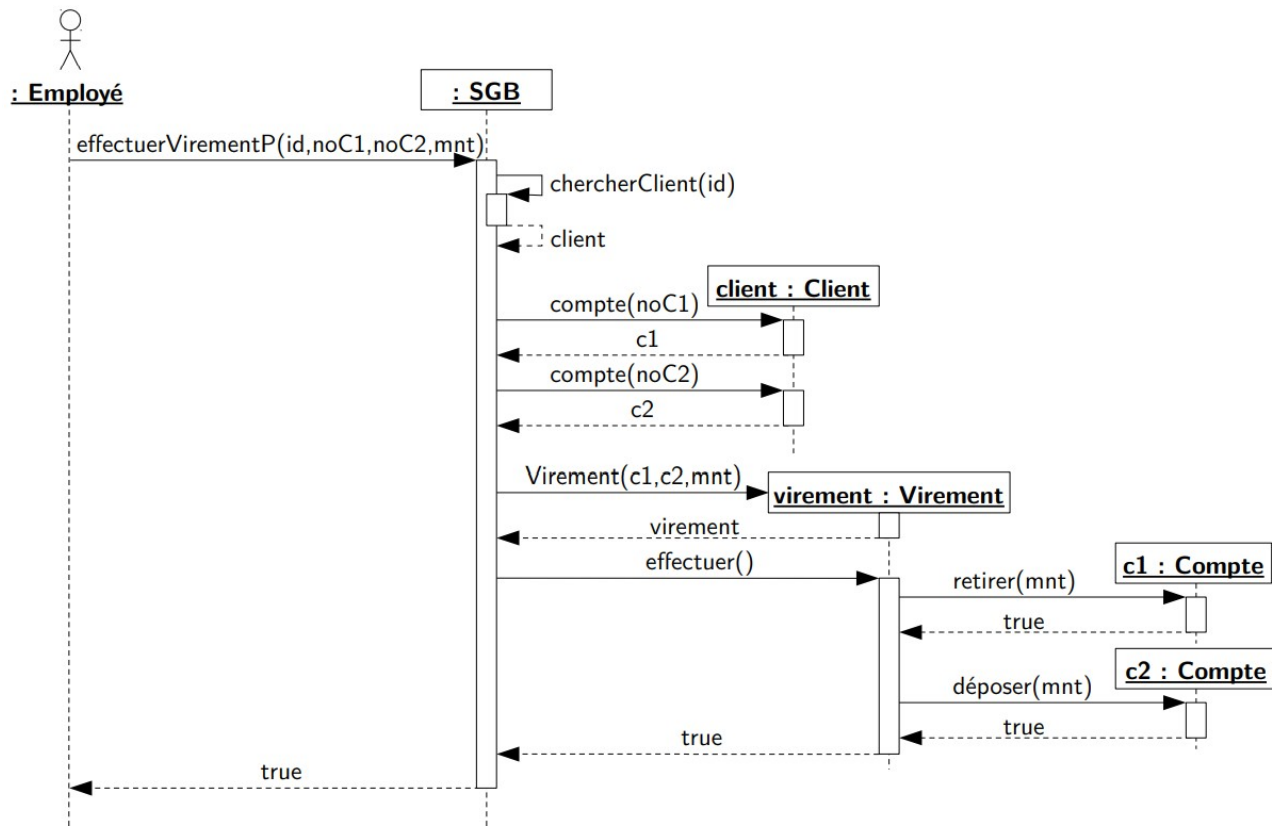


Diagramme de séquences - Représentation

Exemple

Diagramme de classes complété avec les classes techniques et les méthodes nécessaires

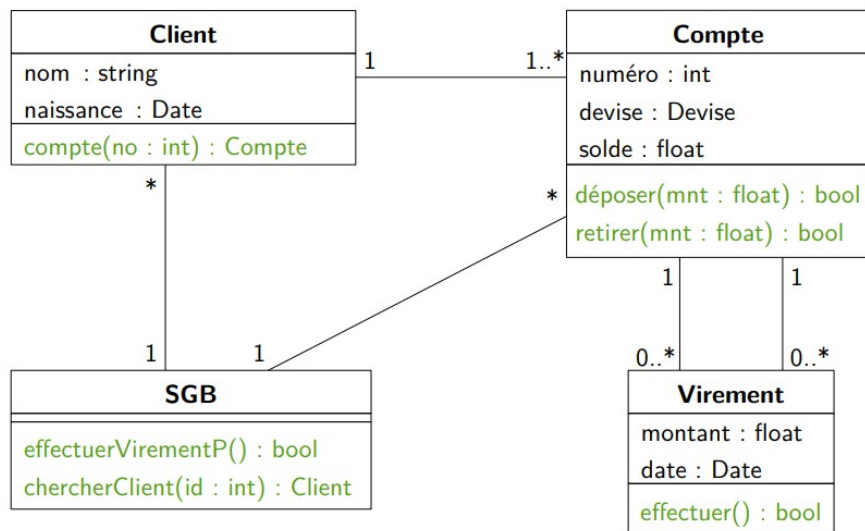
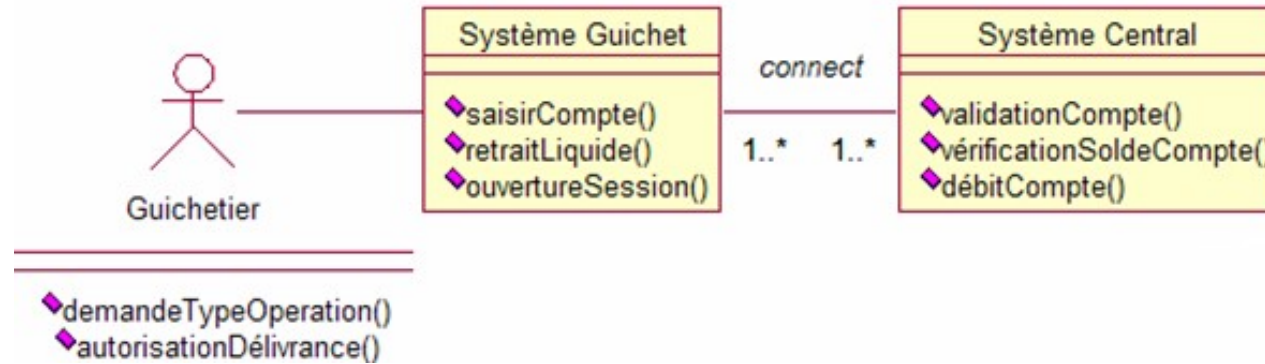


Diagramme de séquences - Représentation

❑ Quelques règles

- Messages entre acteurs et interface
 - « Fausses » opérations liées au cas d'utilisation (même nom)
 - Arguments (saisis) et valeurs de retour (affichées) simples :
texte, nombre
- Messages au sein du système
 - Opérations du diagramme de classes
 - Si message de objA : ClasseA vers objB : ClasseB, alors
opération du message dans ClasseB

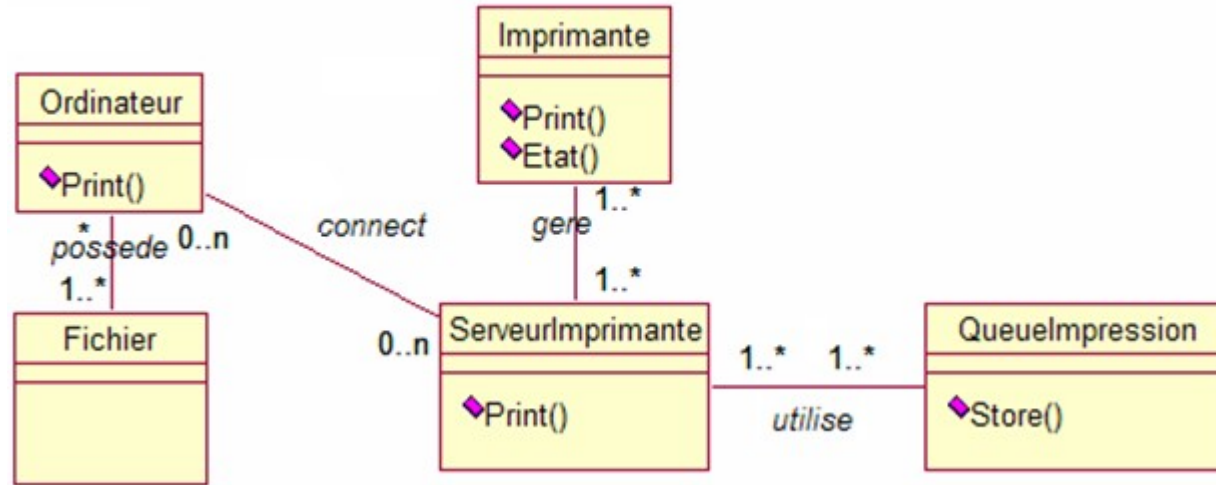
Diagramme de séquences – Exercice 11



Rédigez un diagramme de séquence basé sur l'énoncé suivant:

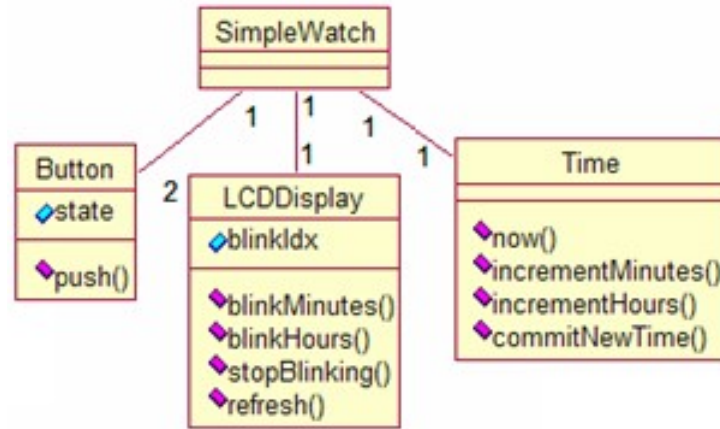
- ❑ Le guichetier ouvre une session
- ❑ Le guichetier saisit le numéro de compte du client
- ❑ Le système guichet valide le compte auprès du système central
- ❑ Le système guichet demande le type d'opération au guichetier
- ❑ Le guichetier sélectionne le montant du retrait
- ❑ Le système guichet interroge le système central pour s'assurer que le compte est suffisamment approvisionné
- ❑ Le système guichet demande au système central de débiter le compte
- ❑ Le système notifie au guichetier qu'il peut délivrer le montant demandé

Diagramme de séquences – Exercice 12



Rédigez un diagramme de séquence pour modéliser le scénario où un utilisateur voudrait imprimer un fichier.

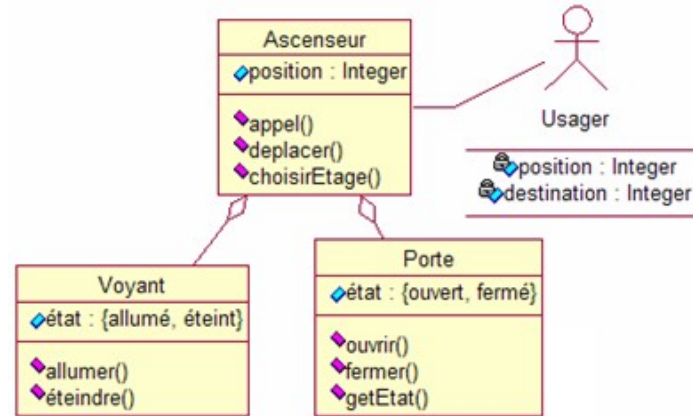
Diagramme de séquences – Exercice 13



Rédigez un diagramme de séquence pour modéliser un scénario où les utilisateurs voudraient régler l'heure (particulièrement les minutes) sur sa montre.

- En appuyant 2x sur le bouton 1, il accède au réglage des minutes (heure clignote puis minute clignote). Ensuite avec le bouton 2 (sans relâcher le bouton) il incrémente les minutes, le LCD display est rafraîchi. En appuyant sur le bouton 1 une autre fois, l'heure est enregistrée et l'affichage s'arrête de clignoter.

Diagramme de séquences – Exercice 14



Rédigez un diagramme de séquence pour modéliser un scénario où un usager voudrait monter en utilisant un ascenseur.