



## **CHAPITRE II:** **Modélisation avec UML**



Les diagrammes fonctionnels (comportement)

**Diagramme de cas d'utilisation** (statiques)

**Diagramme de séquence** (dynamique)

**Diagramme d'activités** (dynamique)

## **Diagramme de séquence**

# Diagramme de séquence

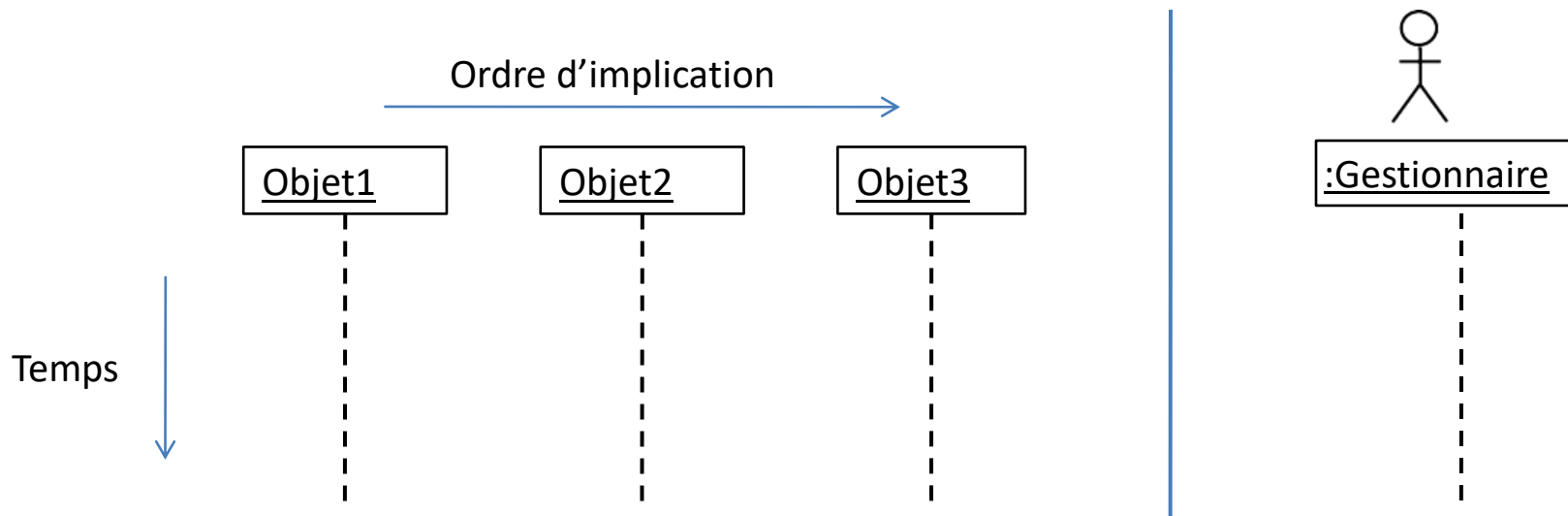
## Définition

- ❖ L'objectif du diagramme de séquence est de **représenter les interactions entre objets** en indiquant la **chronologie** des échanges.
- ❖ Cette représentation **peut se réaliser par cas d'utilisation** en considérant **les différents scénarios** associés.
- ❖ Un diagramme de séquences est un diagramme **d'interaction** qui expose en **détail la façon dont les opérations sont effectuées**: quels **messages** sont envoyés et **quand** ils le sont.
- ❖ L'accent est mis sur la **communication**.
- ❖ Les diagrammes de séquences sont **organisés en fonction du temps qui s'écoule**.

# Diagramme de séquence

## Représentation

- Chaque **objet** est représenté par une **barre verticale**
- Le **temps** s'écoule de **haut en bas**, de sorte que la **numérotation** des **messages** est **optionnelle**.
- Les objets **impliqués** dans l'opération sont répertoriés de **gauche à droite** en fonction du moment où ils prennent part dans la séquence.



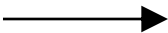
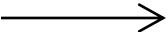
# Diagramme de séquence

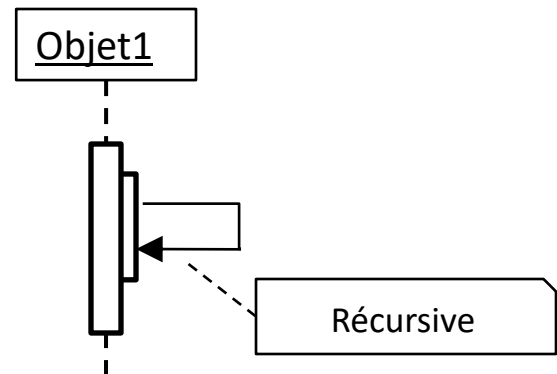
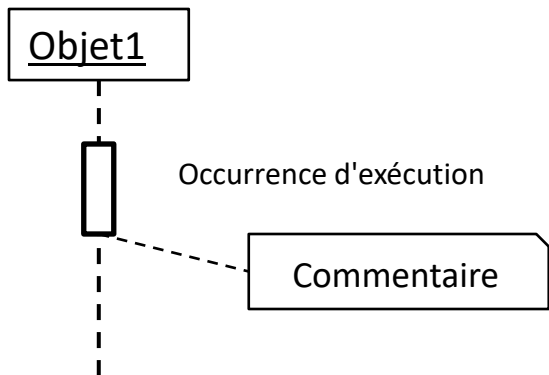
## Représentation

- ❖ Ce type des diagrammes **est composé** par les éléments suivants :
  - **Les lignes de vie** : Une ligne verticale qui **représente la séquence des événements**, produite par un participant, pendant une interaction, alors que le temps progresse en bas de ligne.
- ❖ Ce **participant** peut être une **instance** d'une **classe**, un **composant** ou un **acteur**.

# Diagramme de séquence

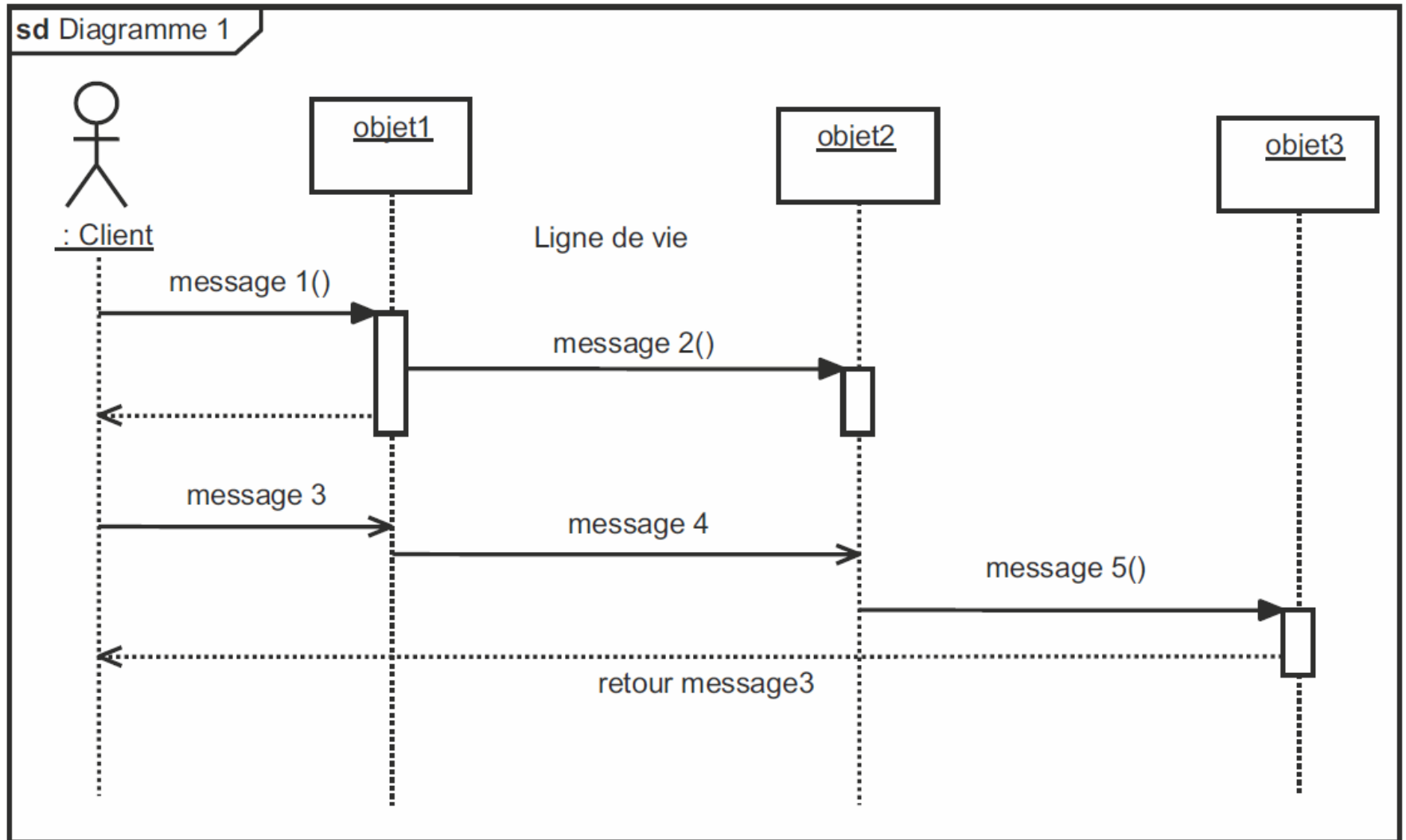
## Représentation

- **Les messages**
  - le **message synchrone**: l'émetteur reste en **attente de la réponse** à son message avant de poursuivre ses actions. 
  - pour représenter des appels de **fonction ordinaires** dans un programme.
  - le **message asynchrone**: l'émetteur **n'attend pas la réponse** à son message, **il poursuit** l'exécution de ses opérations. 
- **Les occurrences d'exécution** : représente la période d'exécution d'une opération.
- **Les commentaires** : Un commentaire peut être joint à tout point sur une ligne de vie.
- **Les itérations** : représente un message de **réponse** suite à une question de vérification.



# Diagramme de séquence

## Représentation

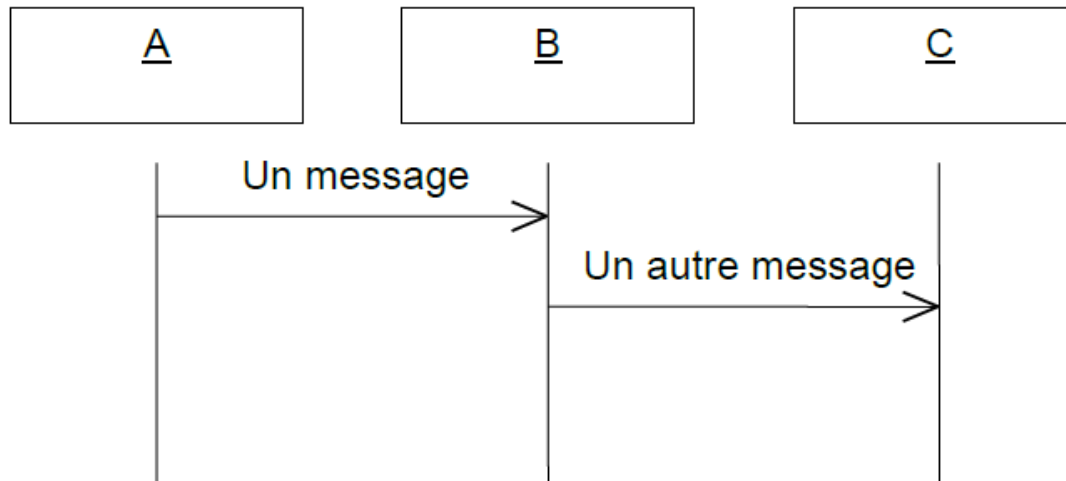




# Diagramme de séquence

## Représentation

Représentation temporelle

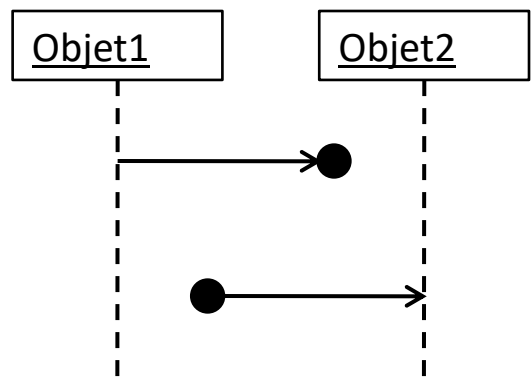


# Diagramme de séquence

## Représentation

### Les messages

- **le message perdu**: un message envoyé, mais **n'arrive pas au destinataire**, ou qui va chez un destinataire non montré sur le diagramme actuel.
- **le message trouvé**: un message qui arrive d'un **expéditeur inconnu**, ou d'un **expéditeur non montré sur le diagramme** actuel.



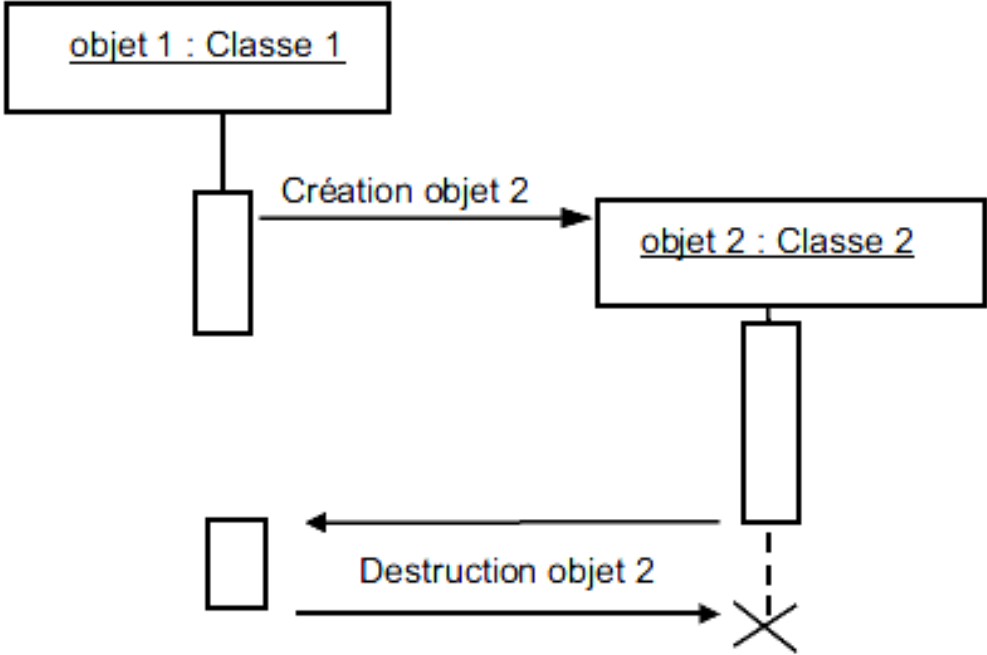
# Diagramme de séquence

## Représentation

### Opérations particulières

#### *Création et destruction d'objet*

- Si un **objet** est **créé par une opération**, celui-ci **n'apparaît qu'au moment** où il est **créé**.
- Si l'objet est **détruit par une opération**, la destruction se représente par « **X** ».



# Diagramme de séquence

## Représentation

### Opérations particulières

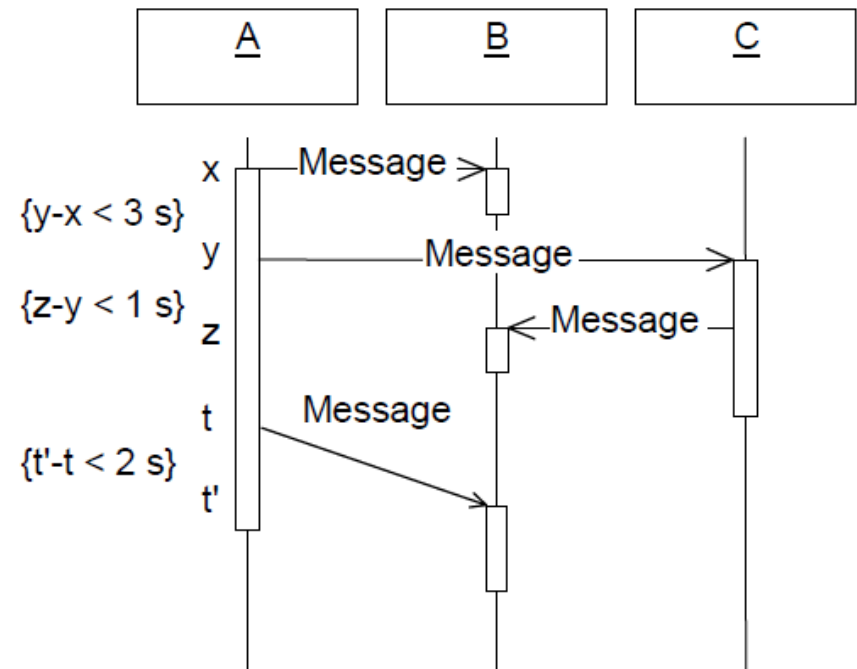
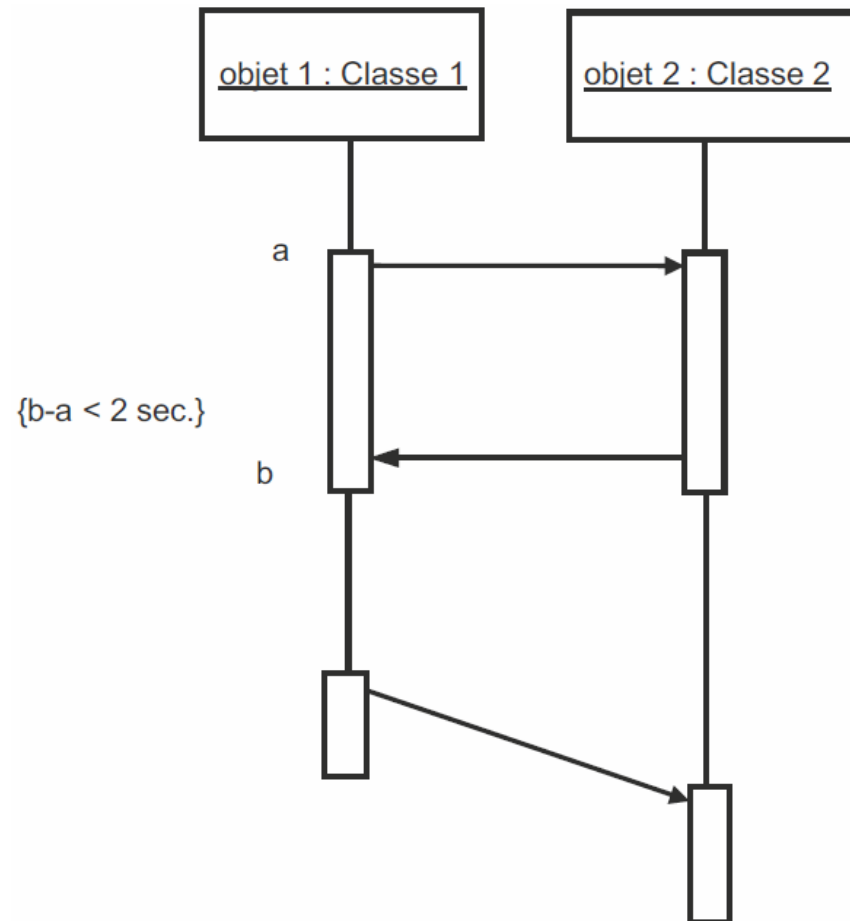
#### *Contrainte temporelle*

- Des **contraintes** de chronologie **entre les messages** peuvent être **spécifiées**.
- De plus lorsque l'émission d'un message requiert une **certaine durée**, il se représente sous la forme d'un **trait incliné**.

# Diagramme de séquence

## Représentation

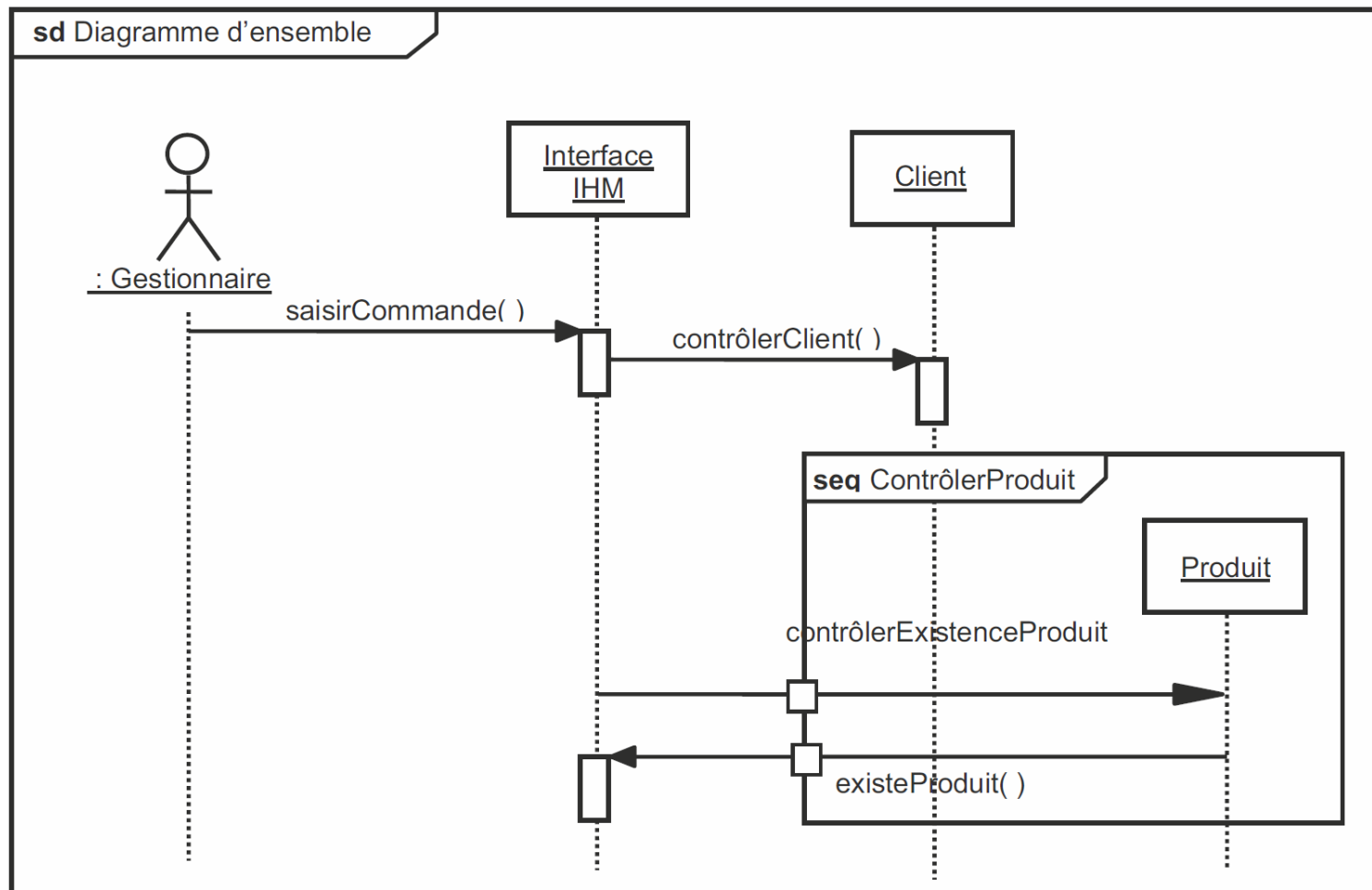
Opérations particulières  
*Contrainte temporelle*



# Diagramme de séquence

## Fragments d'interaction

Dans un DS, il est possible de distinguer des **sous-ensembles** d'interactions qui constituent des **fragments**.



# Diagramme de séquence

## Fragments d'interaction

### *Remarque*

- Un **port d'entrée** et un **port de sortie** peuvent être indiqués pour connaître la manière dont ce fragment peut être relié au reste du diagramme.
- Si aucun port n'est indiqué c'est **l'ensemble** du fragment qui est **appelé pour exécution**.

# Diagramme de séquence

## Fragments d'interaction combiné

- Un **fragment** d'interaction dit **combiné**; correspond à un ensemble d'interaction auquel on applique un **opérateur**.
- Un **fragment combiné** se représente globalement comme un diagramme de séquence avec **indication** dans le **coin à gauche** du **nom de l'opérateur**.

*Treize opérateurs ont été définis dans UML :*

**alt, opt, loop, par, strict/weak, break, ignore/consider, critical, negative, assertion et ref.**



# Diagramme de séquence

## Les opérateurs

### *Opérateur alt*

- Correspond à une instruction de **test** avec **une** ou **plusieurs alternatives** possibles.
- L'opérateur **alt** se représente dans un **fragment possédant au moins deux parties séparées** par des pointillés.

### *Opérateur opt*

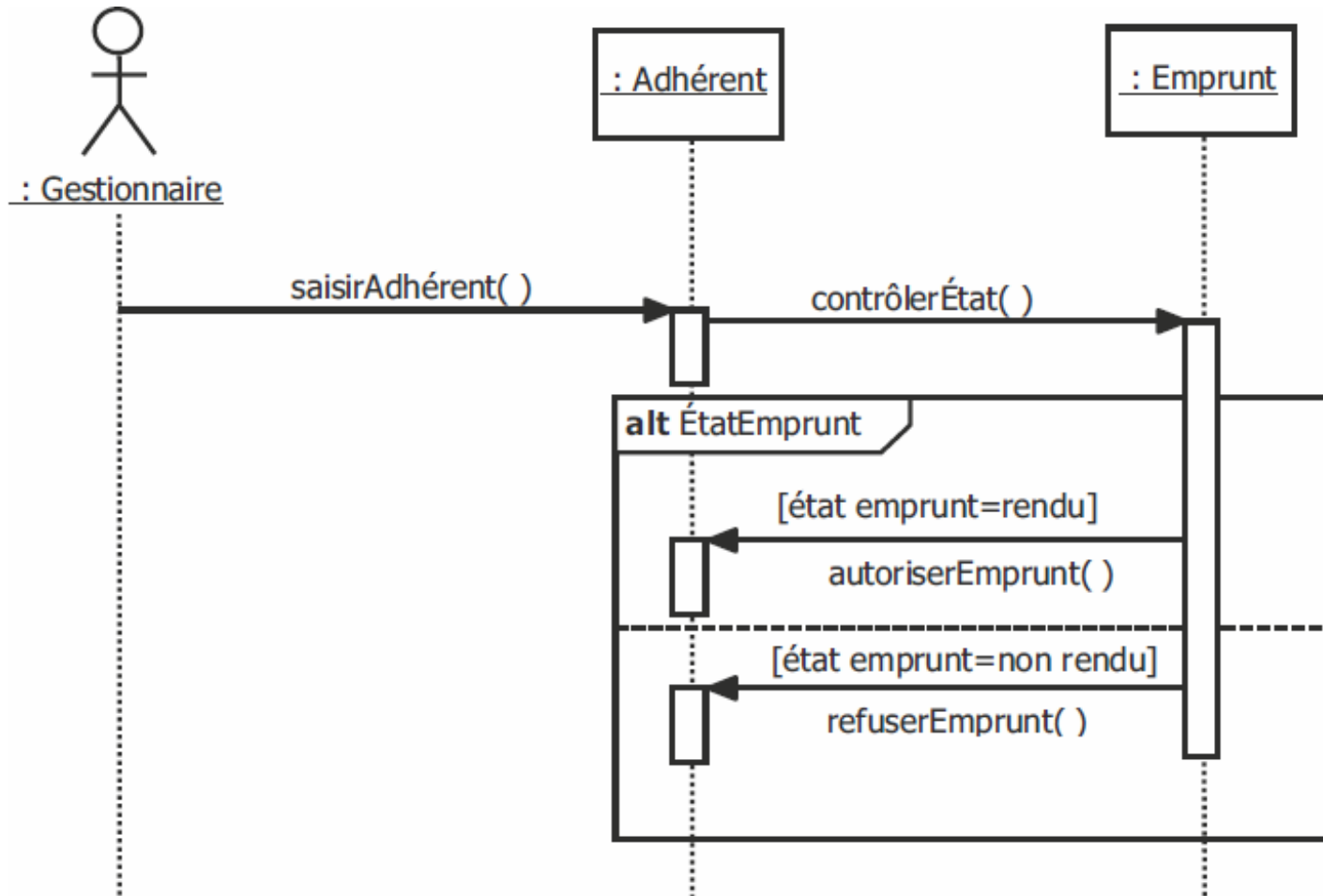
- **opt (optional)** correspond à une instruction de **test sans alternative (sinon)**.
- L'opérateur **opt** se représente dans un **fragment possédant une seule partie**.

### *Opérateur loop*

- Correspond à une instruction de **boucle** qui permet d'exécuter une séquence d'interaction tant qu'une condition est satisfaite.
- Il est possible aussi d'utiliser une condition portant sur un **nombre minimum** et **maximum** d'exécution de la boucle en écrivant : **loop min, max**.
- Dans ce cas, la boucle s'exécutera au minimum **min fois** et au maximum **max fois**.

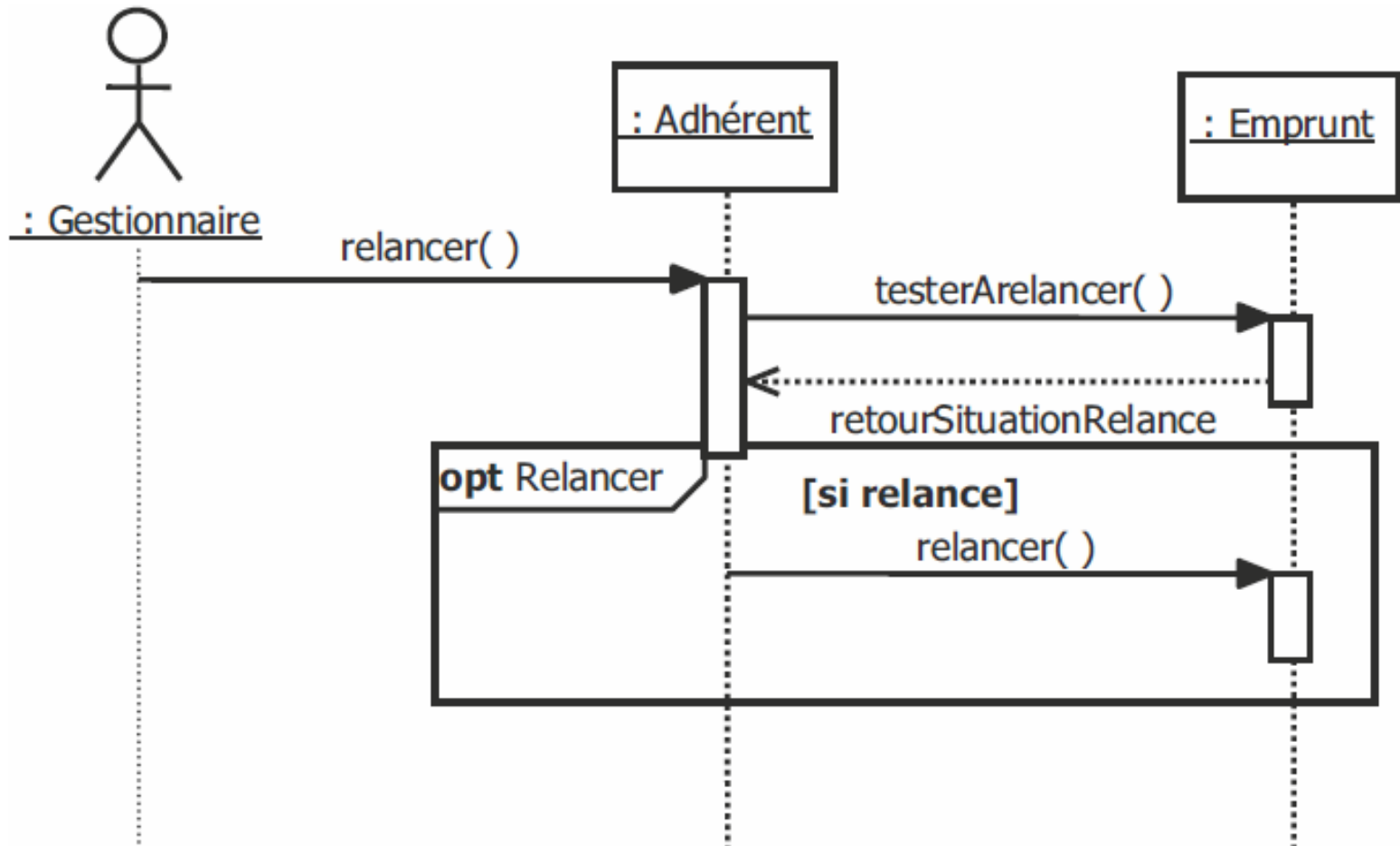
# Diagramme de séquence

## Les opérateurs



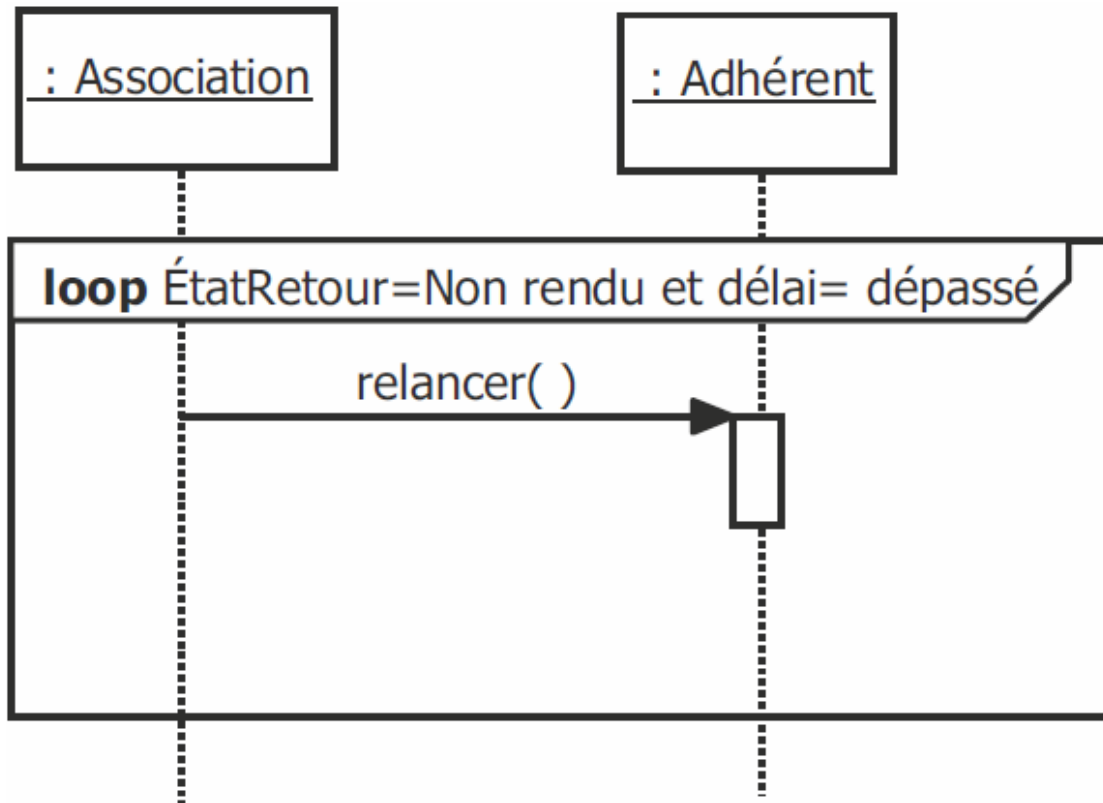
# Diagramme de séquence

## Les opérateurs



# Diagramme de séquence

## Les opérateurs



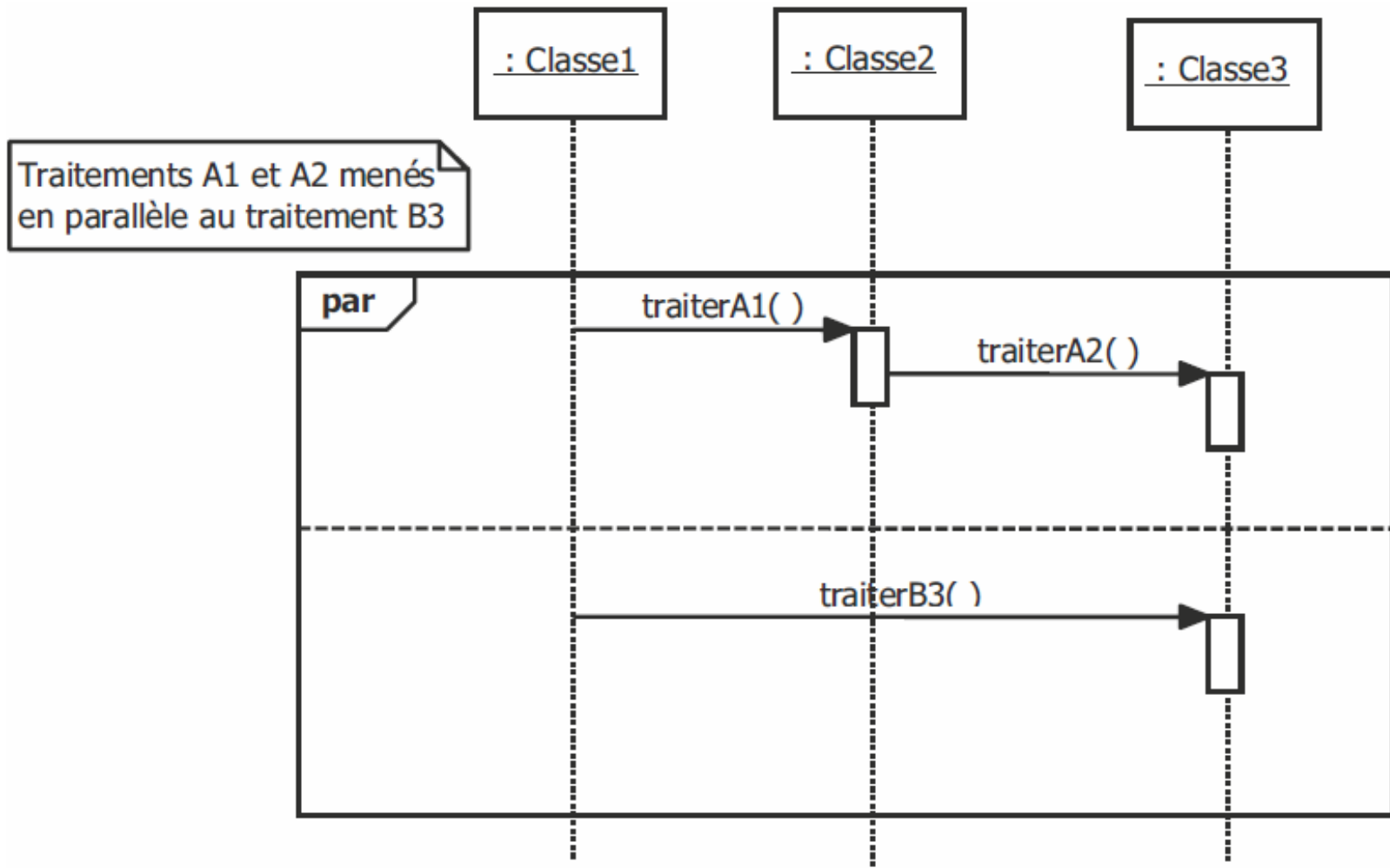
# Diagramme de séquence

## Les opérateurs

- *Opérateur par*
- L'opérateur **par** (**parallel**) permet de représenter **deux séries** d'interactions qui se déroulent en parallèle.
- L'opérateur **par** se représente dans un fragment possédant **deux parties séparées** par une ligne en pointillé.

# Diagramme de séquence

## Les opérateurs



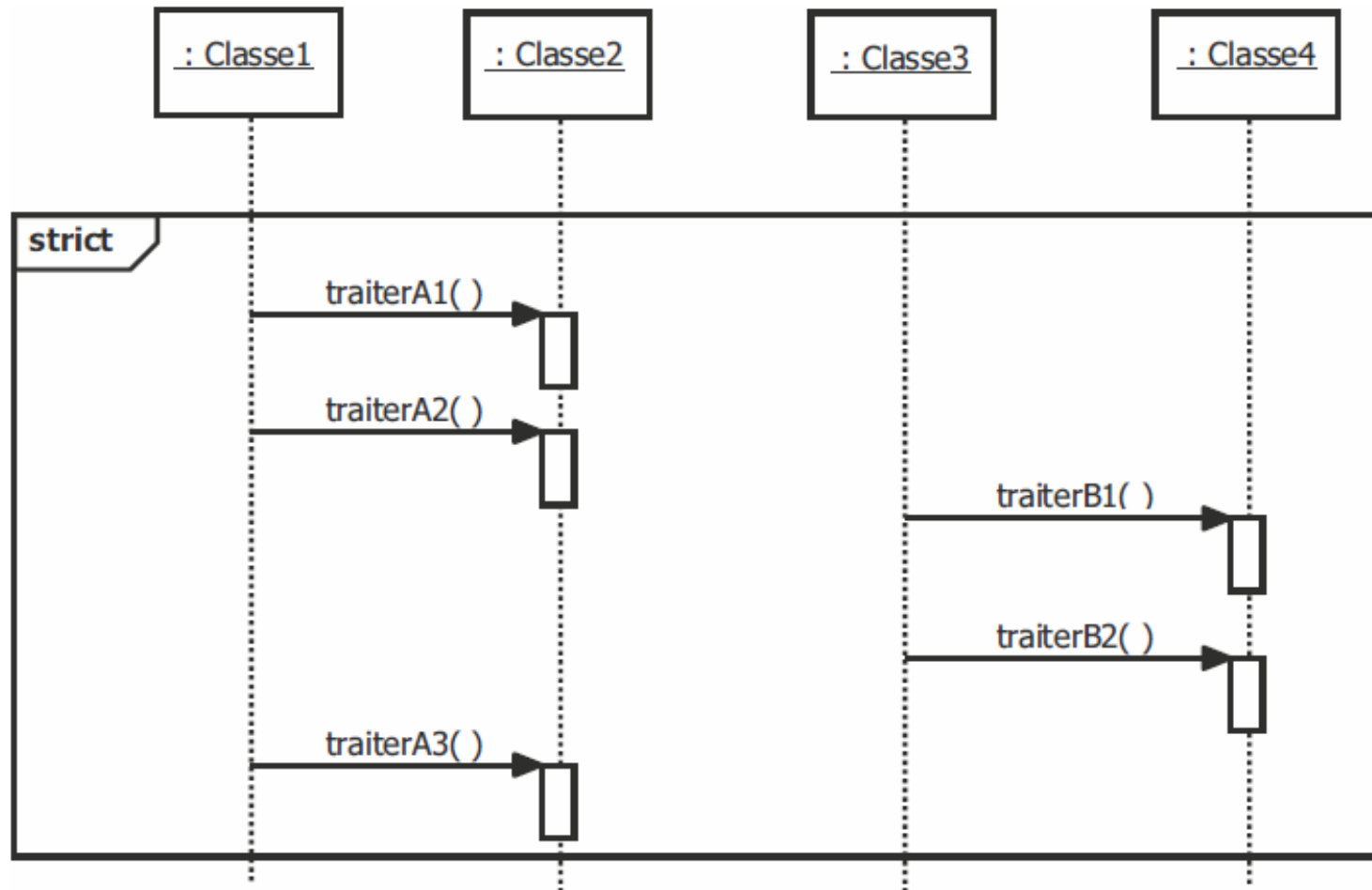
# Diagramme de séquence

## Les opérateurs

- *Opérateur strict et weak sequencing*
- Représentent **une série d'interactions dont certaines s'opèrent sur des objets indépendants** :
- L'opérateur **strict** est utilisé quand **l'ordre** d'exécution des opérations doit être **strictement respecté**.
- L'opérateur **weak** est utilisé quand **l'ordre** d'exécution des opérations **n'a pas d'importance**.

# Diagramme de séquence

## Les opérateurs



Les opérations A1, A2, B1, B2 et A3 doivent être exécutées dans cet ordre.



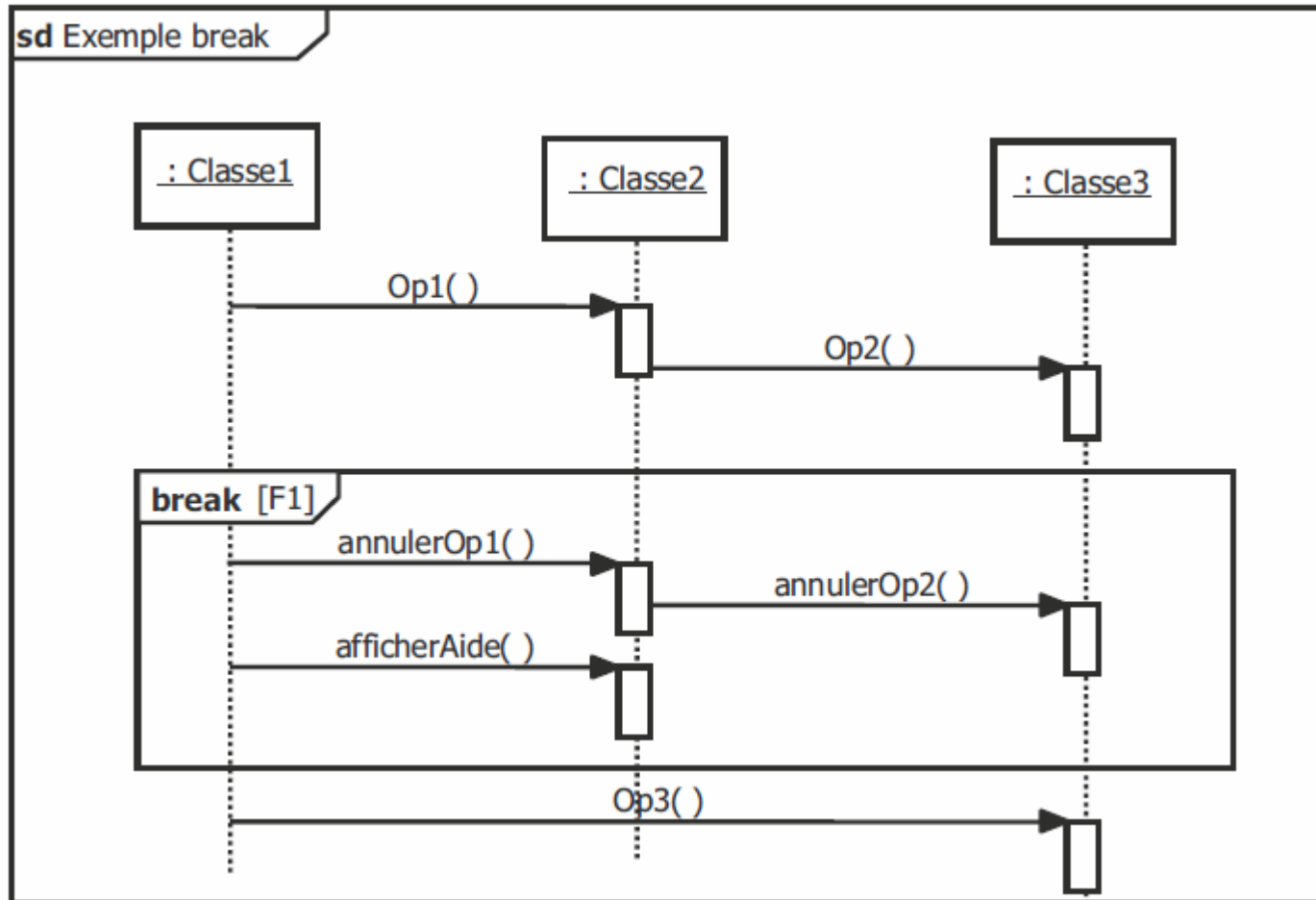
# Diagramme de séquence

## Les opérateurs

- *Opérateur break*
- Permet de représenter une **situation exceptionnelle** correspondant à un **scénario de rupture** par rapport au scénario général.
- Le scénario de rupture **s'exécute si la condition de garde est satisfaite**.

# Diagramme de séquence

## Les opérateurs



les opérations `annulerOp1( )`, `annulerOp2( )` et `afficherAide( )` ne seront exécutées que si la touche `F1` est activée sinon le fragment est ignoré et la séquence de traitement passe directement de l'opération `Op2( )` à l'opération `Op3( )`.

# Diagramme de séquence

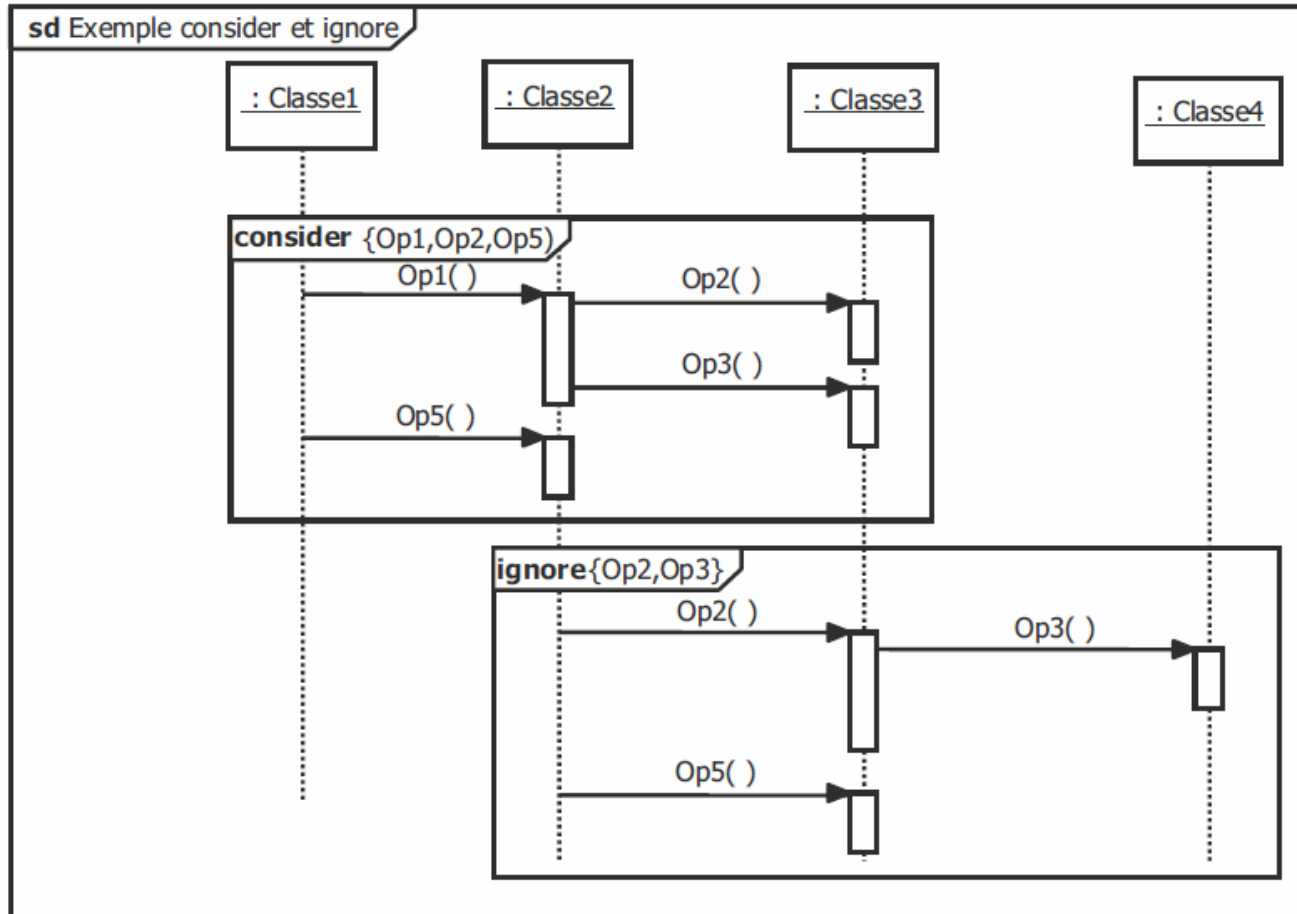
## Les opérateurs

- *Opérateurs ignore et consider*

Les opérateurs **ignore** et **consider** sont utilisés pour des **fragments d'interactions dans** lesquels on veut montrer **que certains messages peuvent être soit absents** sans avoir **d'incidence** sur le déroulement des interactions (**ignore**), **soit obligatoirement** présents (**consider**).

# Diagramme de séquence

## Les opérateurs



- dans le fragment **consider**, les messages Op1, Op2 et Op5 doivent être **obligatoirement** présents lors de l'exécution du fragment sinon le fragment n'est pas exécuté,
- dans le fragment **ignore**, les messages Op2 et Op3 **peuvent** être **absents** lors de l'exécution du fragment.

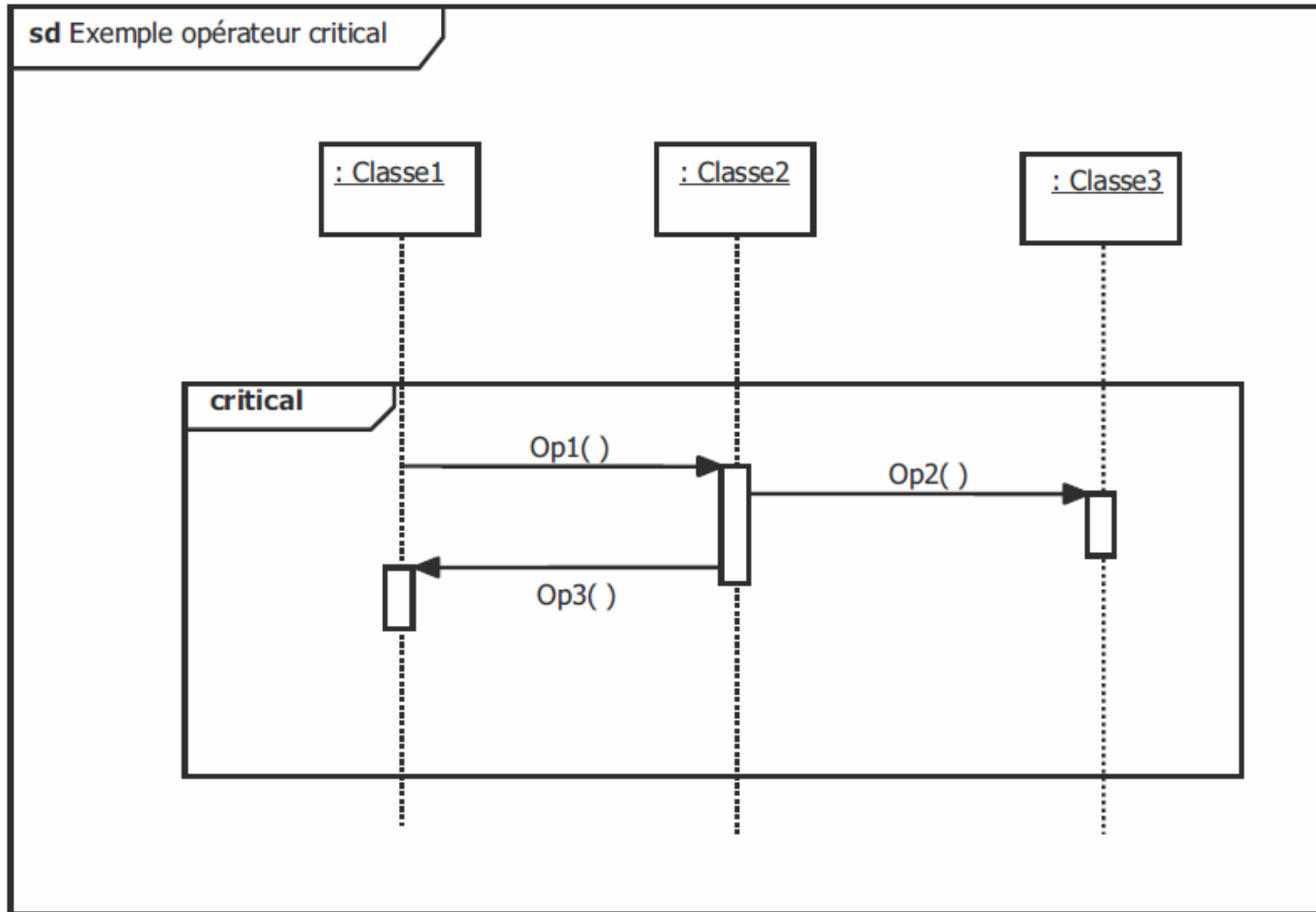
# Diagramme de séquence

## Les opérateurs

- *Opérateur critical*
- Permet d'indiquer qu'une **séquence d'interactions ne peut** être **interrompue** compte tenu du caractère critique des opérations traitées.
- On considère que le **traitement** des interactions comprises dans la séquence critique est **atomique**.

# Diagramme de séquence

## Les opérateurs

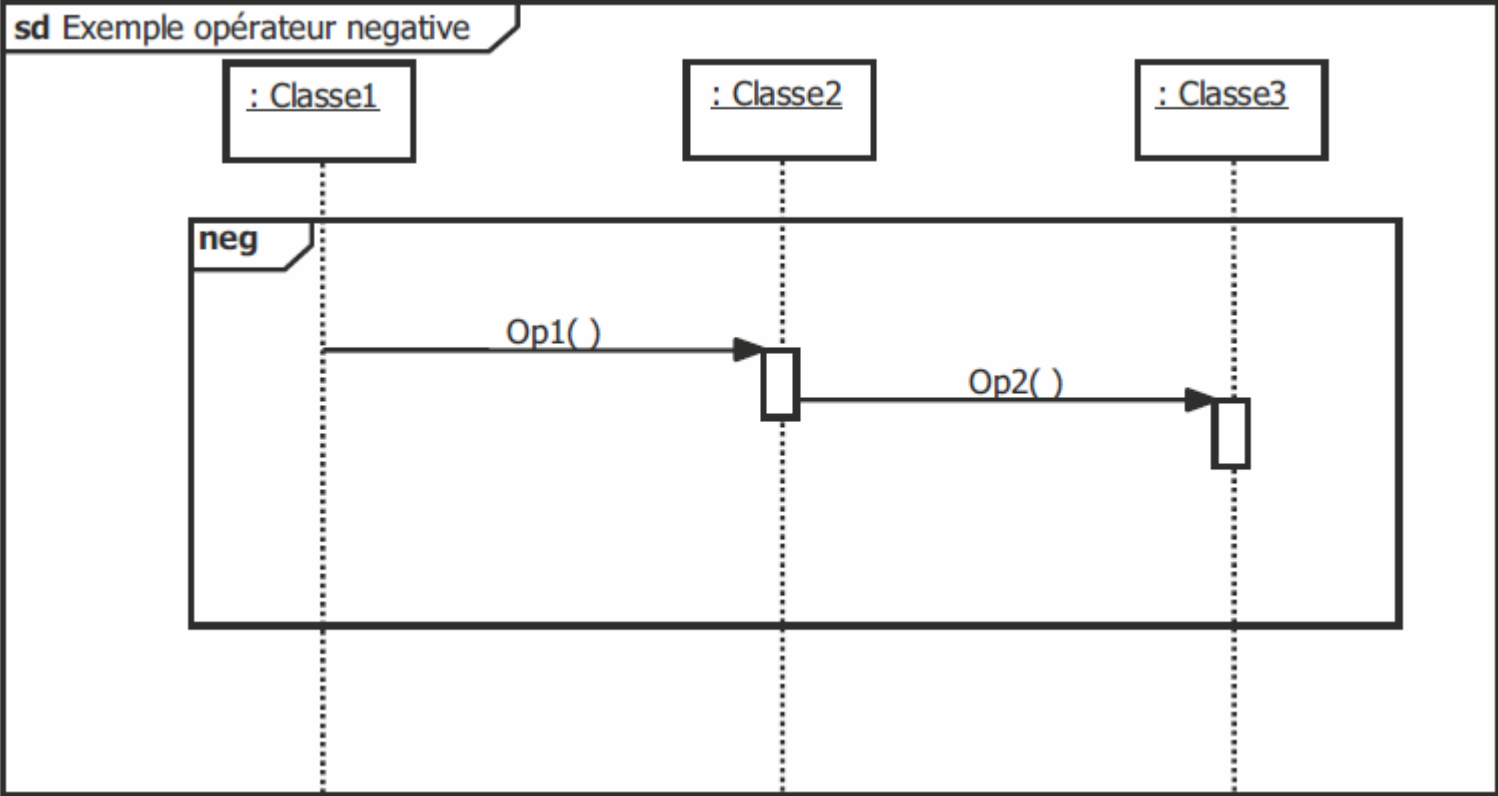


- Les opérations Op1(), Op2() et Op3() du fragment critical doivent s'exécuter sans interruption.

# Diagramme de séquence

## Les opérateurs

- *Opérateur negative neg (negative)*
- Permet d'indiquer qu'une **séquence d'interactions** est **invalide**.

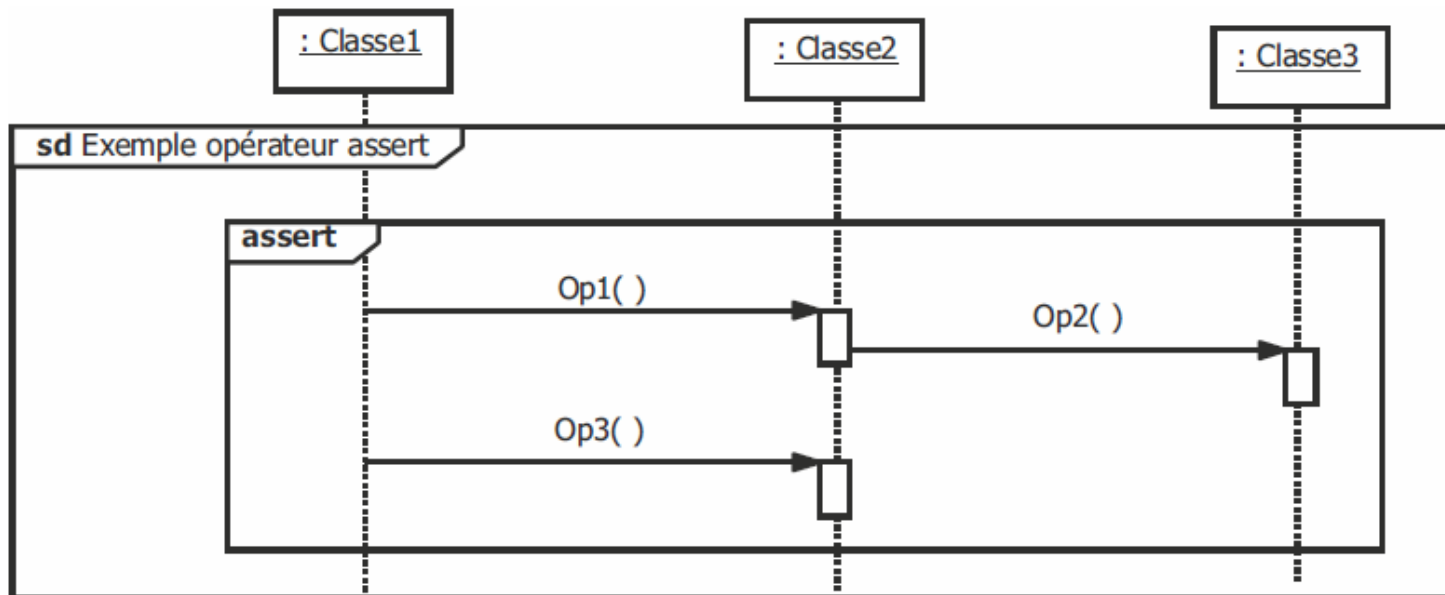


- Les opérations Op1( ) et Op2( ) du fragment **neg** sont invalides.
- Une erreur sera déclenchée dans ce cas à l'exécution du fragment.

# Diagramme de séquence

## Les opérateurs

- **Opérateur assert (assertion)**
- Permet d'indiquer qu'une séquence d'interactions est **l'unique séquence possible** en **considérant** les **messages** échangés dans le fragment.
- Toute **autre** configuration de message est **invalid**.



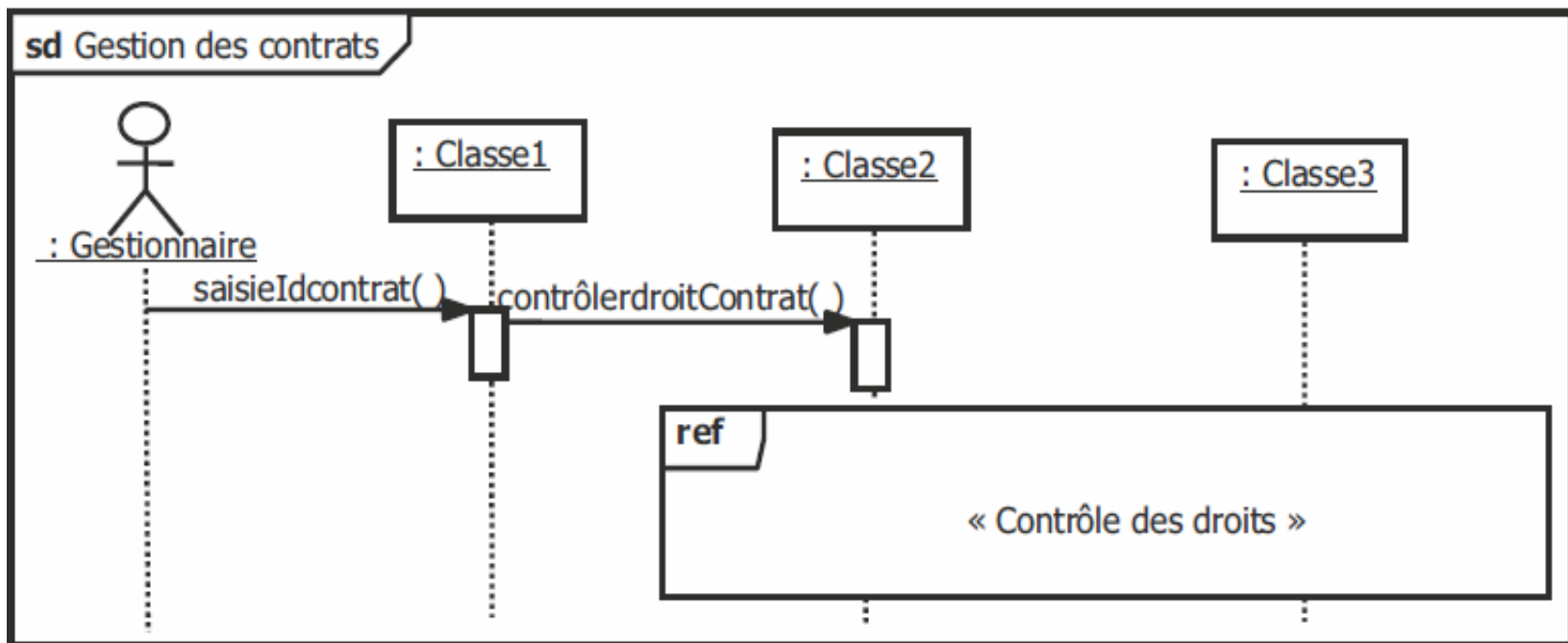
Le fragment **assert** ne s'exécutera que si l'unique séquence de traitement `Op1()`, `Op2()` et `Op3()` se réalise en respectant l'ensemble des caractéristiques de ces opérations (paramètre d'entrée, type de résultat...). Toute autre situation sera considérée invalide.



# Diagramme de séquence

## Les opérateurs

- *Opérateur ref*
- Permet **d'appeler** une séquence d'interactions décrite par ailleurs constituant ainsi une sorte de **sous-diagramme** de séquence.



On fait appel à un fragment « Contrôle des droits » qui est décrit par ailleurs.

# Diagramme de séquence

### Description des scénarios Préconditions

- La caisse du GAB est alimentée.
- Aucune carte ne se trouve déjà coincée dans le lecteur.
- La connexion avec le Système d'autorisation est opérationnelle.

### Scénario nominal

1. Le Porteur de carte introduit sa carte dans le lecteur de cartes du GAB.
2. Le GAB vérifie que la carte introduite est bien une carte bancaire.
3. Le GAB demande au Porteur de carte de saisir son code d'identification.
4. Le Porteur de carte saisit son code d'identification.
5. Le GAB compare le code d'identification avec celui qui est codé sur la puce de la carte.
6. Le GAB demande une autorisation au Système d'autorisation.
7. Le Système d'autorisation donne son accord et indique le crédit hebdomadaire.
8. Le GAB demande au Porteur de carte de saisir le montant désiré du retrait.
9. Le Porteur de carte saisit le montant désiré du retrait.
10. Le GAB contrôle le montant demandé par rapport au crédit hebdomadaire.
11. Le GAB demande au Porteur de carte s'il veut un ticket.
12. Le Porteur de carte demande un ticket.
13. Le GAB rend sa carte au Porteur de carte.
14. Le Porteur de carte reprend sa carte.
15. Le GAB délivre les billets et un ticket.
16. Le Porteur de carte prend les billets et le ticket.

Exemple (GAB)

Diagramme  
de séquence  
du scénario  
nominal de  
Retirer de  
l'argent

