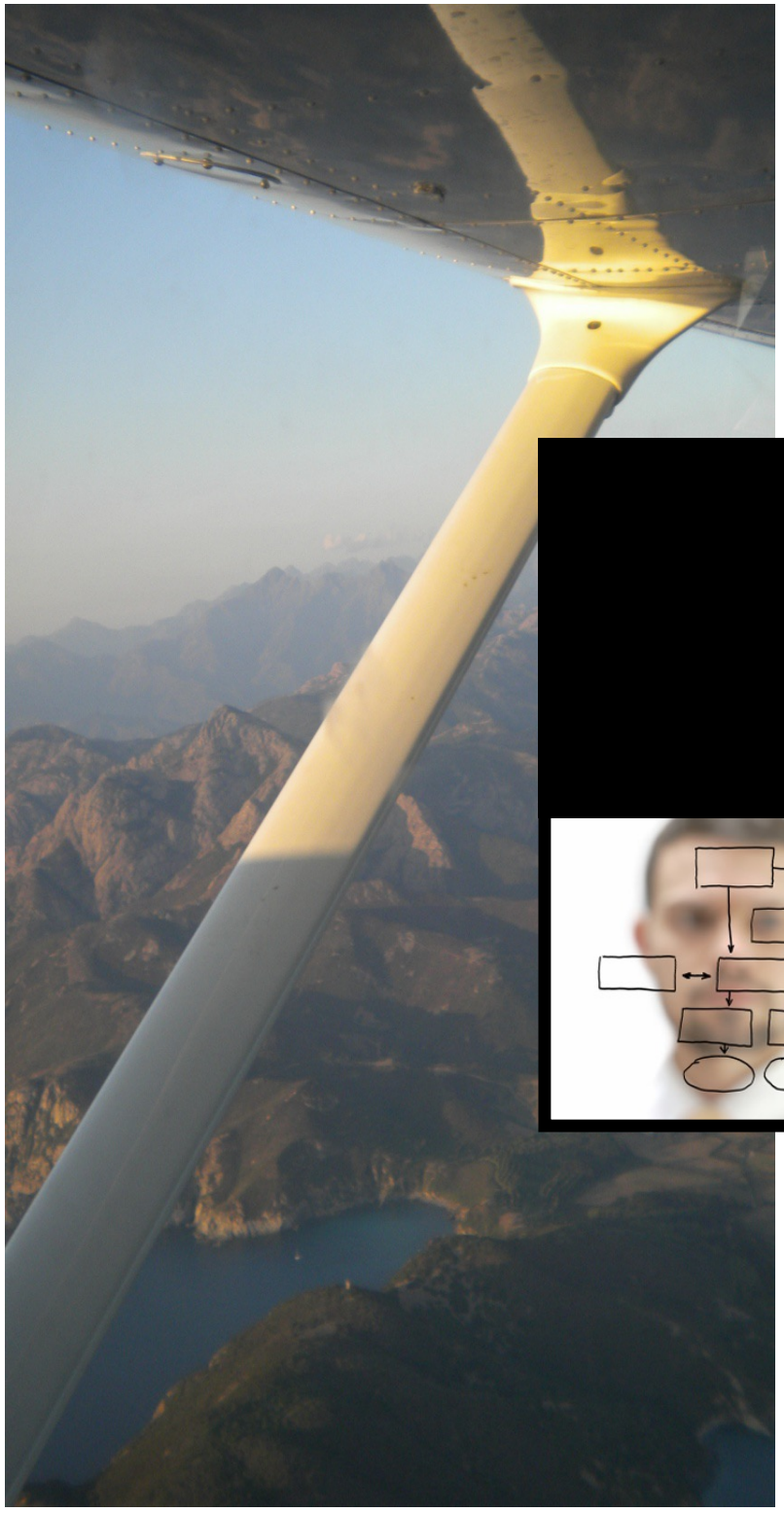
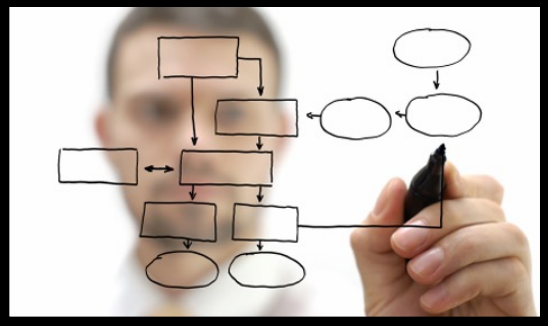
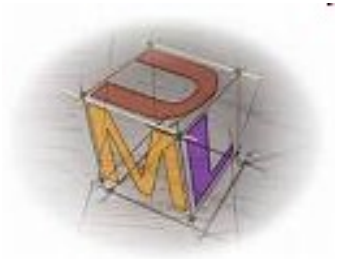


Université de Corse
MASTER Informatique DE-DFS
1ère année

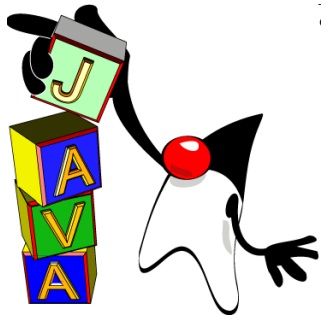
Cours Patterns

Formalisme UML (Rappels)





Evelyne VITTORI
vittori@univ-corse.fr



Formalisme UML (rappels)

- Qu'est ce qu'UML?
- Diagramme de classes
  - Représentation d'une classe en UML
  - Associations, agrégations, compositions
  - Généralisation et classes abstraites
  - Interfaces
- Diagramme de Cas d'utilisation
- Diagramme de séquence

Qu'est ce qu'UML?

Unified Modeling Language

- Un Langage standard pour l'écriture de plans d'élaboration de logiciels.

- Indépendant du langage de programmation et de la méthode.

- Notation graphique.

Qu'est ce qu'UML?

Diagramme des cas d'utilisation

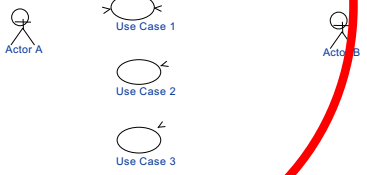


Diagramme de classe

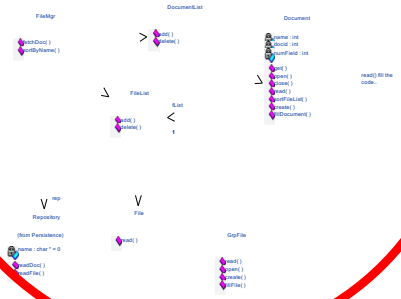


Diagramme de machines d'état

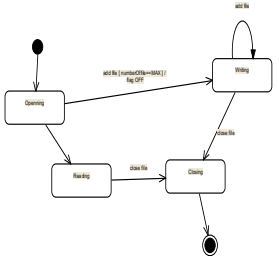


Diagramme de communication

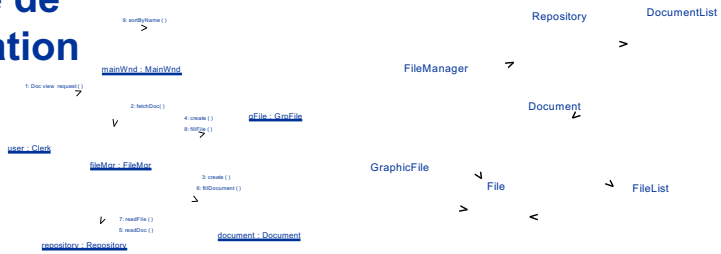


Diagramme de déploiement

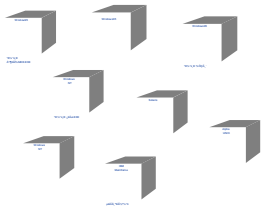
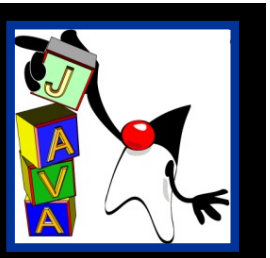


Diagramme de composants

Diagramme de séquence

Une boite à outils de 23 Diagrammes

La « boite à outils » UML: les Diagrammes UML2



Diagrammes structuraux

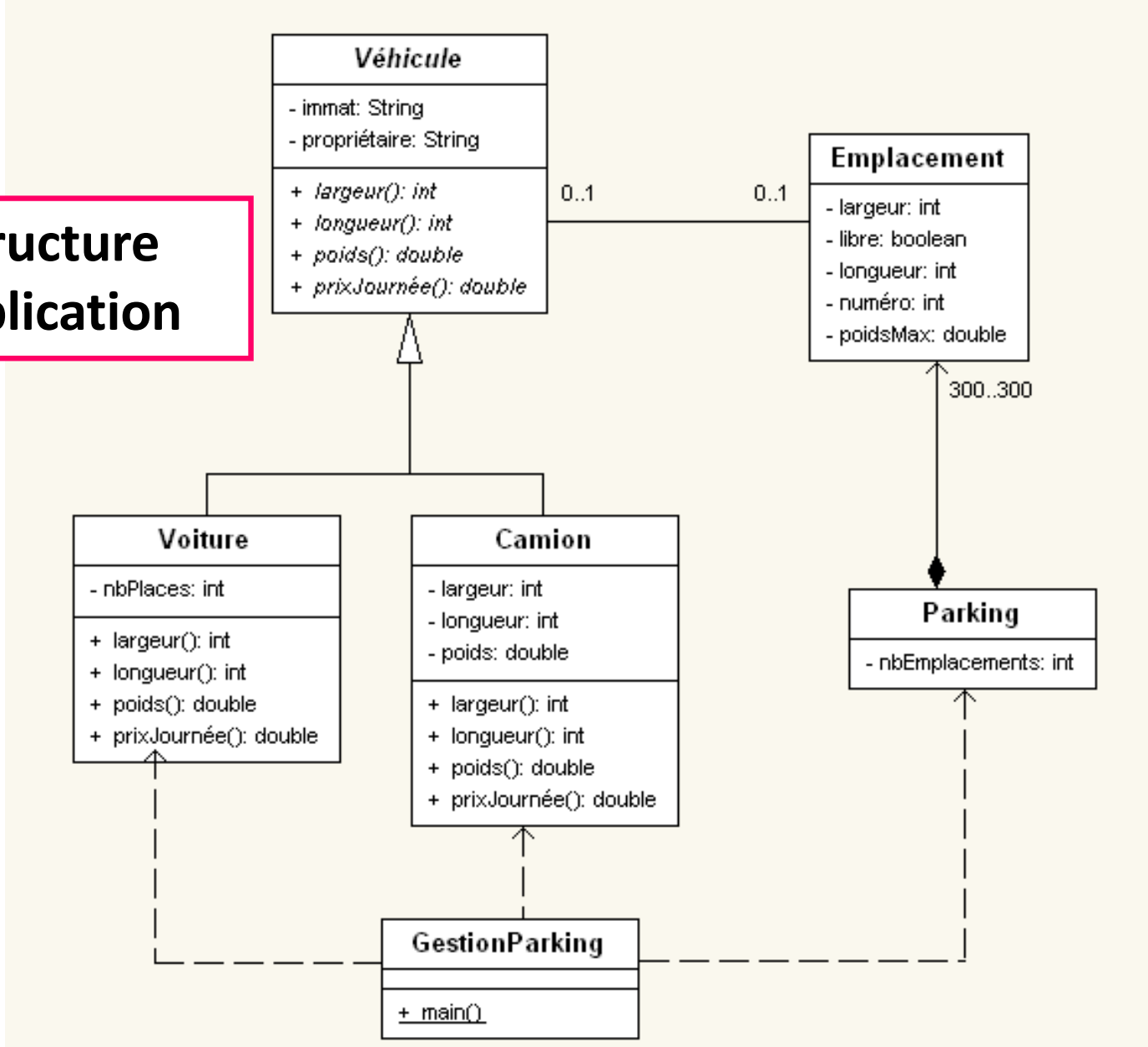
- **D. de Classe**
- D. d'objets
- D. de Package
- D. de Composants
- D. de Déploiement
- D. de Structure Composite

Diagrammes comportementaux

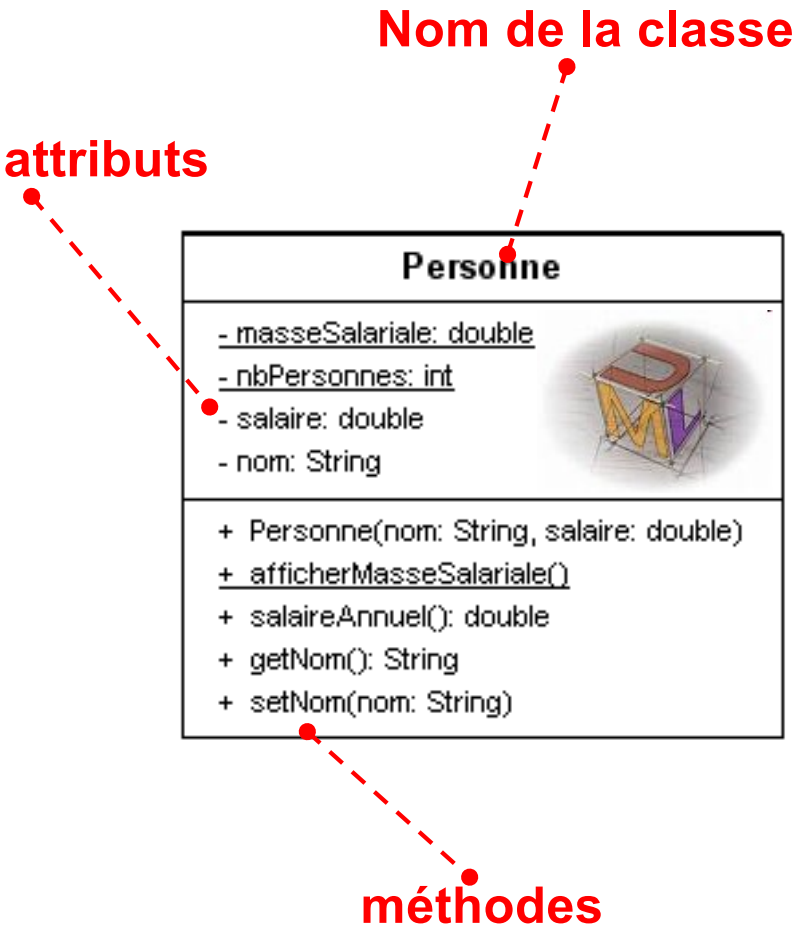
- **D. de Cas d'utilisation**
- *Diagrammes d'intéractions*
  - **D. de Séquence**
  - D. de Communication
  - D. Vue d'ensemble des intéractions
  - D. de temps
- D. d'activités
- D. de machine d'états

Diagramme de classe

Pour décrire la structure statique d'une application



Représentation d'une classe




```java
public class Personne {
        private static double masseSalariale;
        private static int nbPersonnes;
        private double salaire;
        private String nom;

public Personne(String nom, double salaire){
      this.nom=nom;
      this.salaire=salaire;
}
public static void afficherMasseSalariale(){
        System.out.println("Masse salariale: " +
                masseSalariale);
}
public double salaireAnnuel(){
        return salaire*12;
}
public String getNom(){
    return nom;
}
public void setNom(String nom ){
    this.nom = nom;
}
```

Visibilité : + (public) – (private)
Attributs/méthodes de classe: soulignés

Relations entre classes

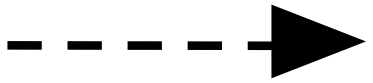
Relations entre classes /interfaces

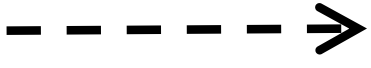
Association 

Agrégation 

Composition 

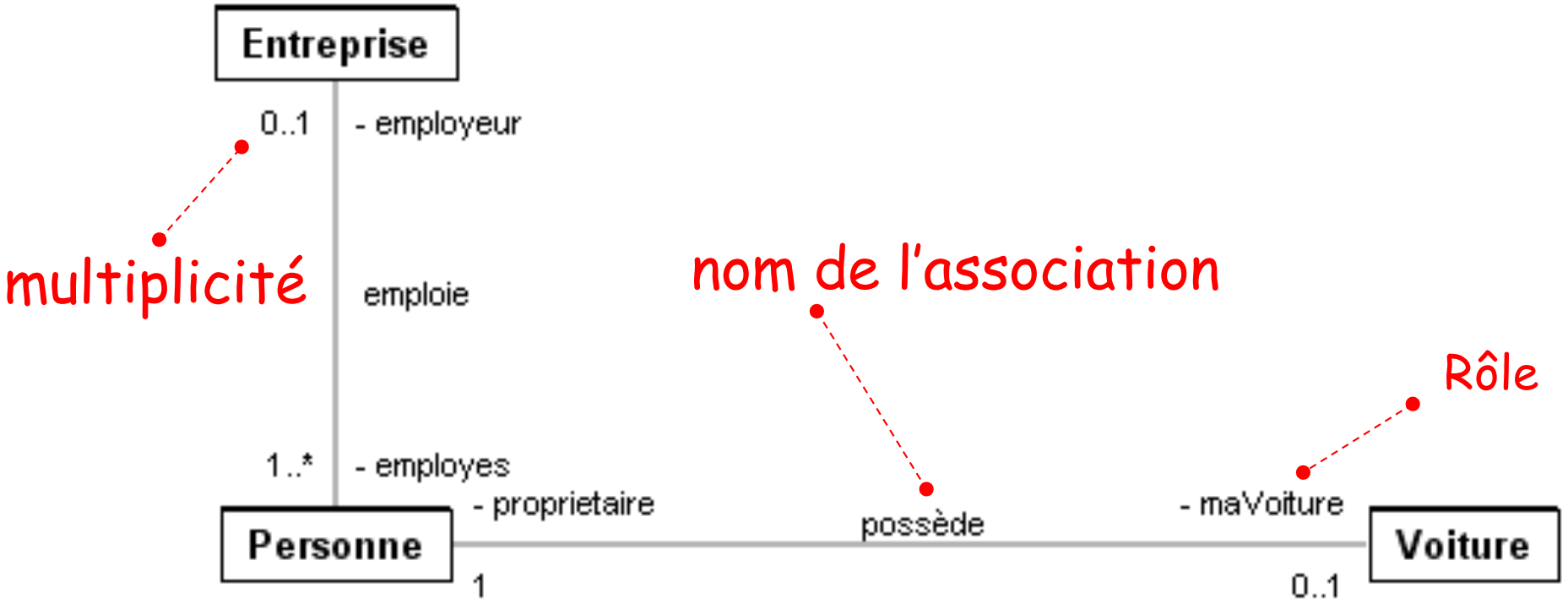
Généralisation (héritage) 

Réalisation (implémentation) 

Dépendance (utilisation) 

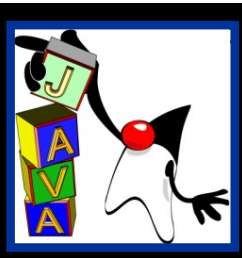
Associations entre classes

- Une association représente un lien structurel entre deux classes.



Une entreprise emploie 1 ou plusieurs personnes (ses employes).
Une personne est employée par au maximum une entreprise (son employeur).
Une personne peut ne posséder aucune voiture ou en posséder une seule.
Une voiture appartient à une personne exactement.

Associations entre classes



- Lors de la programmation, une association est traduite par l'ajout d'un (ou plusieurs) attribut(s) dans les classes associées.

- Le type des attributs dépend de la multiplicité:
  - Multiplicité ≤ 1 :

    attribut de type classe simple

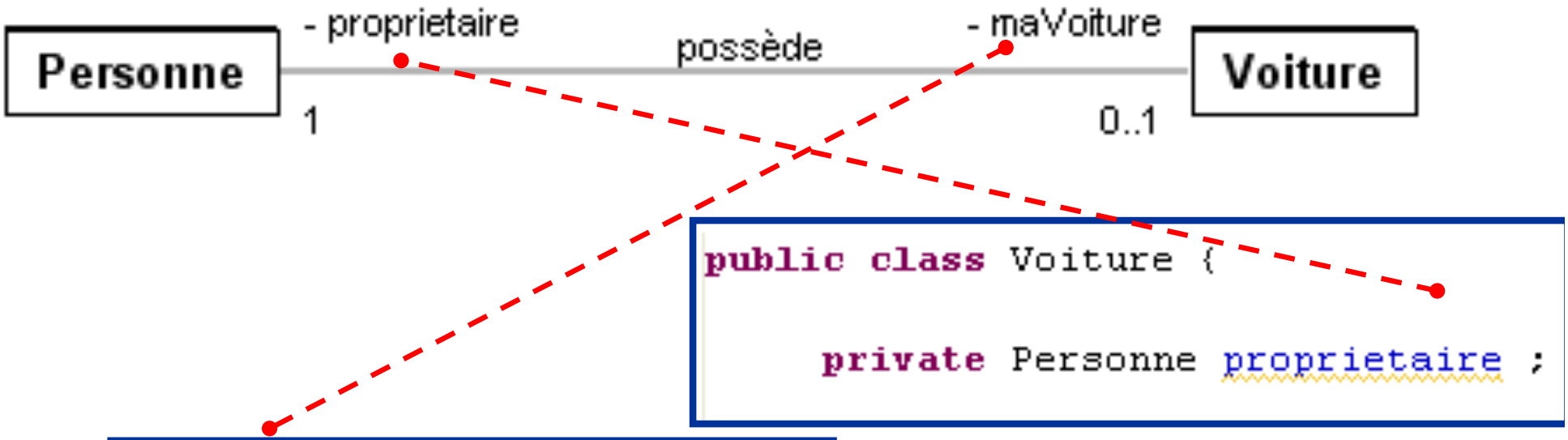
  - Multiplicité >1:

    attribut de type collection ou tableau

Associations entre classes

Multiplicité ≤1

- Les attributs ajoutés pour représenter l'association sont du type des classes associées.




```
public class Voiture {

    private Personne proprietaire ;
```

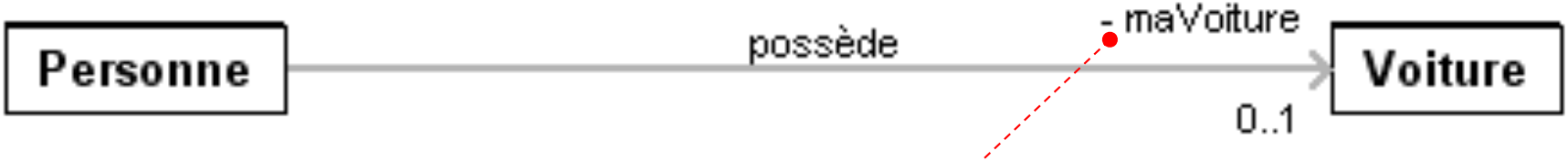
```
public class Personne {

    private Voiture maVoiture;
```

Associations entre classes

Multiplicité ≤1

- Association unidirectionnelle:

L'attribut ne sera ajouté que dans une seule classe.




```
public class Personne {

        private Voiture maVoiture;
        ...
```

Associations entre classes

Multiplicité >1

- L'attribut ajouté pour représenter l'association est de type tableau ou collection.

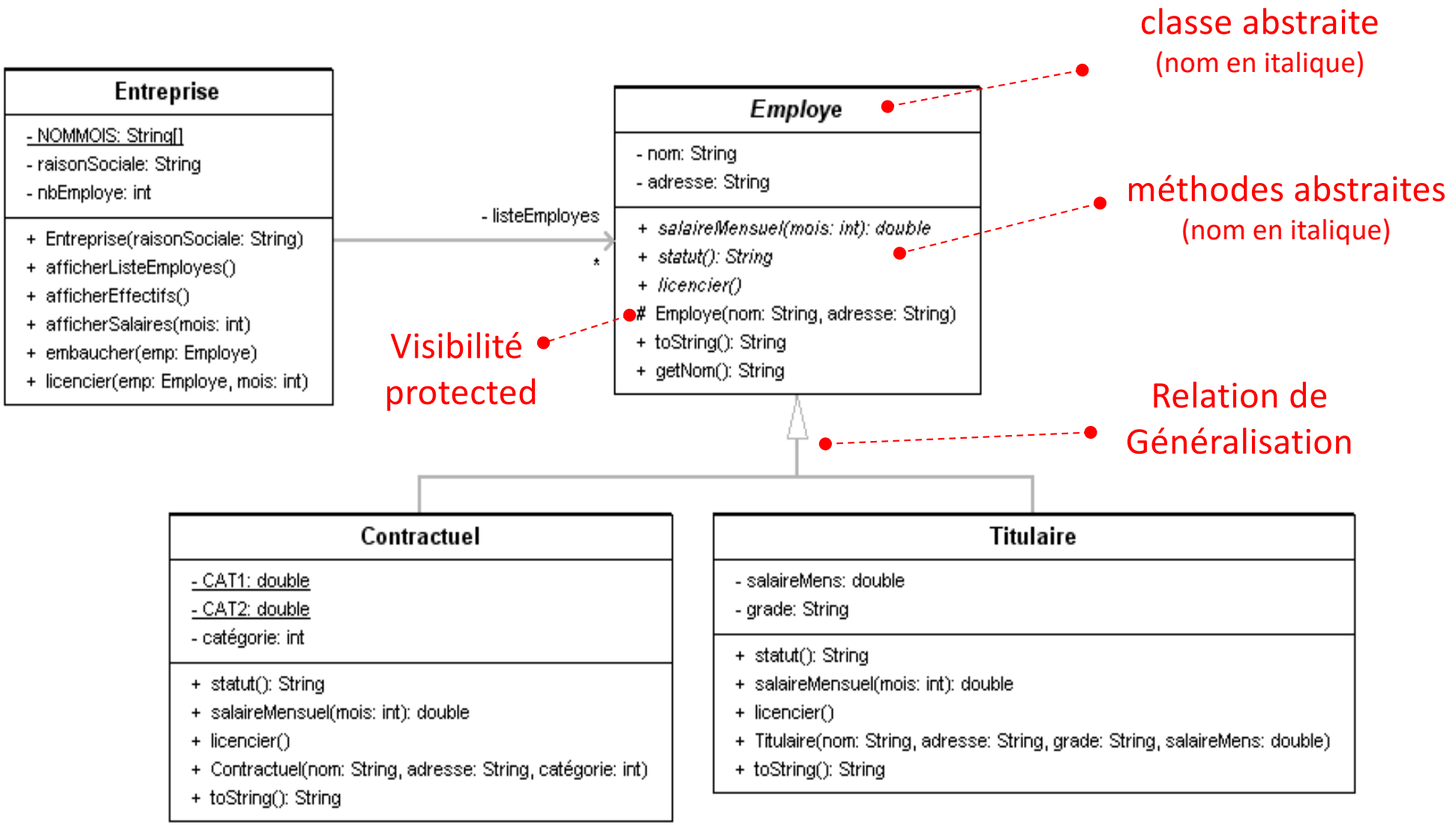

```
public class Entreprise {

    private Collection<Personne> employes;
    ...
```

Agrégation et Composition

- Une agrégation est une association non symétrique dont la sémantique évoque une relation de contenance.

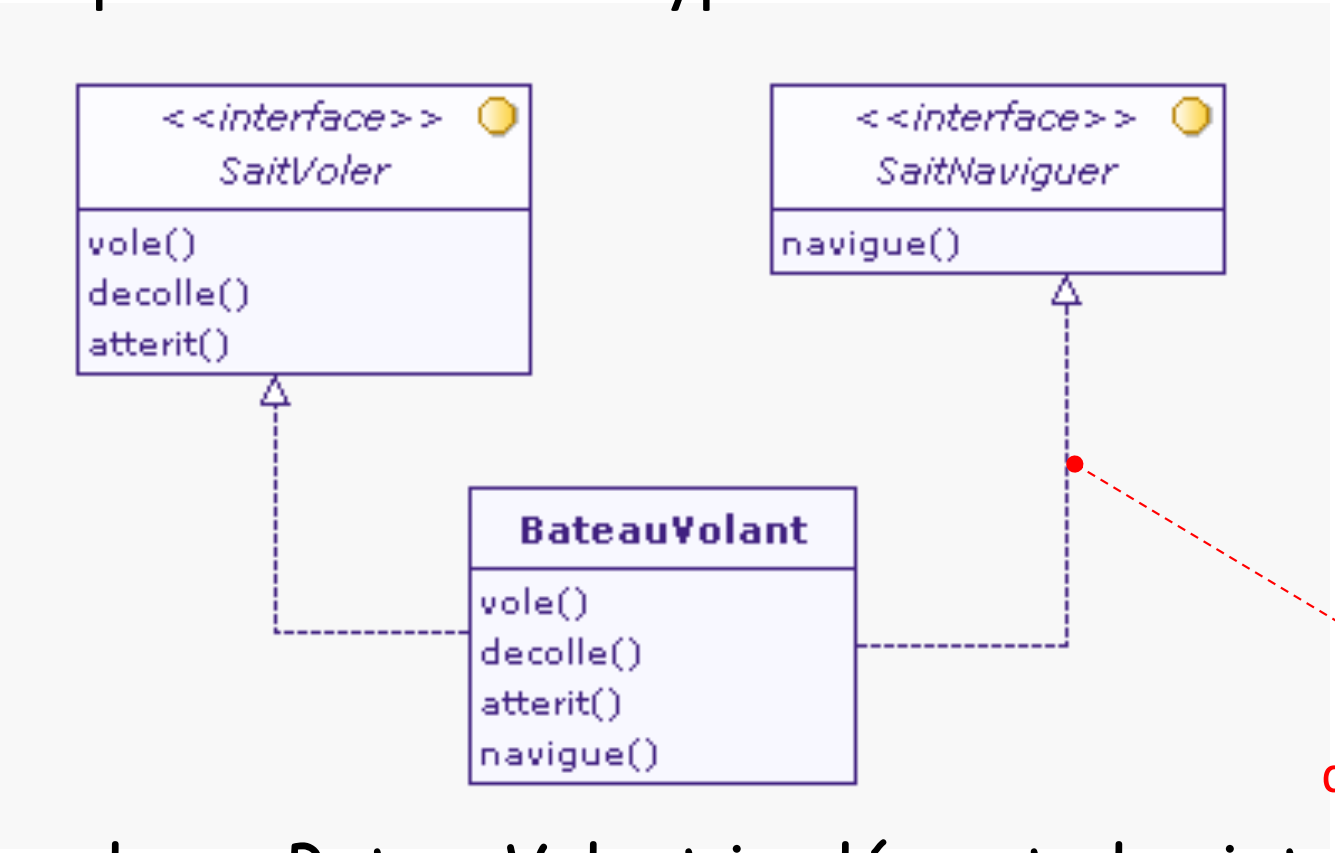
- Une composition est une agrégation non partageable avec des contraintes fortes sur les cardinalités et les durées de vie composant/composé.

Généralisation- Classes abstraites



Interfaces

- Une interface est représentée en UML par une classe comportant le stéréotype <<interface>>.



La classe BateauVolant implémente les interfaces SaitVoler et SaitNaviguer.

Dépendances

- Une dépendance est une relation non structurelle entre classes (communication momentanée, limitée dans le temps):

Au moins une méthode de A :
- *contient une variable locale de type B*
- *possède un paramètre de type B*
- *renvoie un résultat de type B*

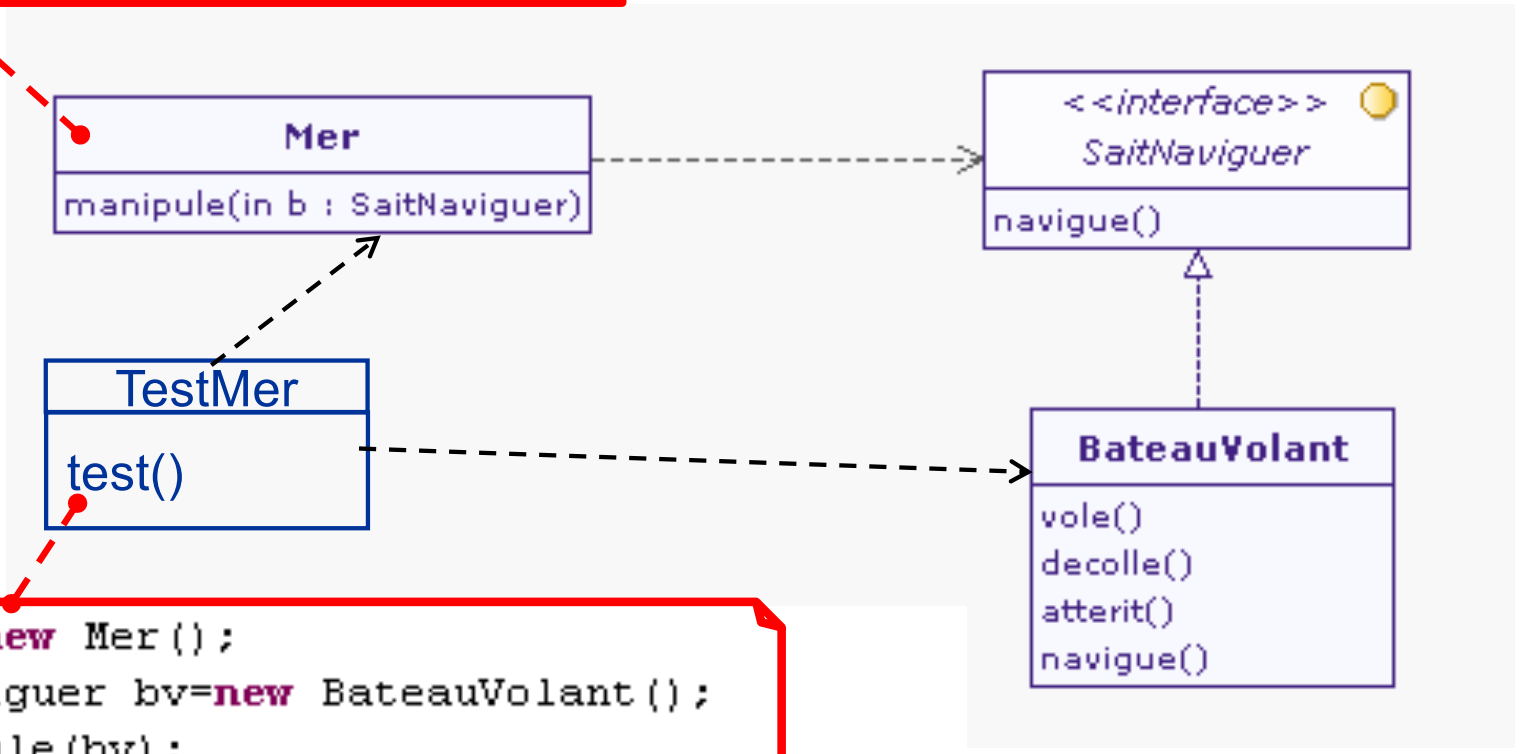
Variable locale de type Robot

```
public class ManipRobot {
    public static void main(String[] args) {
        Robot r=new Robot("Toto",10,20,Robot.NORD);
        r.changerOrientation(Robot.SUD);
        r.déplacer();
    }
}
```

Dépendances

- Une classe peut dépendre d'une interface.

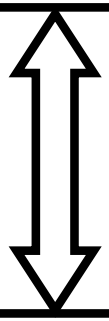
b devra être une instance d'une classe implémentant l'interface SaitNaviguer



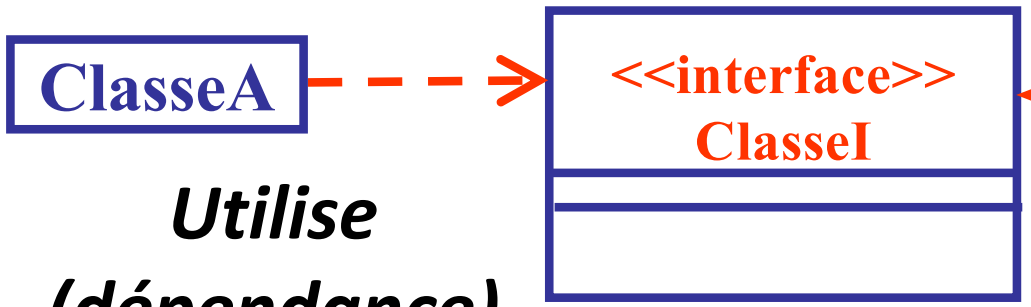

```
Mer m= new Mer();
SaitNaviguer bv=new BateauVolant();
m.manipule(bv);
```

Interfaces et dépendances

Autre représentation des interfaces (***lollipops***)



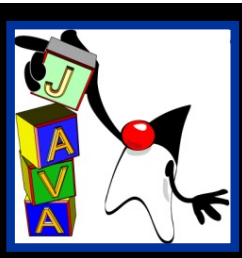
Représentations équivalentes



Utilise (dépendance)

Implémente (réalisation)

Formalisme UML



- Qu'est ce qu'UML?
- Diagramme de classes
- Diagramme de Cas d'Utilisation
  - Objectif
  - UC et Acteur
  - Documentation textuelle
  - Scénarios
- Diagramme de séquence

Diagramme de Cas d'utilisation
Objectif

Pour décrire le comportement d'un système du point de vue de l'utilisateur

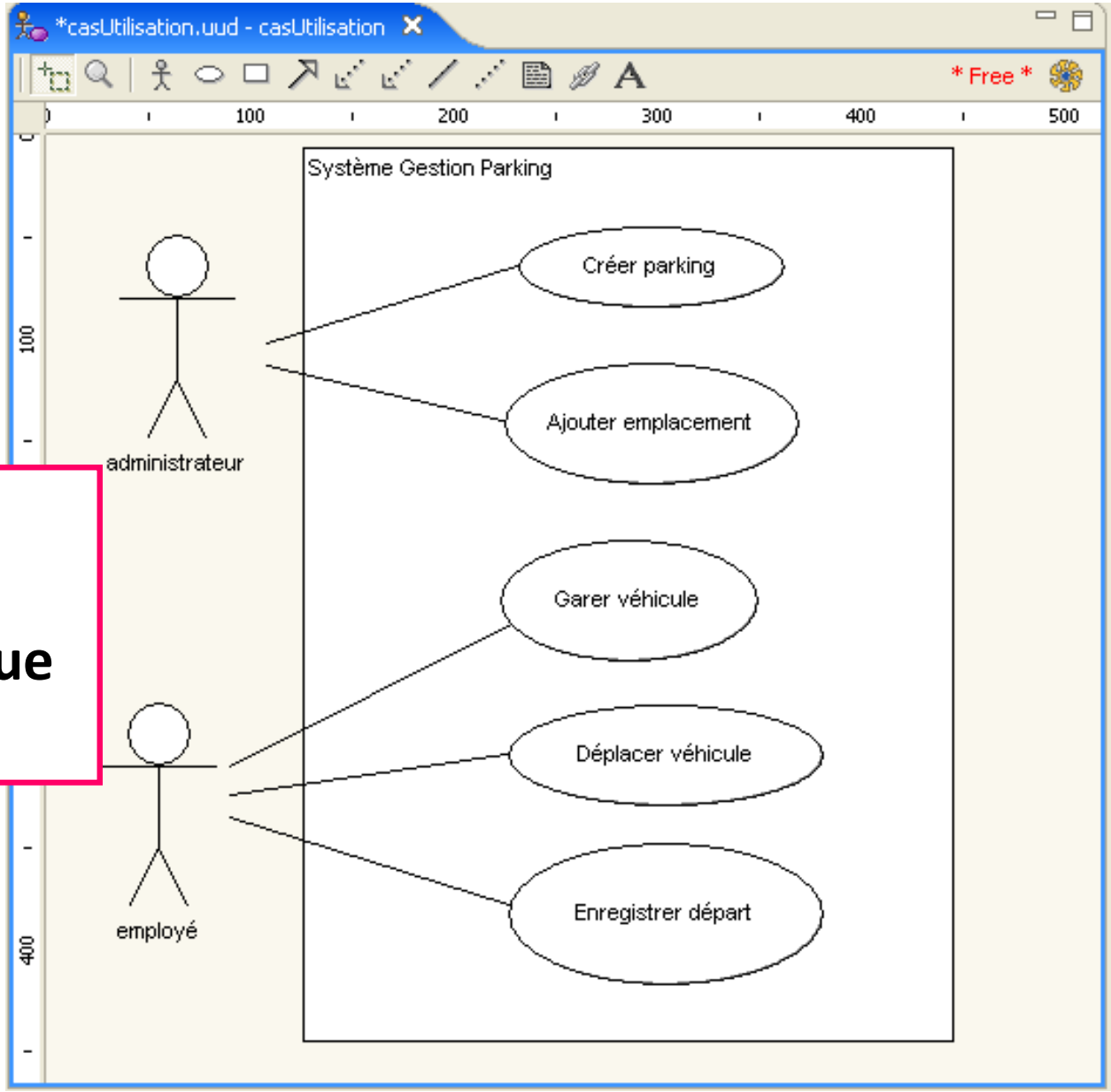


Diagramme de Cas d'utilisation

➔Un cas d'utilisation est la représentation d'une fonctionnalité du système déclenchée en réponse à la stimulation d'un acteur externe.

Cas d'utilisation

➔ Un acteur représente un rôle joué par une personne ou une chose qui interagit avec le système.

NomActeur

Diagramme de Cas d'utilisation

Relations

Communication entre acteur et UC

Généralisation

- ***entre UC***
- ***entre acteurs***

Dépendance d'inclusion

Dépendance d'extension

Diagramme de Cas d'utilisation
Exemple Médiathèque

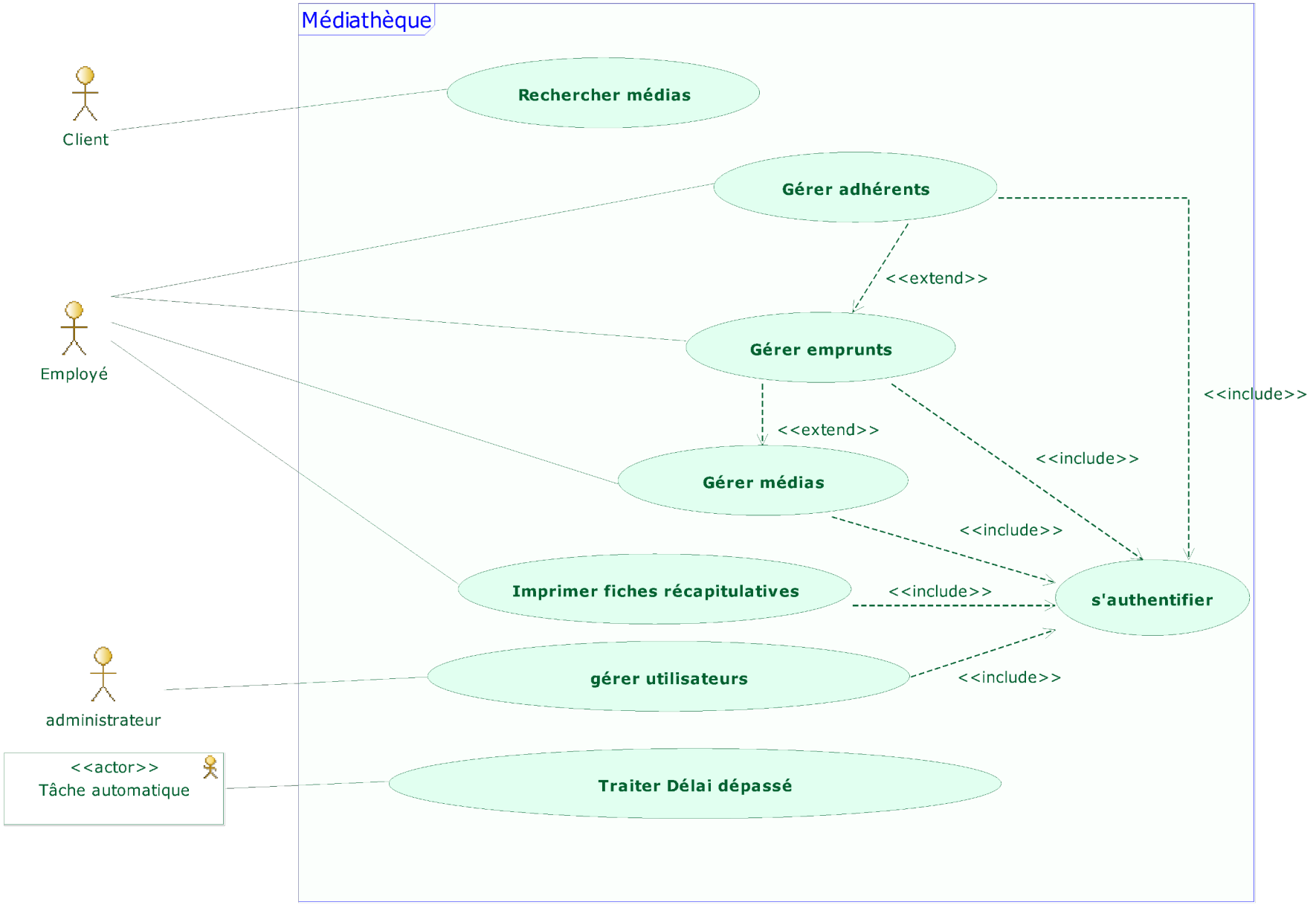
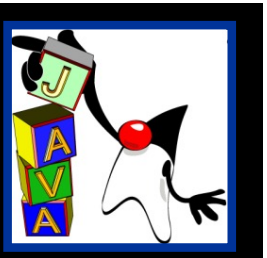


Diagramme de Cas d'utilisation
Description textuelle d'un UC



Sommaire d'identification

Titre	
Objectifs	***Description synthétique de l'objectif***
Acteurs	***Acteur principal (détient l'objectif) et acteurs secondaires***
Evènement déclencheur	***Le cas d'utilisation débute quand l'évènement se produit***
Préconditions	***conditions pour que l'exécution de l'UC puisse démarrer***
Postcondition	***conditions qui deviennent vraies à la fin de l'exécution normale de l'UC (scénario nominal ou alternatifs) sauf pour les exceptions***
Exigences spécifiques	***Exigences non fonctionnelles: performance, sécurité, ergonomie, …***

Description des scénarios

Scénario Nominal	***Scénario dans le cas où "tout se passe bien"***
Scénarios alternatifs	***Situations optionnelles***
Scénarios d'exception	***Description des évènements provoquant la terminaison prématurée de l'UC***

Description textuelle d'un UC
Exemple UC Gérer Emprunts

Sommaire d'identification	
Titre	**GERER EMPRUNTS**
Objectifs	L'employé de la médiathèque enregistre un emprunt ou un retour dans le système.
Acteurs	L'employé de la médiathèque.
Evènement déclencheur	Un adhérent se présente au guichet de la médiathèque pour effectuer un emprunt ou rendre un média.
Préconditions	L'employé est identifié dans le système (voir UC. « S'authentifier »).
Postcondition	Un nouvel emprunt ou retour a été enregistré dans le système

Description textuelle d'un UC
Exemple UC Gérer Emprunts

Description des scénarios

acteur

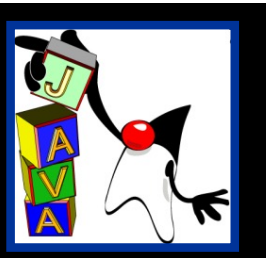
Scénario nominal

Ce cas d'utilisation commence quand l'employé choisit l'option « gérer les emprunts ».

Système (représente le logiciel à concevoir)

1. Le système demande à l'employé de saisir le numéro de l'adhérent qui souhaite emprunter ce média.
2. L'employé saisit le numéro de l'adhérent et valide.
3. Le système recherche l'adhérent.
4. Le système affiche un menu permettant à l'employé de choisir la fonction qu'il souhaite exécuter.
5. L'employé indique la fonction qu'il désire exécuter (enregistrer emprunt ou enregistrer retour)
6. Selon le choix de l'employé un des sous-scénarios est exécuté:
   - P: Enregistrer un emprunt
   - R: Enregistrer un retour

Description textuelle d'un UC
Exemple UC Gérer Emprunts



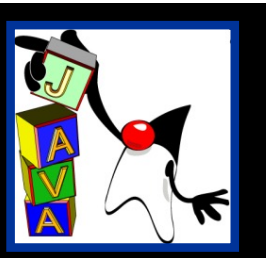
Scénario nominal (suite)

Sous-scénario P : Enregistrer un emprunt

Ce sous-scénario démarre quand l'employé sélectionne l'option « saisir nouvel emprunt ».

1. Le système vérifie que le nombre de média emprunté par l'adhérent est strictement inférieur à 3
2. Le système demande à l'employé de saisir la référence du média concerné par l'emprunt.
3. L'employé saisit la référence et valide.
4. Le système vérifie la disponibilité du média.
5. Le système affiche les informations relatives à l'emprunt (adhérent et média concernés, date de l'emprunt) et demande une confirmation.
6. L'employé confirme.
7. Le système met à jour la disponibilité du média et incrémente le nombre d'emprunts de l'adhérent
8. Le système enregistre l'emprunt et affiche le numéro d'emprunt généré.
9. L'employé ferme la gestion des emprunts
10. Le système affiche le menu général

Description textuelle d'un UC
Exemple UC Gérer Emprunts



Scénarios alternatifs

A1 - En P.2, si aucun exemplaire n'est disponible en médiathèque pour ce média,

1 - Le système affiche un message d'erreur,

2 - Le UC reprend en P.2.

A2 - En P.4, si aucun exemplaire n'est disponible en médiathèque pour ce média,

1 - Le système affiche un message d'erreur,

2 - Le UC reprend en P.2.

A3 - En P.9, si l'employé désire indiquer l'emprunt de plusieurs médias, le UC reprend en P.2.

Description textuelle d'un UC
Exemple UC Gérer Emprunts

Scénarios d'exception

E1 - En 3, si l'adhérent n'existe pas dans le système, le système affiche un message et le UC se termine.

E2 - En P.1, si l'adhérent a déjà 3 emprunts en cours, le système affiche un message et le UC se termine.

E3 - En P.6, si l'employé décide de ne pas valider l'emprunt (annulation), le système affiche un message et le UC se termine.

Formalisme UML

- Qu'est ce qu'UML?
- Diagramme de classes
- Diagramme de Cas d'Utilisation
- Diagramme de séquence
  - Objets et messages
  - Création/destruction d'objets
  - Messages et classes
  - Cadres d'interaction

Diagramme de Séquence

Pour montrer comment est réalisée une fonctionnalité du système d'un point de vue temporel

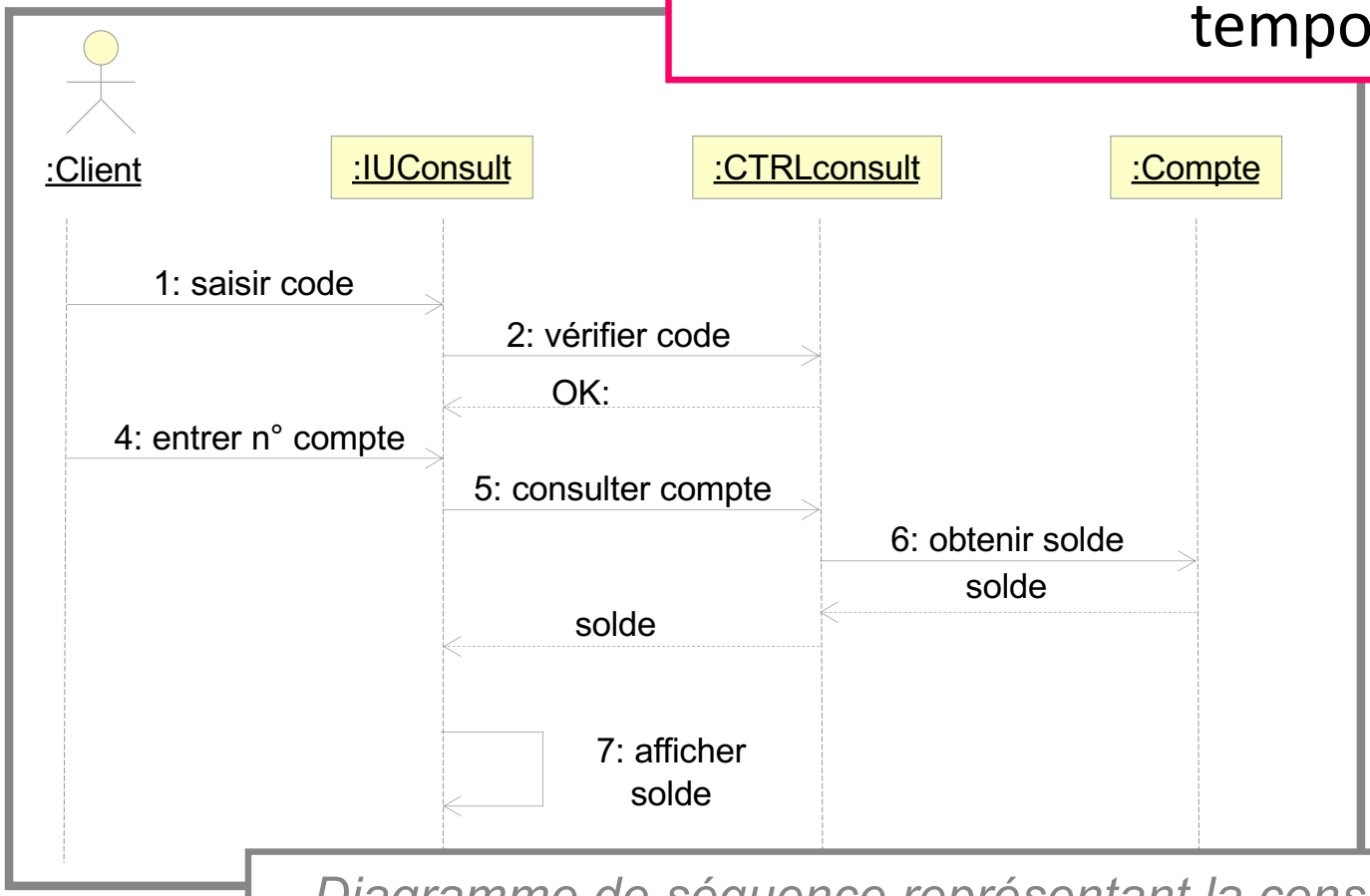
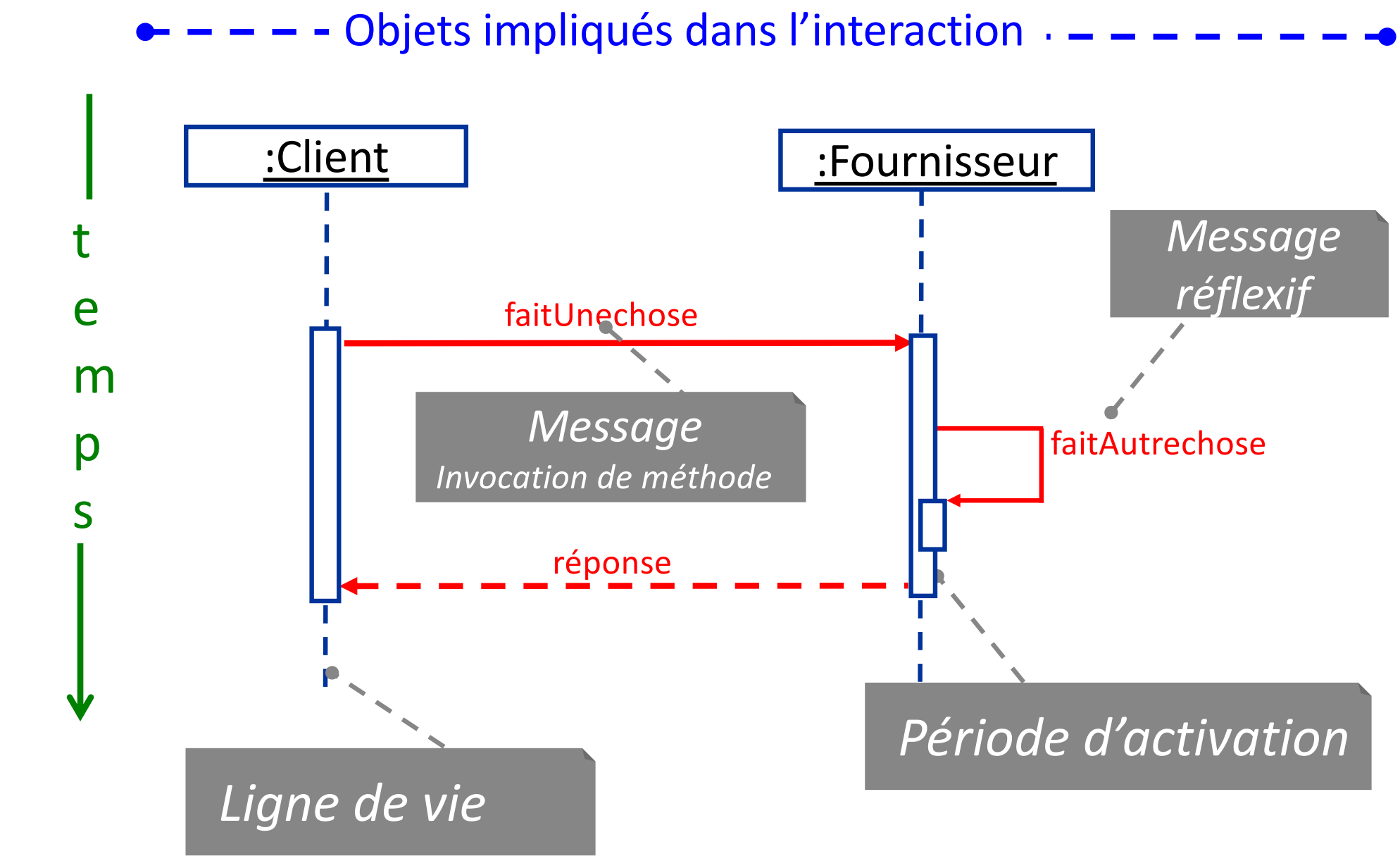
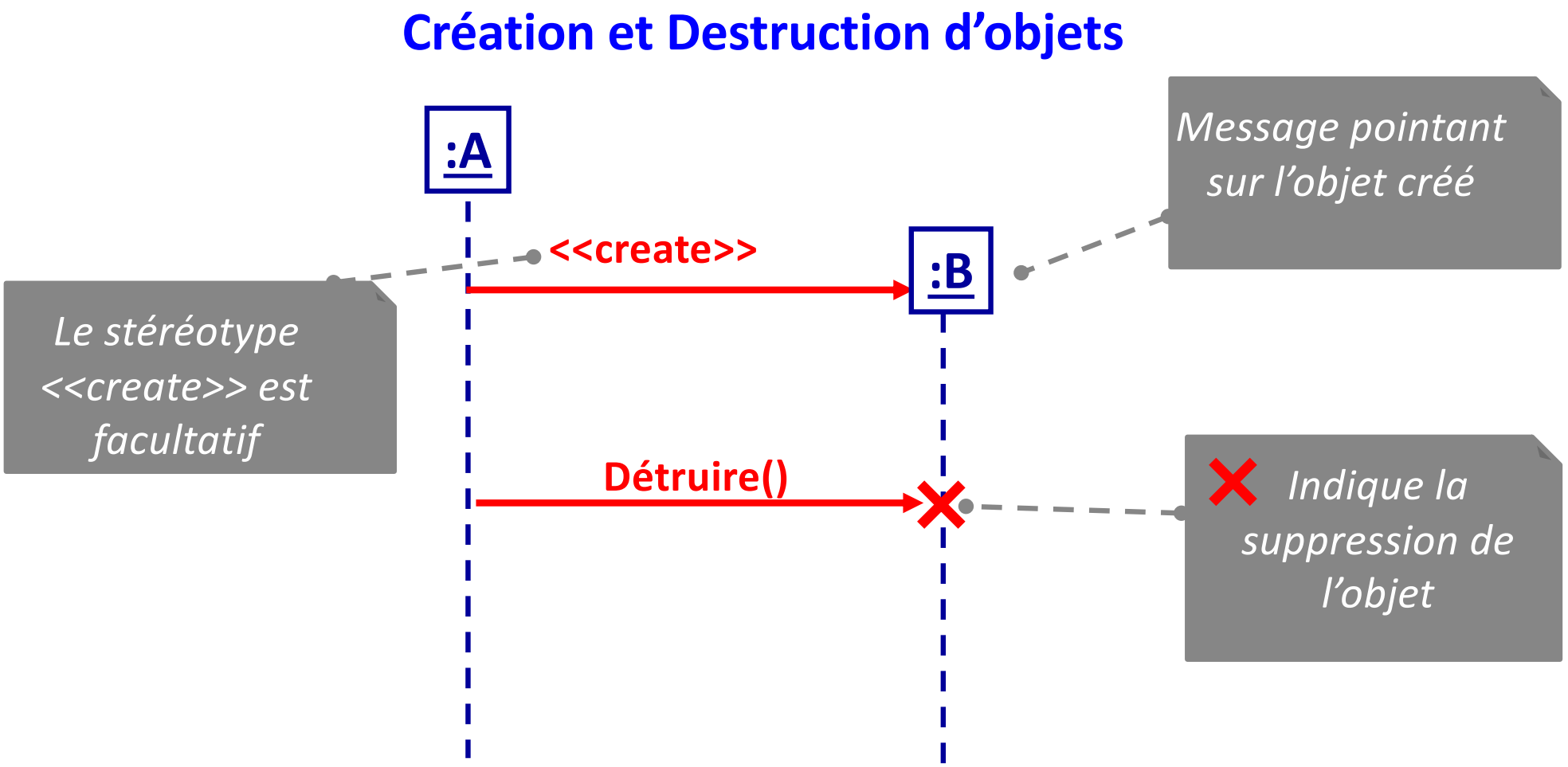


Diagramme de séquence représentant la consultation d'un Compte

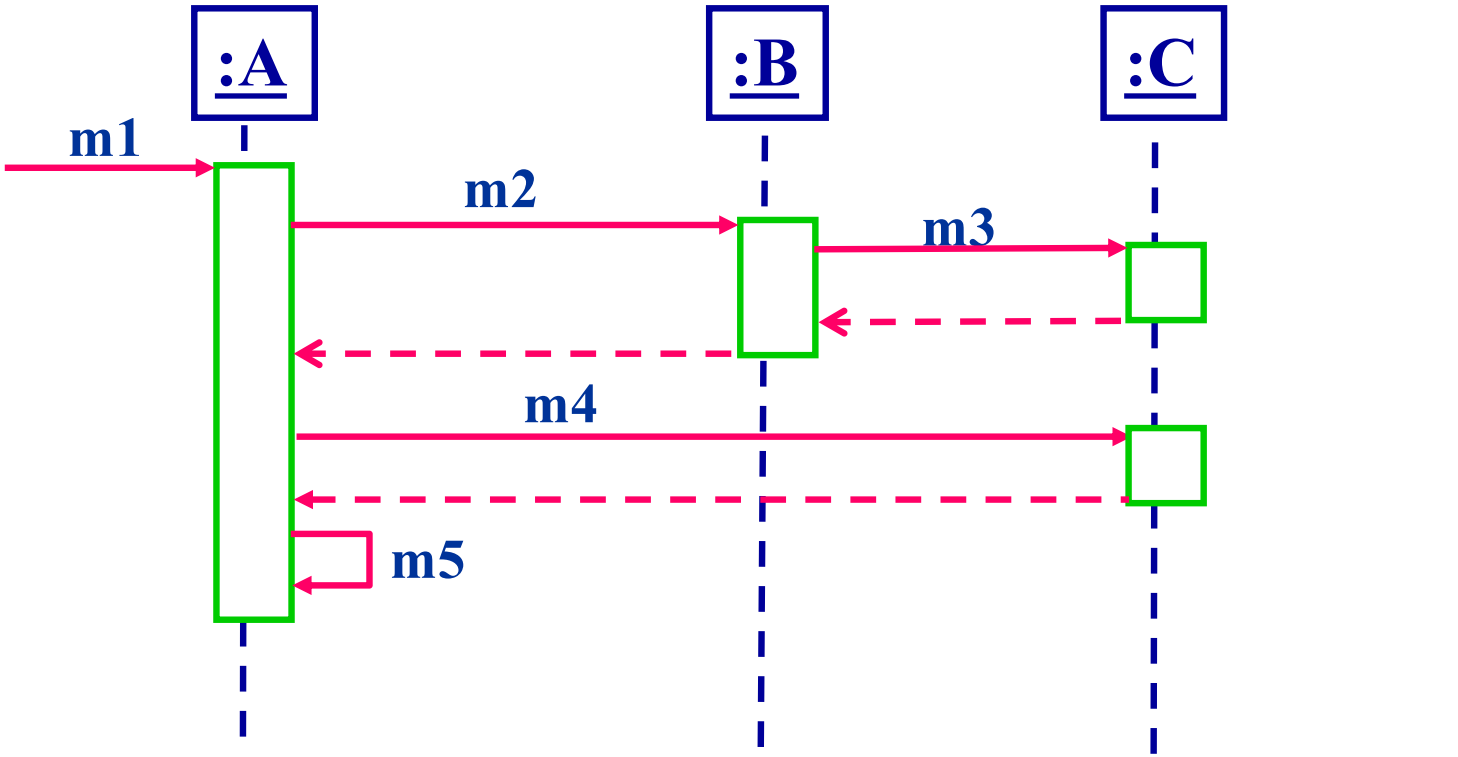
Objets et messages

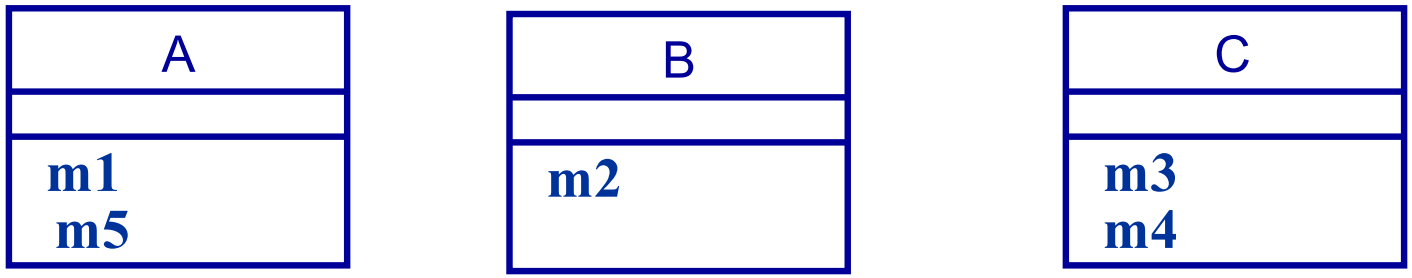


Création/Destruction d'objets

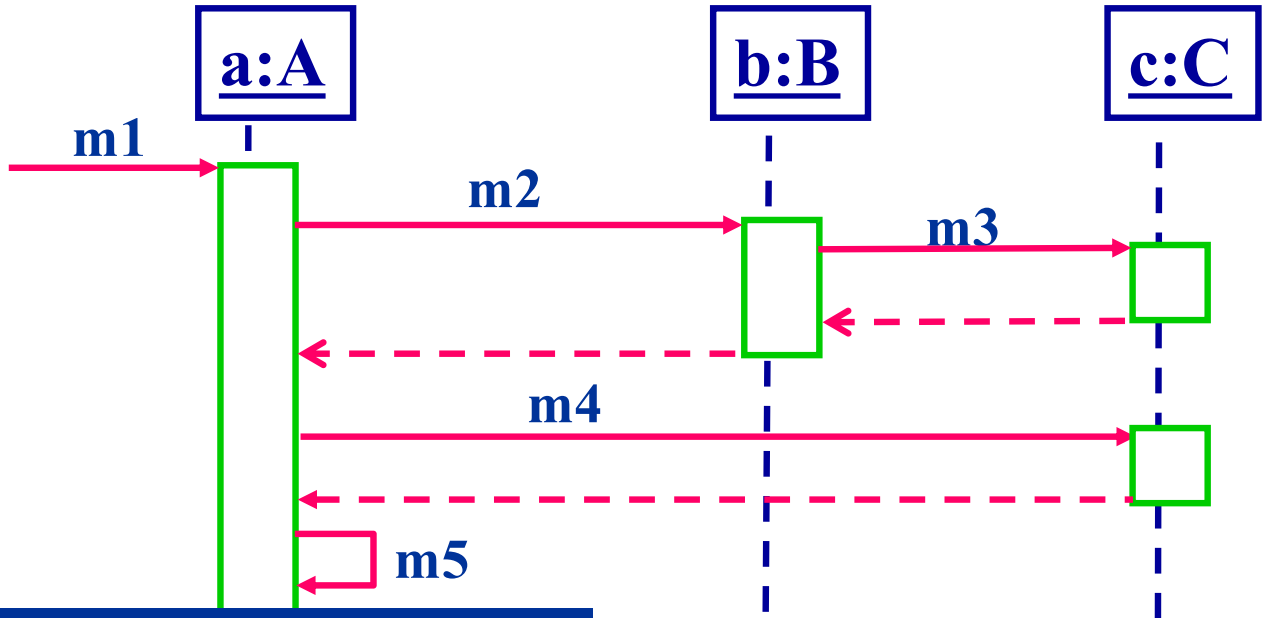


Messages et classes





Messages et classes



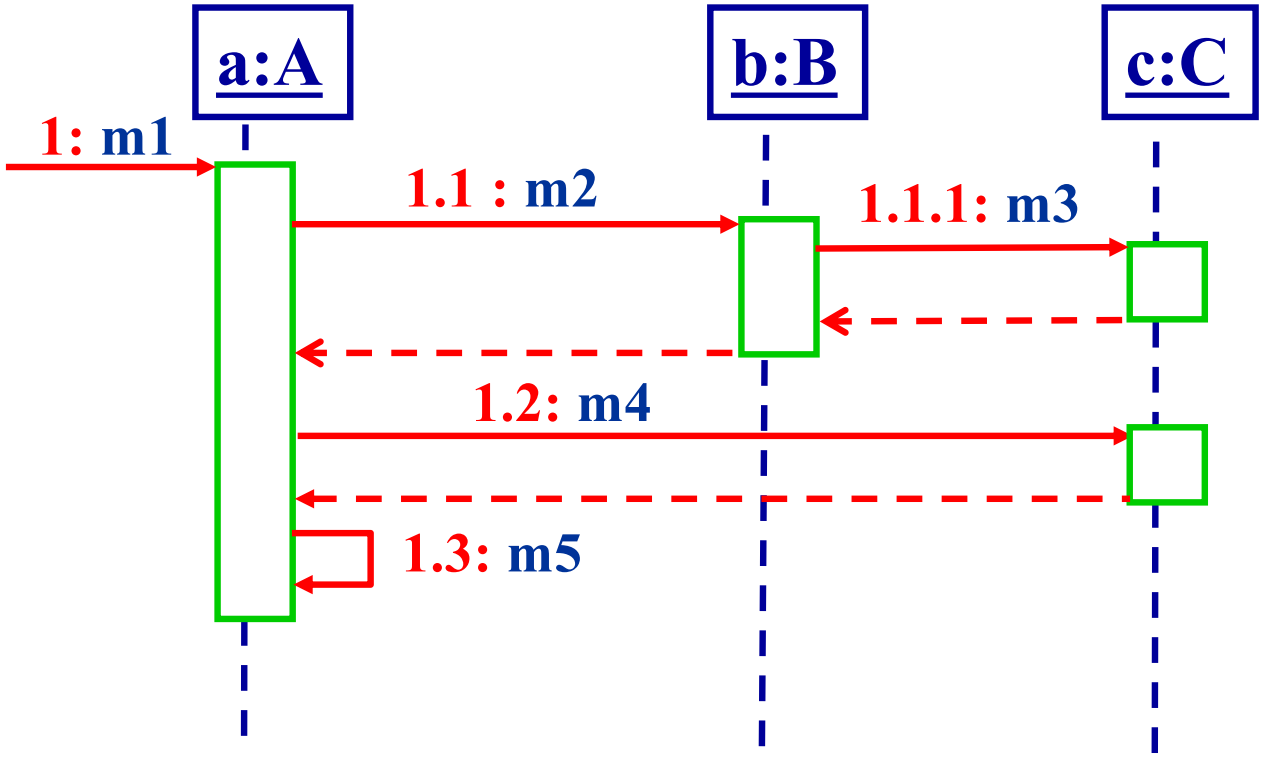

```
Class A {
void m1() {
        b.m2(); ….
        c.m4();   ….
        m5();   …}
void m5(){ ….}
```

```
Class B {
void m2() {
        c.m3();
…. }
```

```
Class C {
void m3() {
…. }
void m4()
{….} …
```

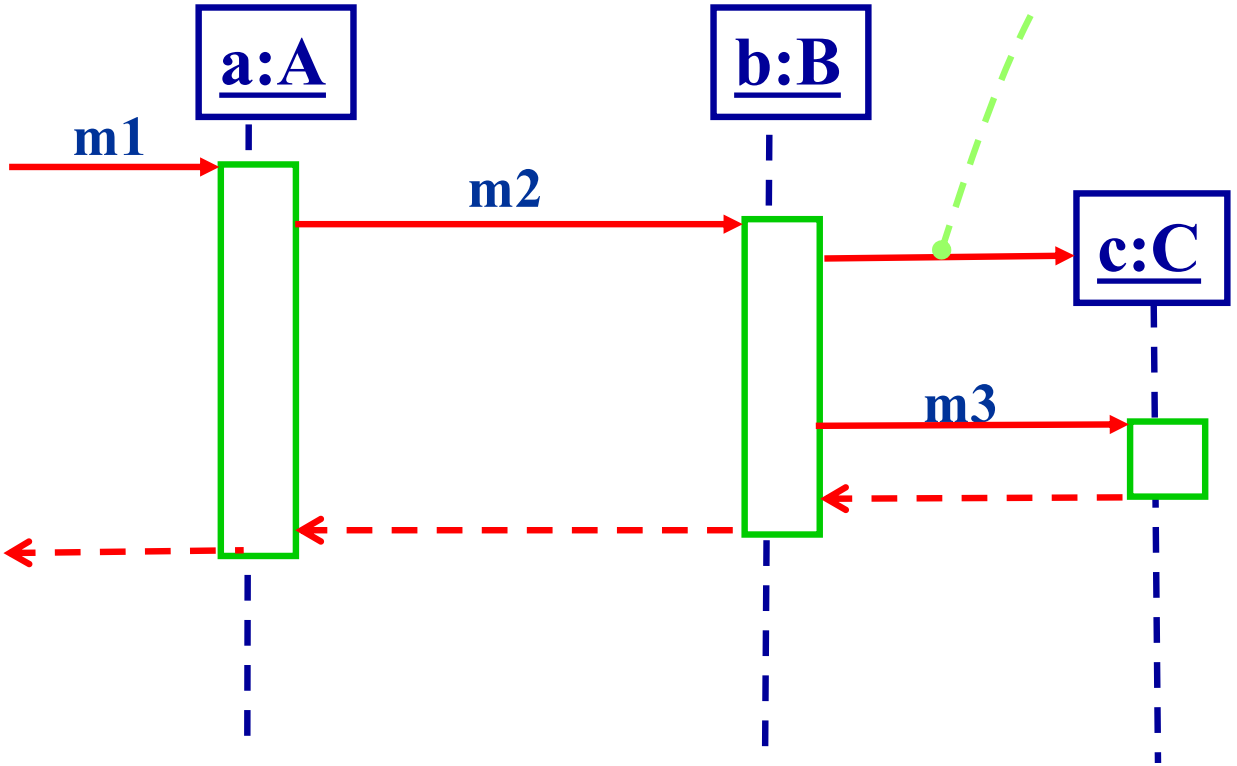
Messages et classes

Numérotation hiérarchique des messages



Messages et classes

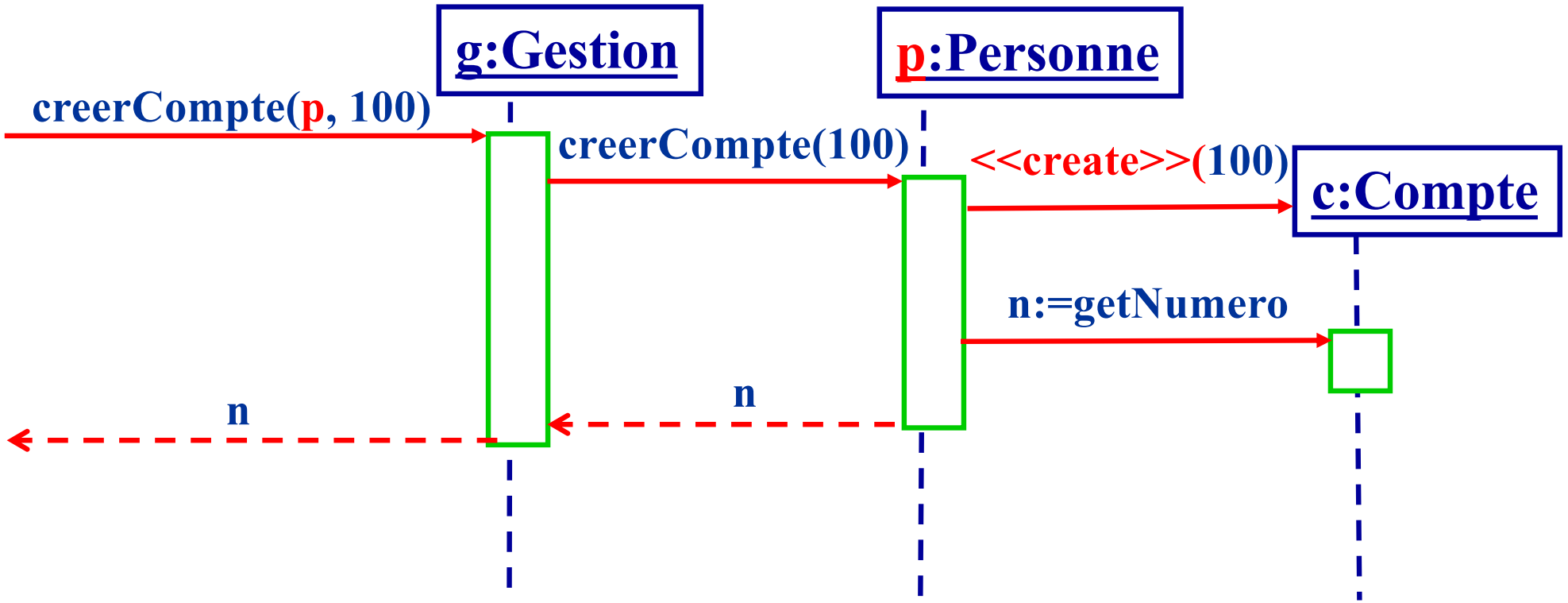
Création d'objets




```
Class B {
void m2() {
      c=new C();
      c.m3(); ….
}
```

Messages et classes

Paramètres et Résultats

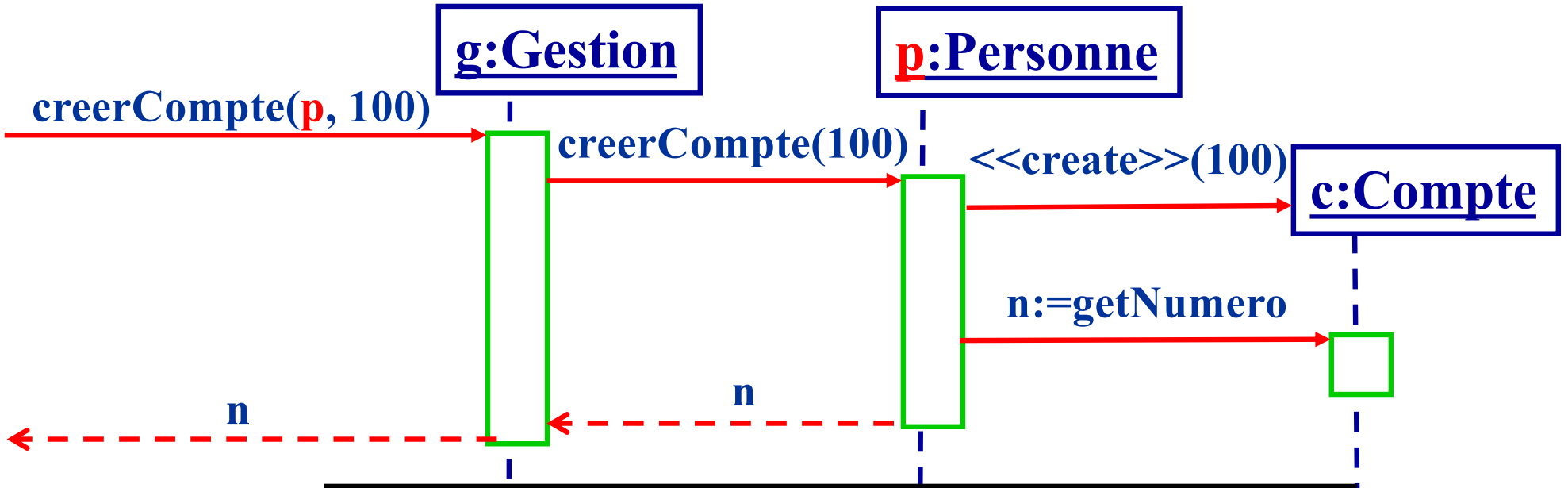


Syntaxe des messages

numeroOrdre: Valeur-retournée := nom-message(arguments)

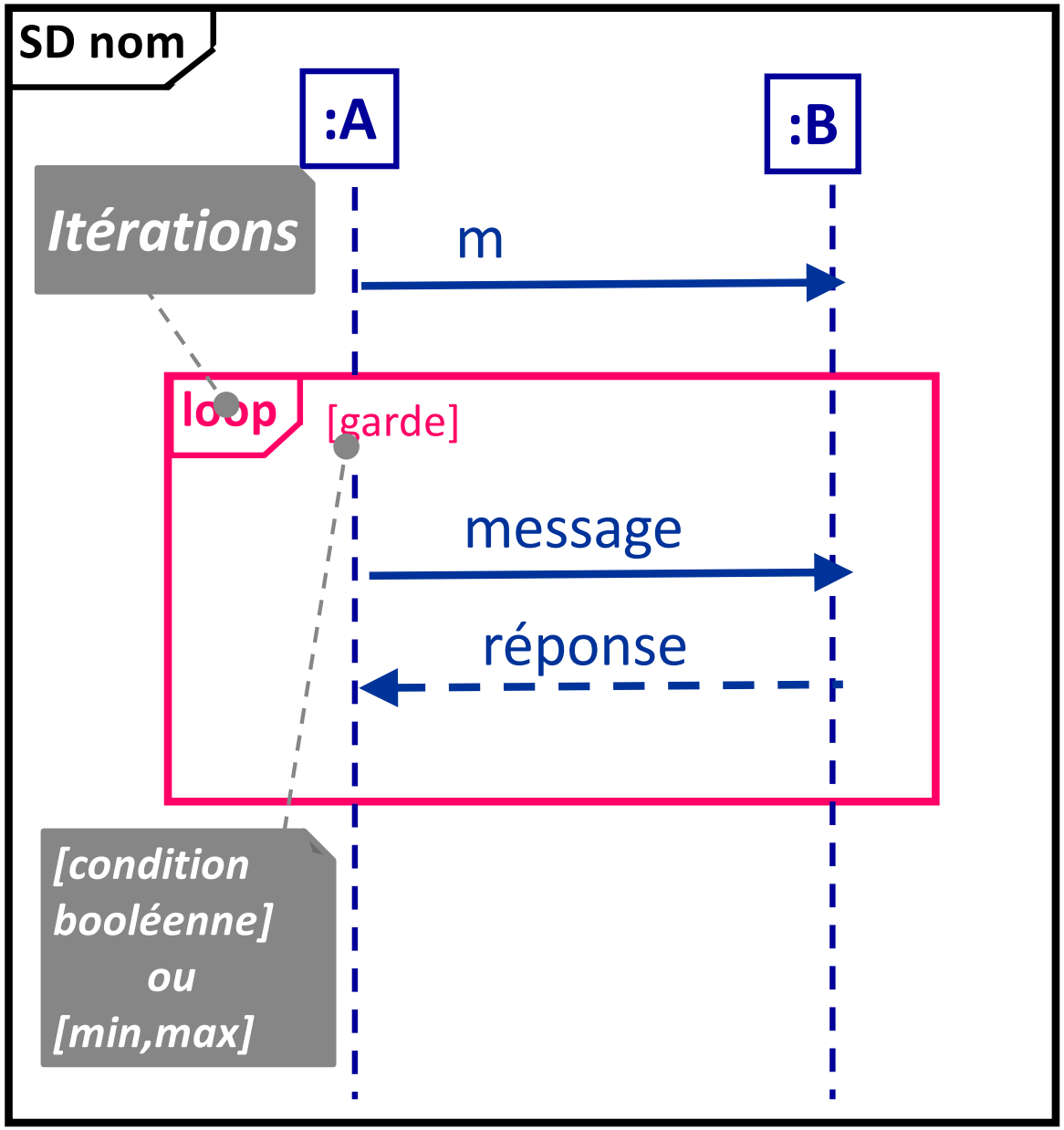
Messages et classes

Paramètres et Résultats

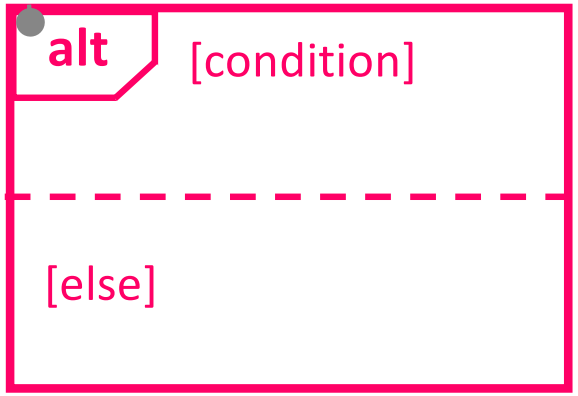


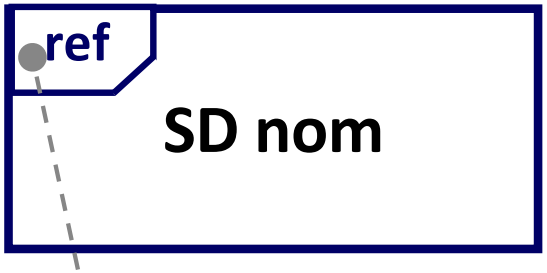

```
class Personne {
int creerCompte(int m) {
      c=new Compte(m);
      return c.getNumero(); }
```

Cadres d'intéraction (UML 2)



Schémas alternatifs



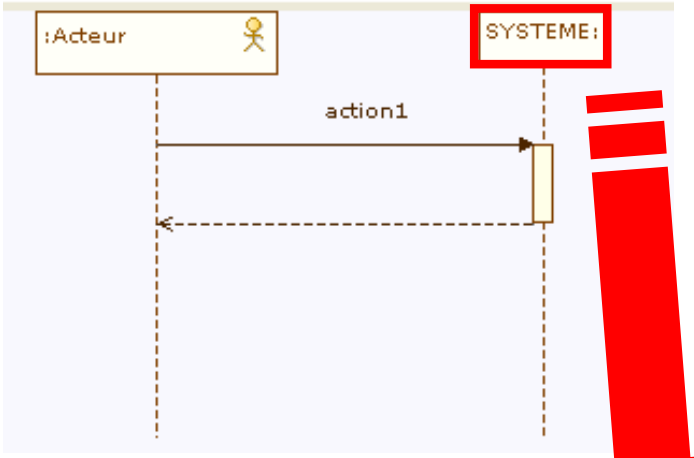


Référence à un autre SD

Diagramme de Séquence

Deux niveaux d'utilisation:

- **DSS**
  - **Diagrammes de séquence Système**
  - **Interactions Acteurs/Système**



- **Diagrammes de séquence d'analyse**
  - **Interactions entre objets**

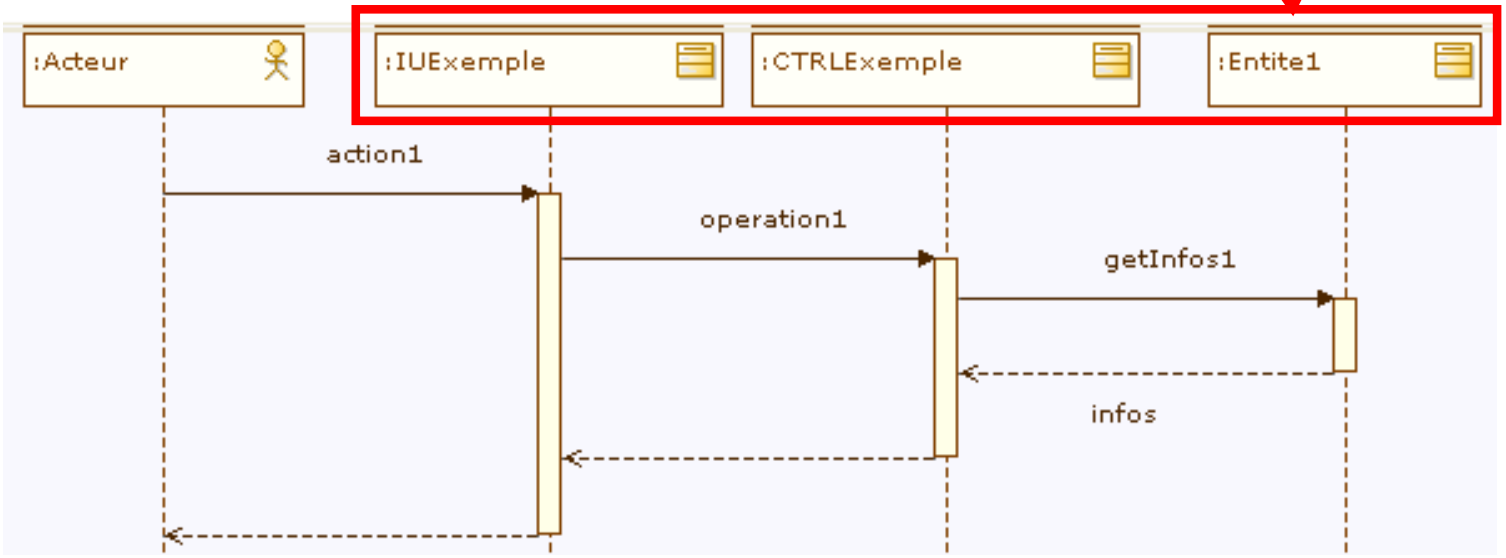
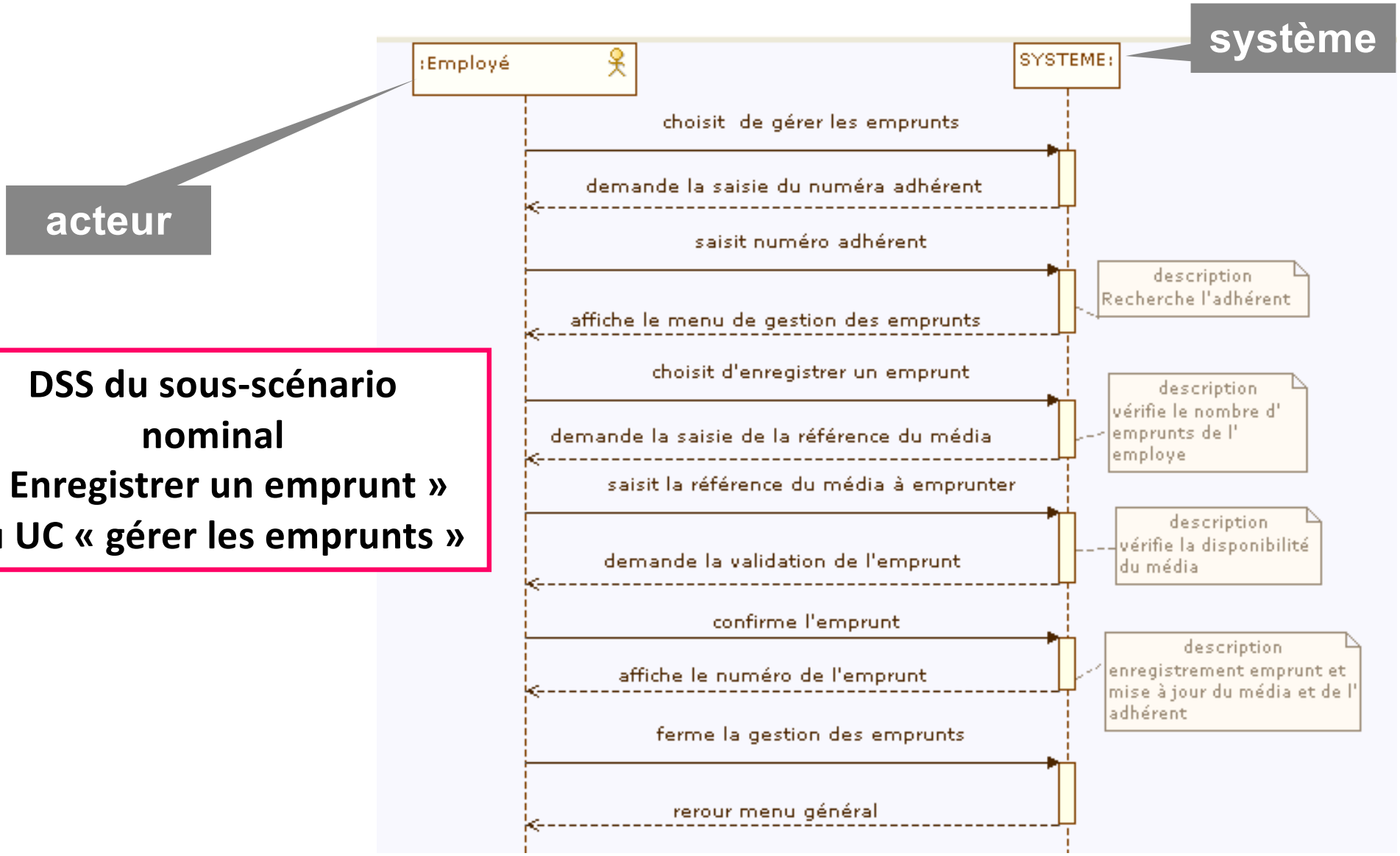


Diagramme de Séquence

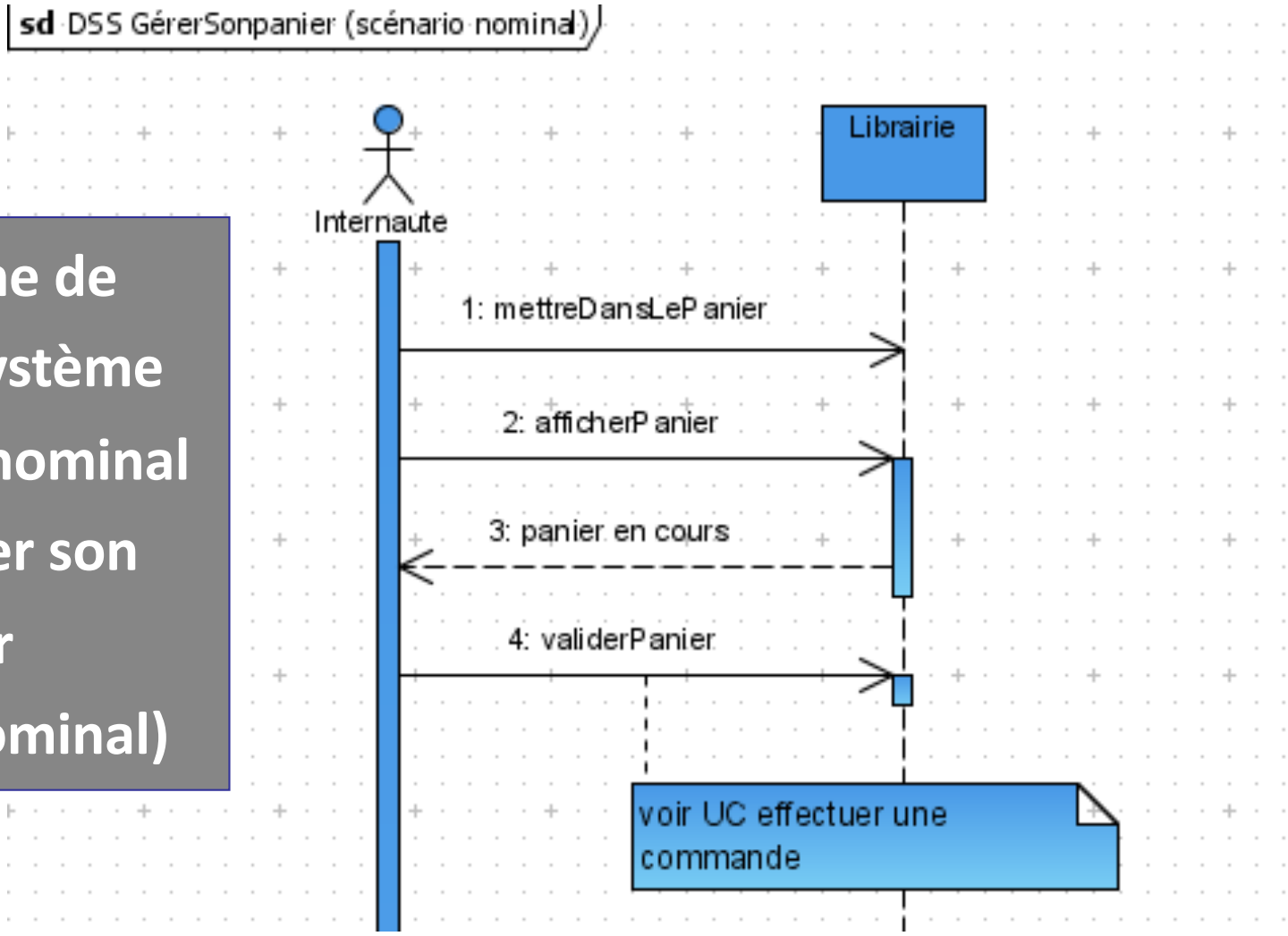
DSS : Diagramme de Séquence Système



DSS du sous-scénario nominal
« Enregistrer un emprunt »
du UC « gérer les emprunts »

Exemple de DSS

Diagramme de séquence système du scénario nominal du UC Gérer son panier (scénario nominal)



Exemple de DS d'analyse

Objets
Instances de classes

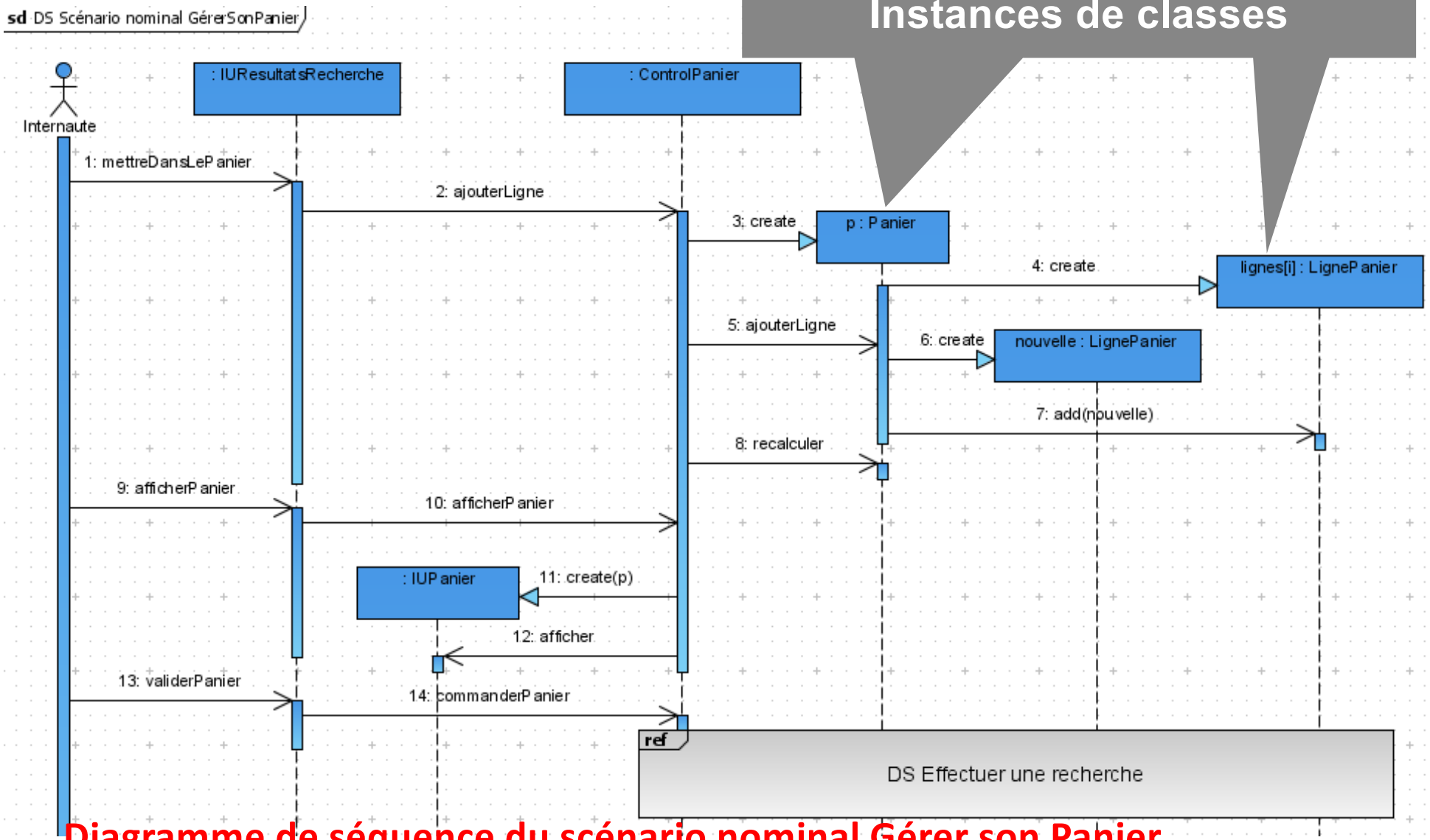


Diagramme de séquence du scénario nominal Gérer son Panier