

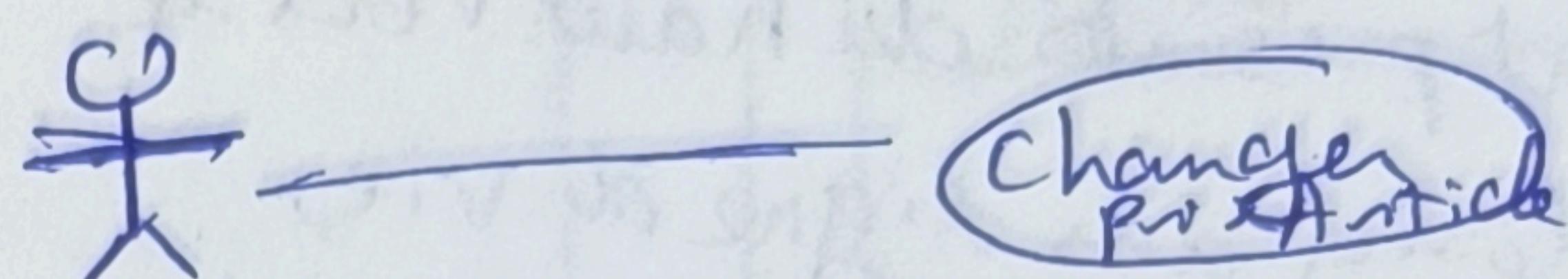
Conception des systèmes et l'information

Diagramme de séquence:

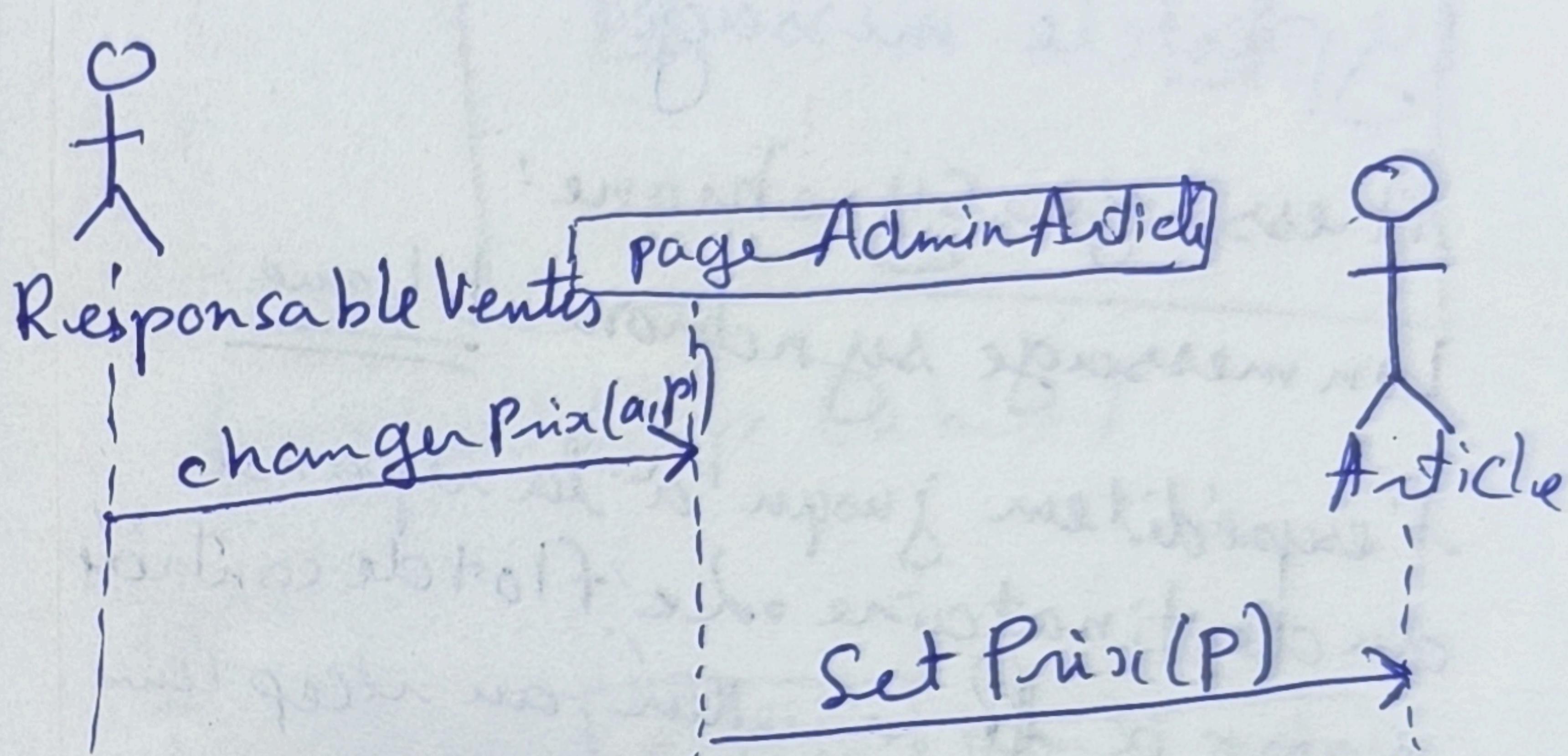
- Les diagrammes de cas d'utilisation modélisent à Quoi sera le système, en organisant les interactions possibles avec les acteurs.
- Les diagrammes de classes permettent de spécifier la structure et les liens entre les objets dont le système est composé : il spécifie Qui sera à l'œuvre dans le système pour réaliser les fonctionnalités décrites par les diagrammes de cas d'utilisation.
- Les diagrammes de séquences permettent de décrire COMMENT les éléments du système interagissent entre eux et avec les acteurs :
 - Les objets du système interagissent en s'échangeant des messages
 - Les acteurs interagissent avec le système au moyen d'IHM (Interfaces Homme-Machine)

Interaction:

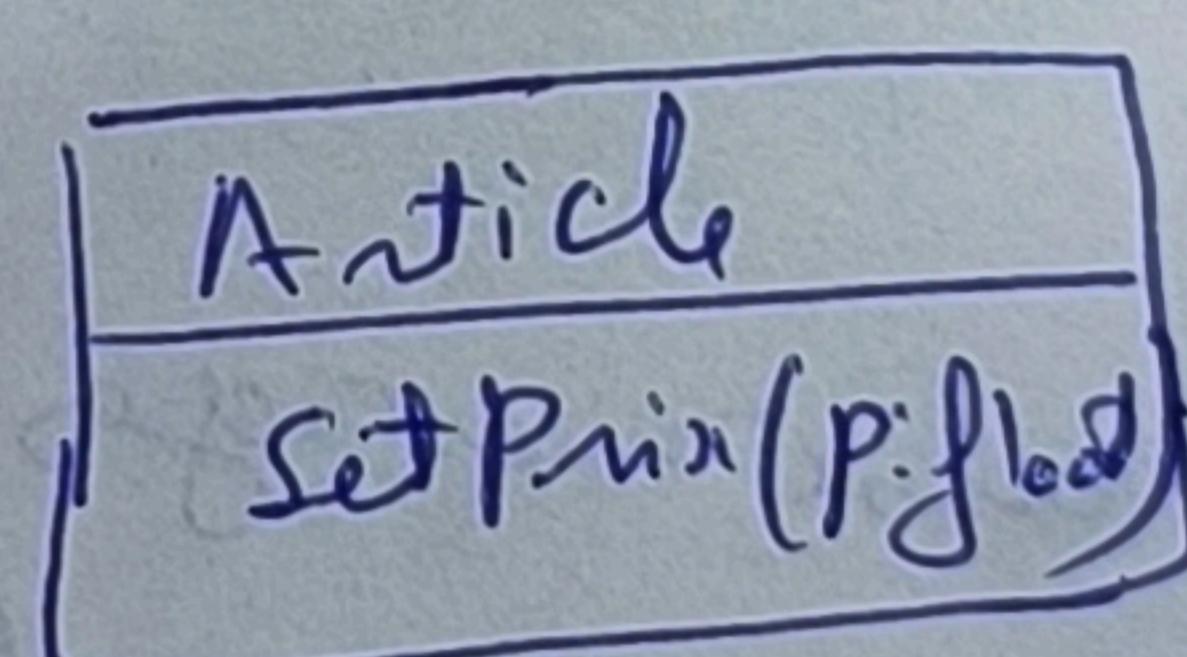
cas d'utilisation



séquence

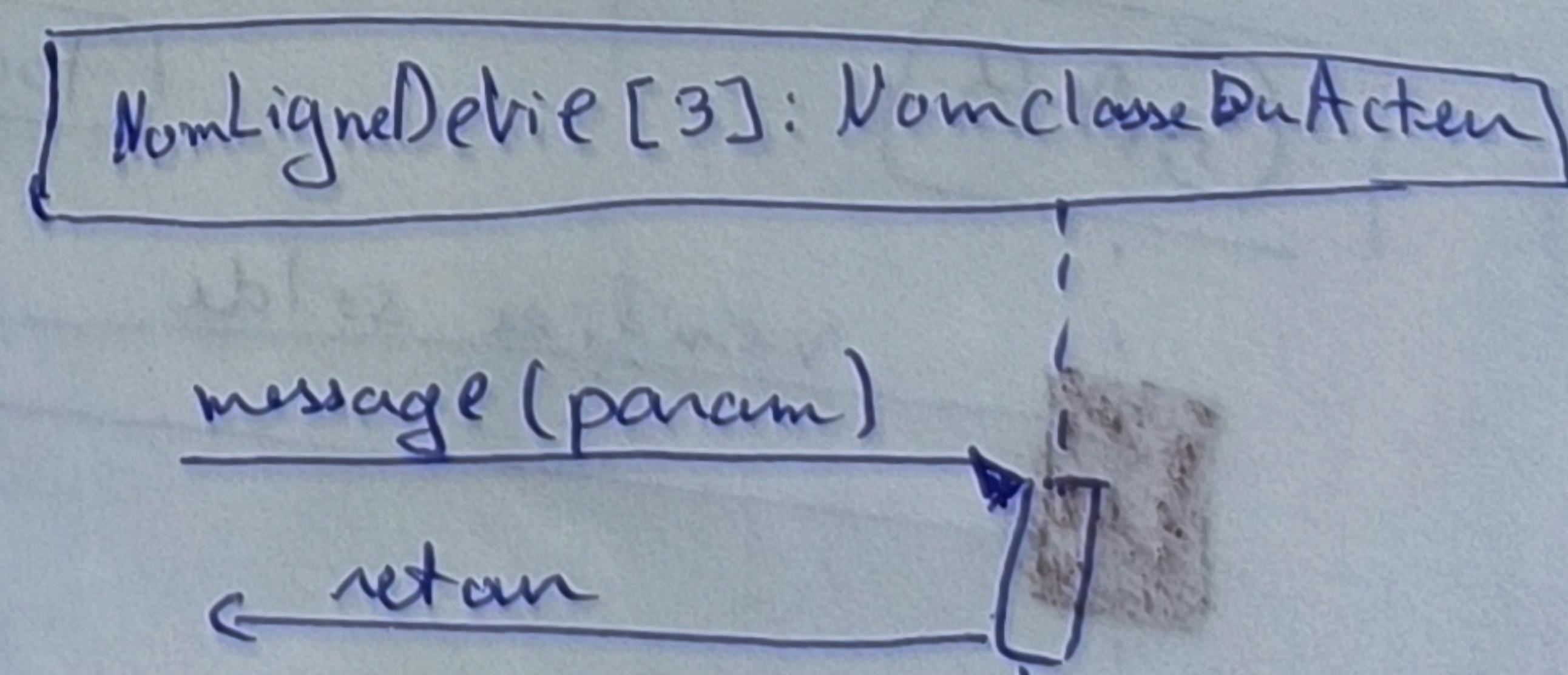


- Classes pour spécifier les opérations nécessaires



Ligne de vie

Une ligne de vie représente un participant à une interaction



dans le cas d'une collection de participants, un sélection permet de choisir un objet parmi n

Messages:

Le plus important compartiment dans les diagrammes de séquence sont les messages

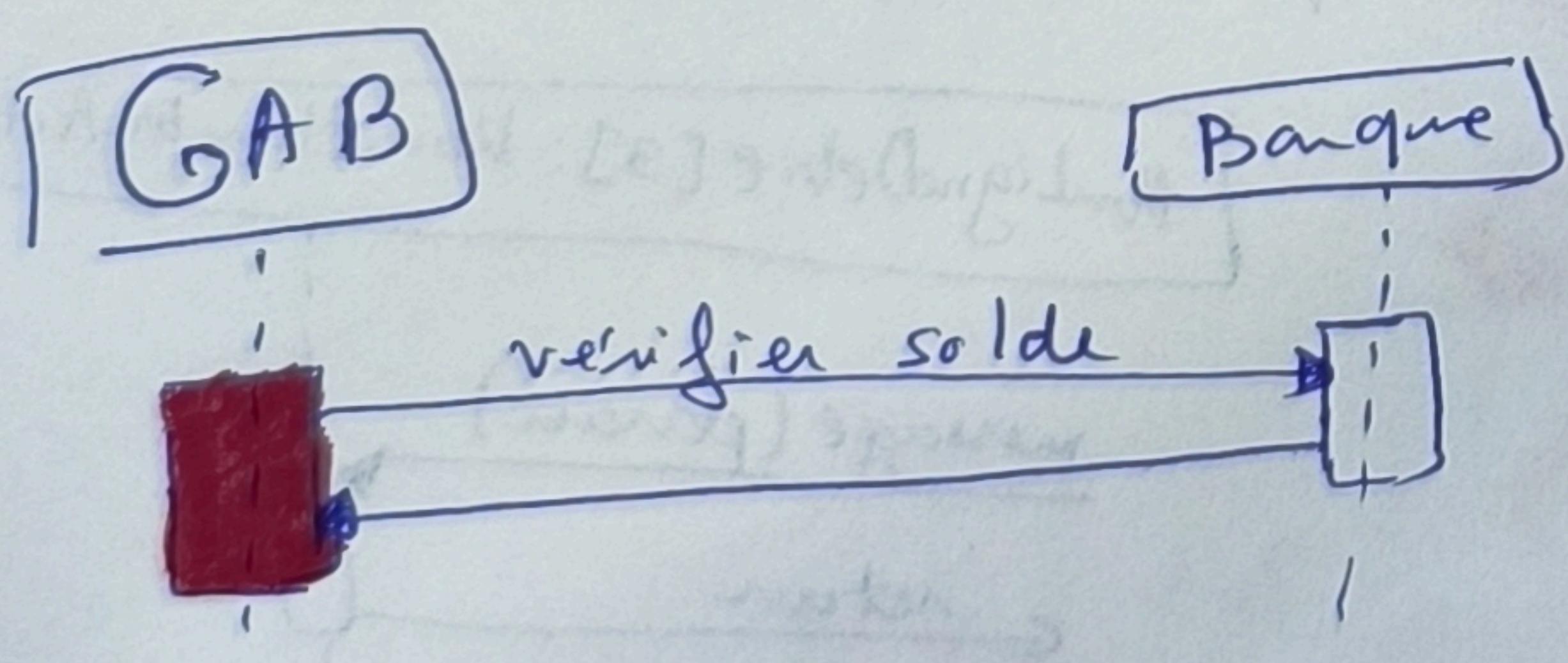
- Ils sont représentés par des flèches
- Ils sont présents du haut vers le bas de long des lignes de vie, dans un ordre chronologique

Types de messages:

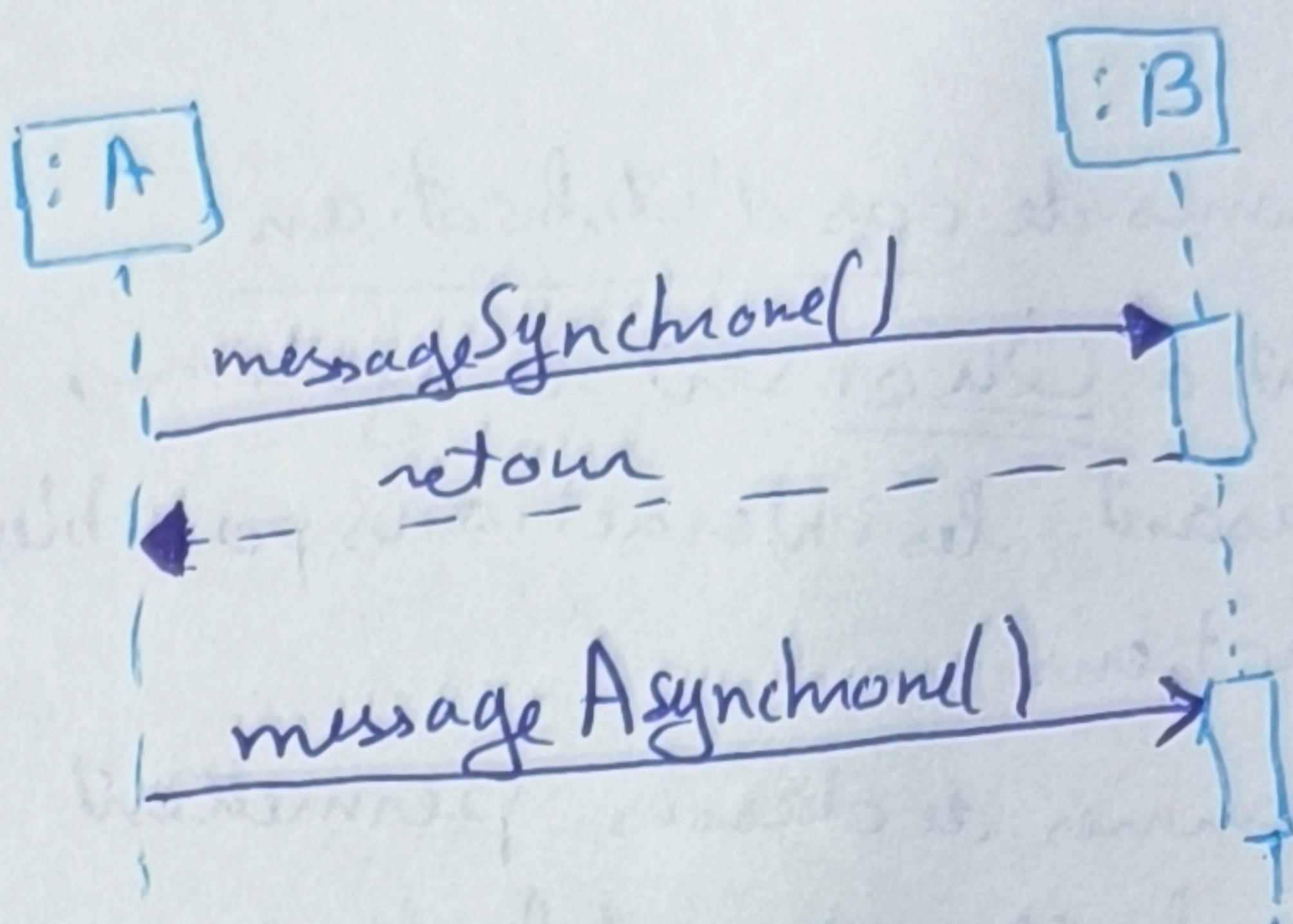
Messages synchrones:

Un message synchrone bloque l'expéditeur jusqu'à la réponse du destinataire où le fil de contrôle passe à l'émetteur au récepteur.

- Si un objet A envoie un message synchrone à un objet B, A reste bloqué tant que B n'a pas terminé
- On peut associer aux messages d'appel de méthode un message de retour (en pointillés) manquant la reprise du contrôle par l'objet émetteur du message synchrone



Un message asynchrone n'est pas bloquant pour l'expéditeur. Le message envoyé peut être pris en compte par le récepteur à tout moment ou ignoré

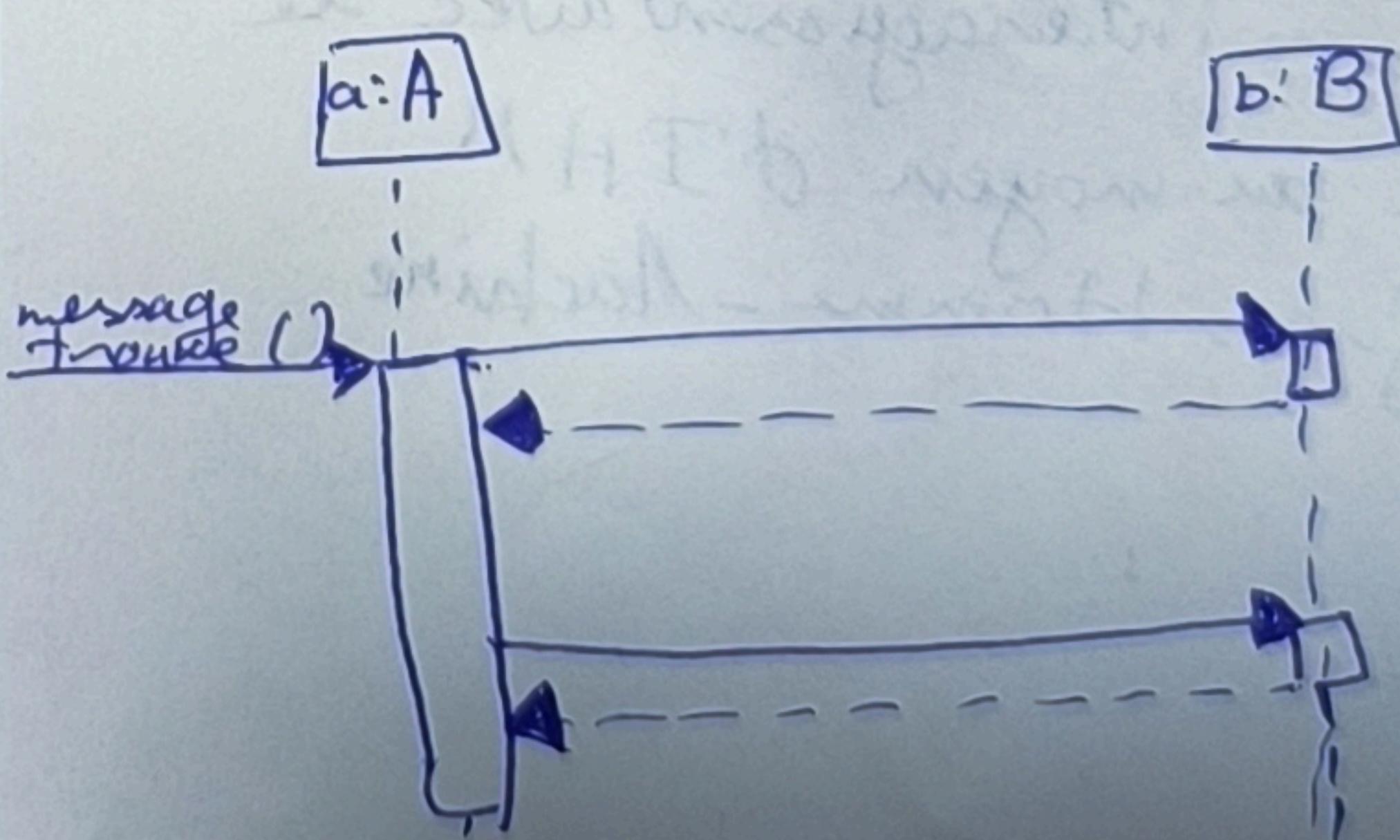


Message de retour:

Le récepteur d'un message synchrone répond à l'émetteur en lui envoyant un message de retour

Messages trouvés:

Les diagrammes de séquences peuvent être employés pour décrire les traitements d'un message résultant de l'envoi d'un message, indépendamment de l'émetteur. Dans ce cas, l'émetteur importe peu et on le spécifie par



Ce type de message peut être synchronisé ou asynchrone

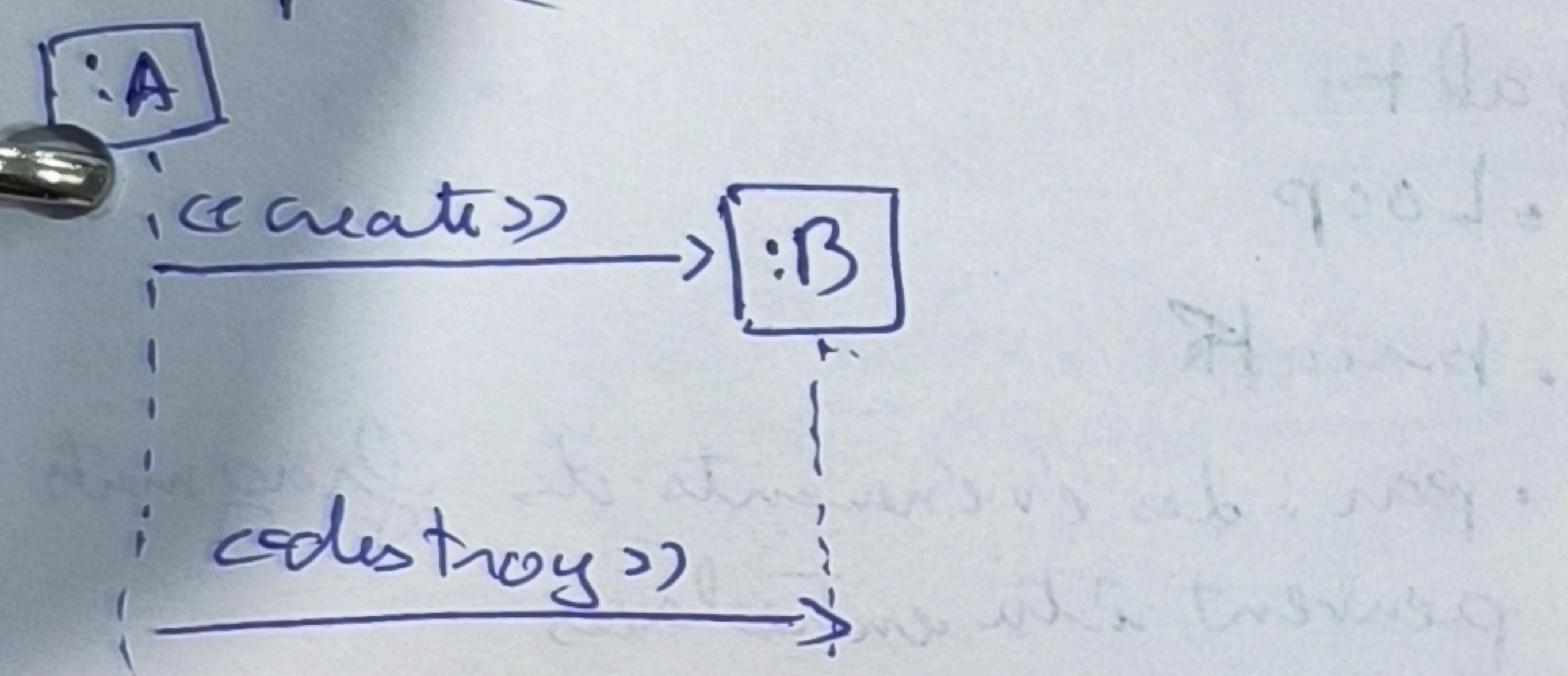
ay messages Perdus

les messages perdus, on connaît l'émetteur mais pas le récepteur à l'inverse des messages trouvés

On utilise souvent des messages de retour perdu afin de spécifier le résultat d'un message synchrone trouvé.

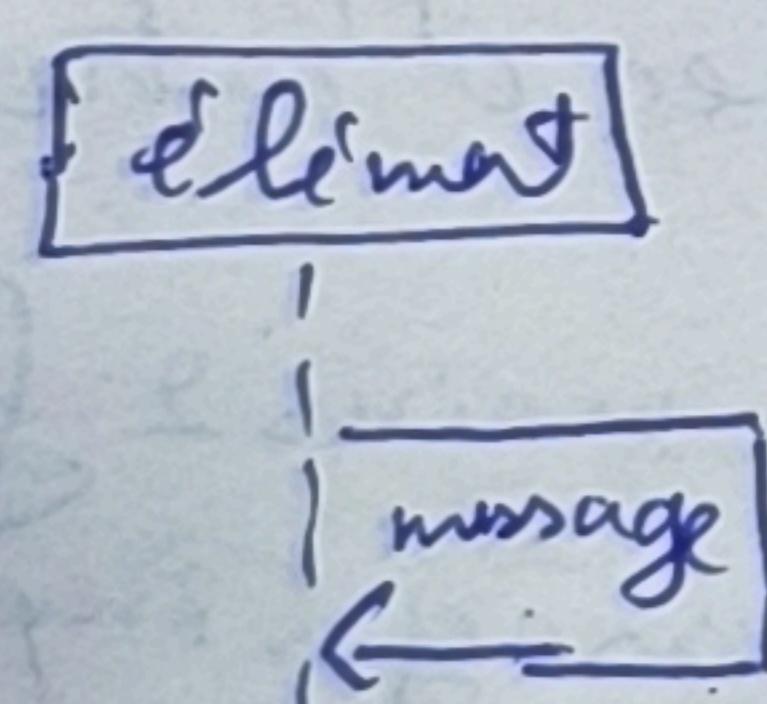
Création et destruction d'objets (et de lignes clé vierges)

Principale



~~Fragment combiné~~

Message réflexif:

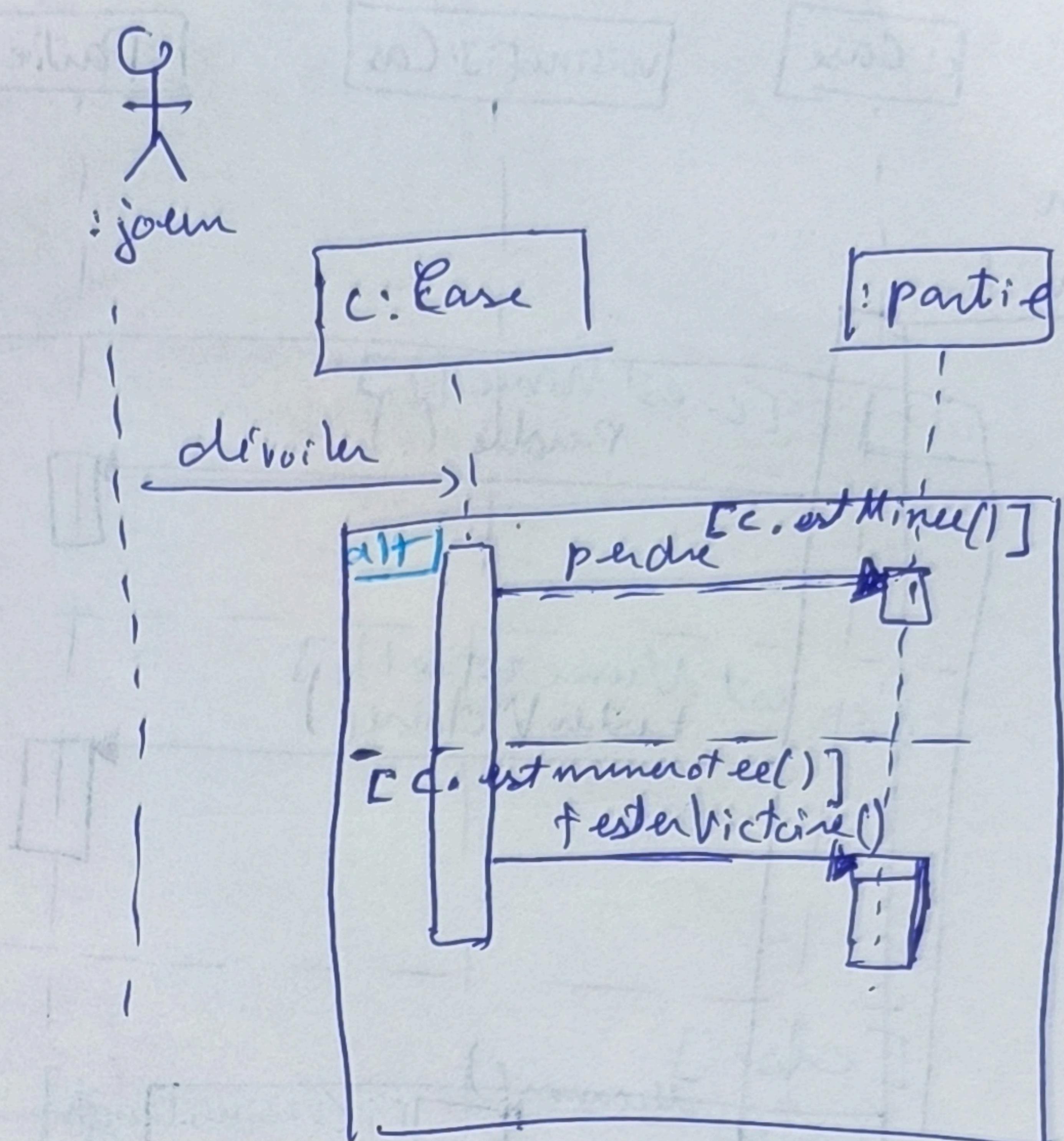


Un objet s'envoie à lui-même

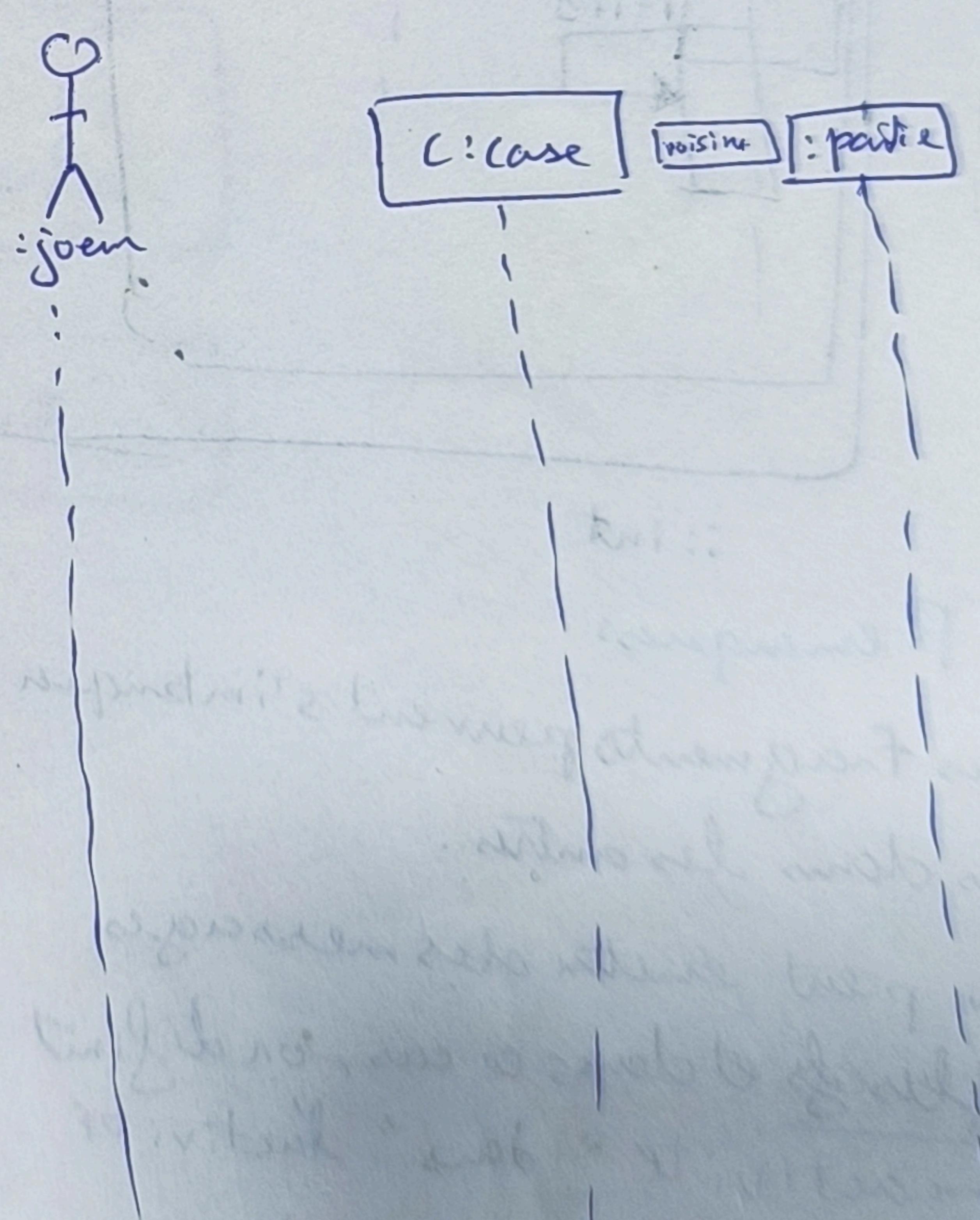
Fragments combinés

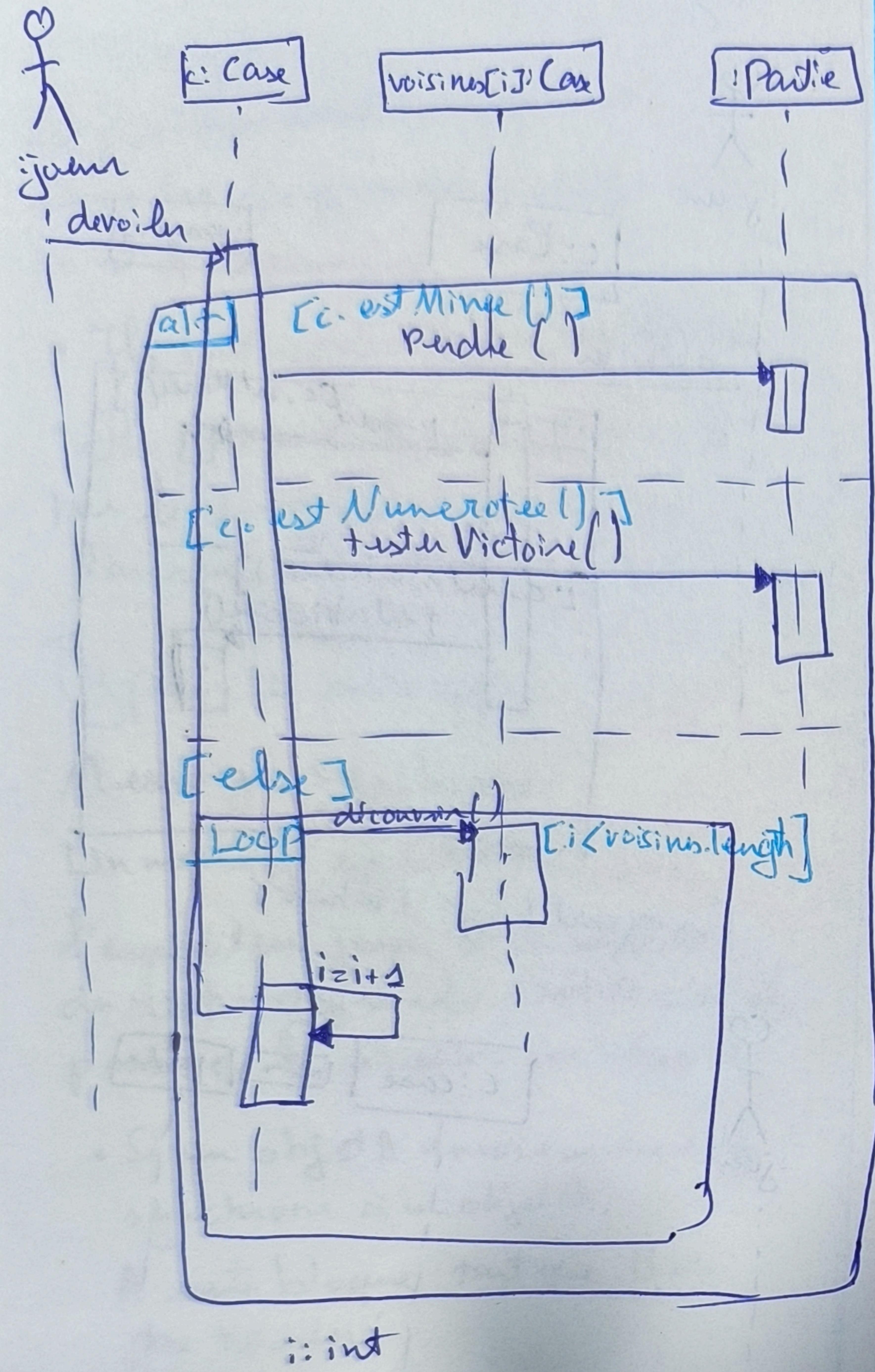
Un fragment combiné se représente de la même façon qu'une interaction

Fragment alt opérations conditionnel



Fragment Loop (while)





:: int

Remarques:

- des fragments peuvent s'imbriquer les uns dans les autres.
- on peut émettre des messages réfléchissants et dans ce cas, on définit une activité "dans" l'activité
- lorsqu'on décrit une opération dans le détail, il est permis de placer des commandes sur les flèches au lieu de messages correspondant à des opérations ou des signaux

- Tous les éléments d'un diagramme doivent être définis. Typiquement, les attributs doivent correspondre soit à des attributs définis dans un diagramme de classes au niveau de la ligne de vie et contrôlés par une séquence
 - soit à des attributs définis localement au diagramme de séquence (ici, i)
- Opérations de flux de contrôle**
- opt (If) facultatif
 - alt
 - Loop
 - break
 - Par : des événements de fragments peuvent être entrelacés
 - Critical : Utilisé dans un fragment par ou seq. Indique que les messages de fragment ne doivent pas être entrelacés avec d'autres messages
 - Seq : Il existe au moins 2 fragments d'opérande. Les messages impliquant la même ligne de vie doivent se produire dans l'ordre des fragments. Lorsqu'ils n'impliquent pas la même ligne de vie, les messages des différents fragments peuvent être entrelacés en parallèle.
 - Strict : Il existe au moins deux fragments d'opérande. Les fragments doivent se produire dans l'ordre donné. ref permet d'indiquer la réutilisation

cas d'utilisation

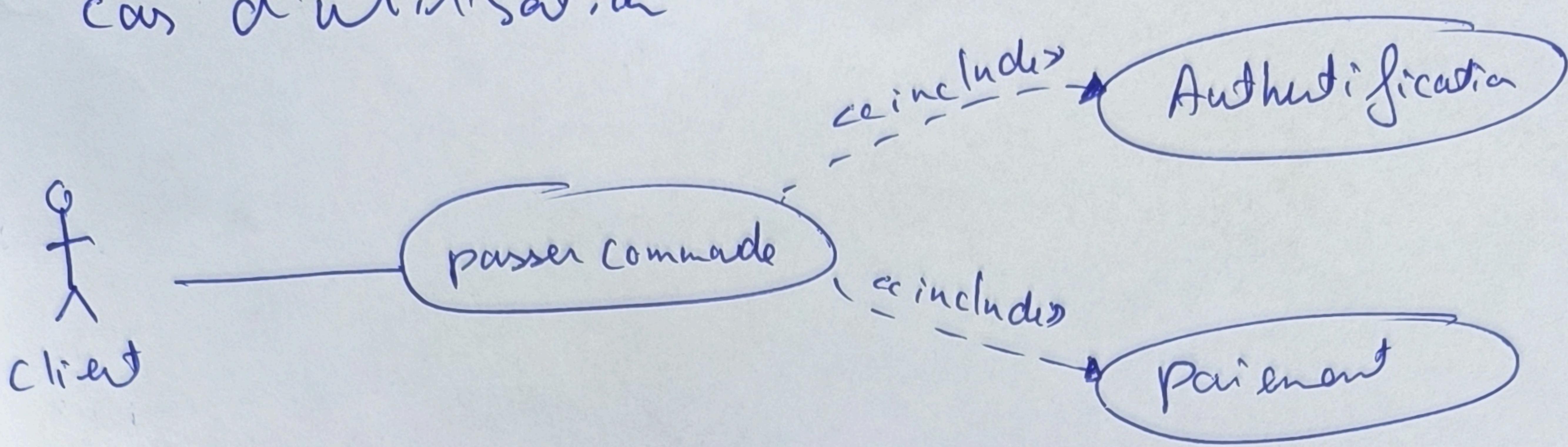
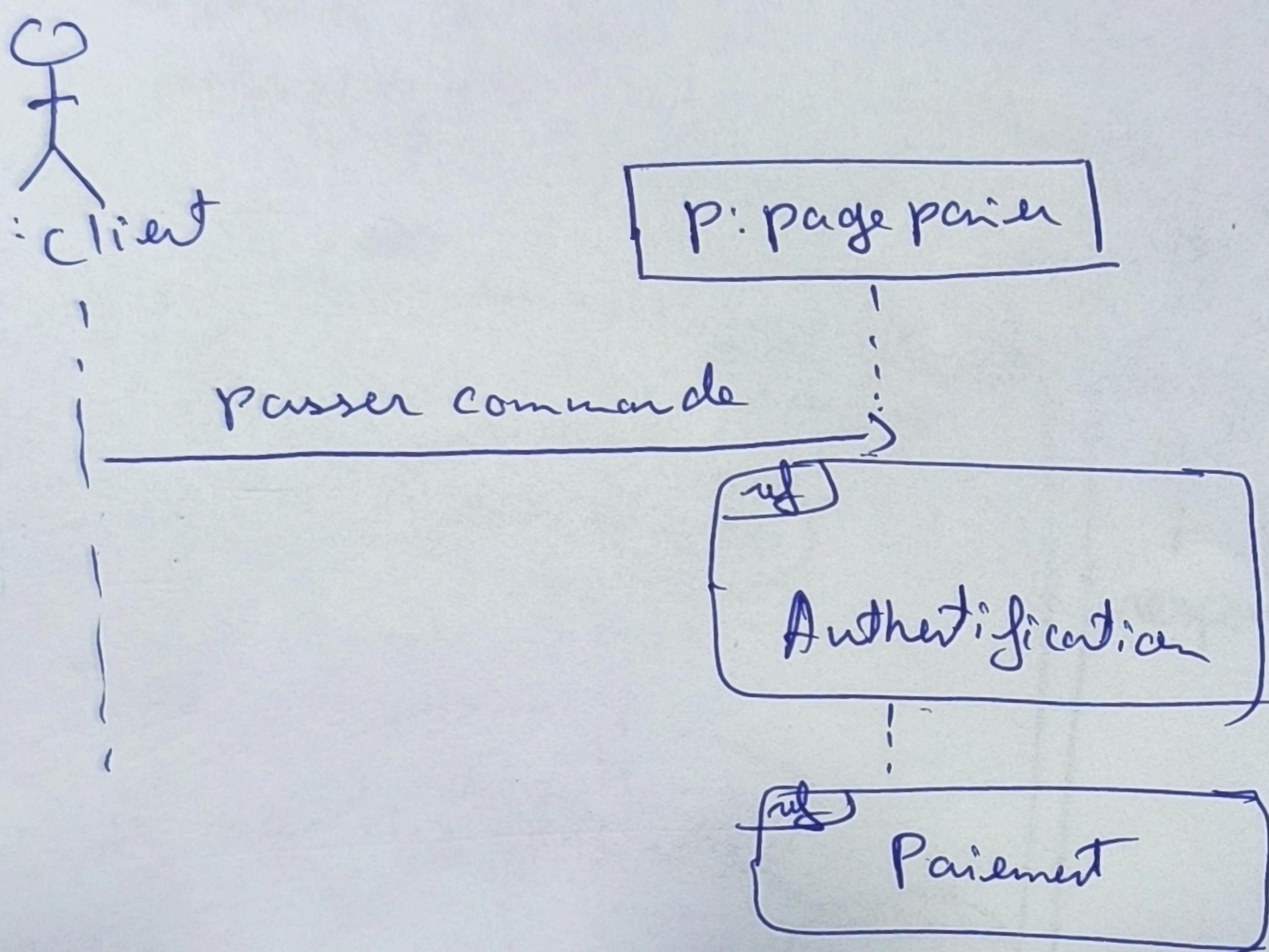


diagramme de séquence



fragment assert