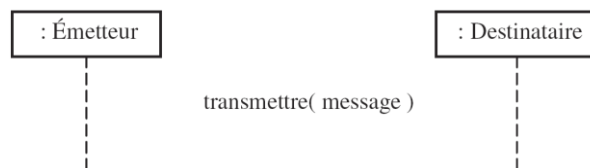


## TP02 : Diagrammes de séquences

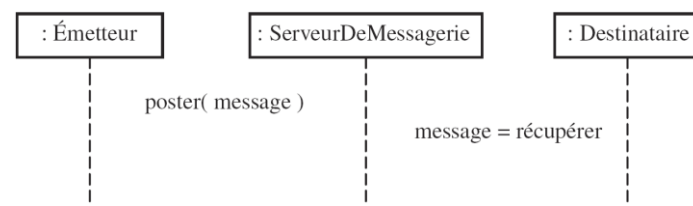
### Exercice 1 : Types de messages

**Question 1 :** Quand un courrier électronique est envoyé par l'émetteur, celui-ci ne veut pas attendre que le destinataire l'ait reçu et il n'y a pas d'intermédiaire. Peut-on utiliser un message synchrone ? Complétez la figure ci-dessous par des flèches représentant des messages



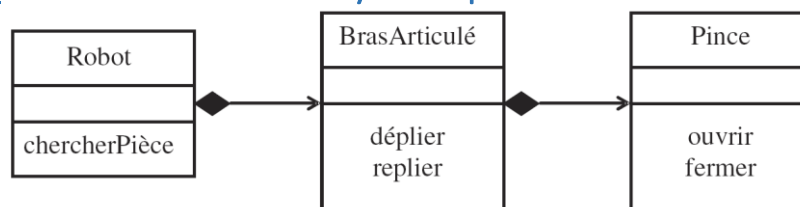
**Question 2 :** Est-ce que transmettre est une opération ou un signal ? Dans tous les cas, donnez des Éléments d'un diagramme de classe cohérent avec le diagramme de séquence.

**Question 3 :** Un serveur de messagerie sert d'intermédiaire entre l'émetteur et le récepteur d'un email. Le serveur est toujours en fonction. Est-ce qu'on peut utiliser des messages synchrones pour l'envoi et la récupération de emails ? Complétez la figure ci-dessous par des flèches représentant des messages.



**Question 4 :** Est-ce que poster est une opération ou un signal ? Dans tous les cas, proposez un diagramme de classe cohérent avec le diagramme de séquence.

### Exercice 2 : Modélisation de la dynamique de l'interaction entre classes



Le diagramme de classes ci-dessus décrit des robots composés de bras articulés, eux-mêmes composés de pinces. L'implémentation des classes en question est la suivante :

```

class Robot{
    privée :
  
```

```

        BrasArticulé brasArticulé ;
    publique :
        void chercherPièce() {
            brasArticulé.déplier() ;
            brasArticulé.replier() ;
        }
}

class BrasArticulé {
    privée :
        Pince pince ;
    publique :
        void déplier() {
            ...
            pince.fermer() ;
        }
        void replier() {
            ...
            pince.ouvrir() ;
        }
}

class Pince {
    privée :
        ...
    publique :
        void fermer() { ... }
        void ouvrir() { ... }
}

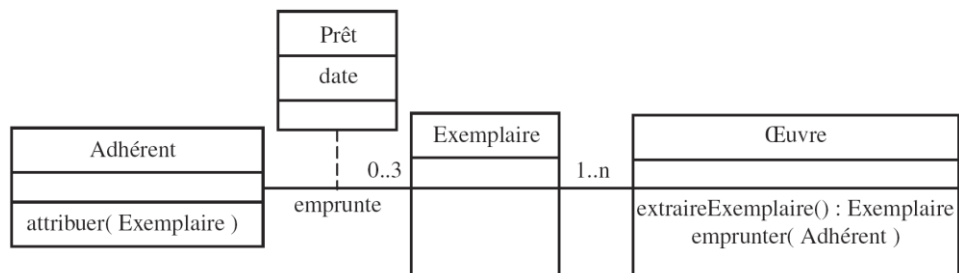
Début programme principal
Robot robot ;
robot.chercherPièce() ;
Fin programme principal

```

**Question :** Donnez un diagramme de séquence représentant l'échange de messages résultant de l'exécution du programme principal.

### Exercice 3 : Modélisation d'une interaction interne

Le diagramme de classes présenté ci-dessous modélise la structure interne de la bibliothèque.



Un acteur adhérent peut emprunter un exempleire d'une œuvre donnée. L'emprunt se fait de la façon suivante : la méthode emprunter est appelée avec un objet de classe Adhérent donné en argument ;

s'il reste des exemplaires dans la bibliothèque, l'un des exemplaires associés à l'œuvre est extrait via la méthode `extraireExemplaire`, une instance de la classe `Prêt` est créée, puis l'exemplaire extrait de la bibliothèque est attribué à l'adhérent grâce à l'opération `attribuer`. S'il restait un exemplaire, l'œuvre retourne « OK » et dans le cas contraire, elle retourne « PasOK ».

## Exercice 4 : Documentation d'un cas d'utilisation

La rubrique « enchaînement nominal » du cas d'utilisation retrait d'espèces contient les éléments suivants :

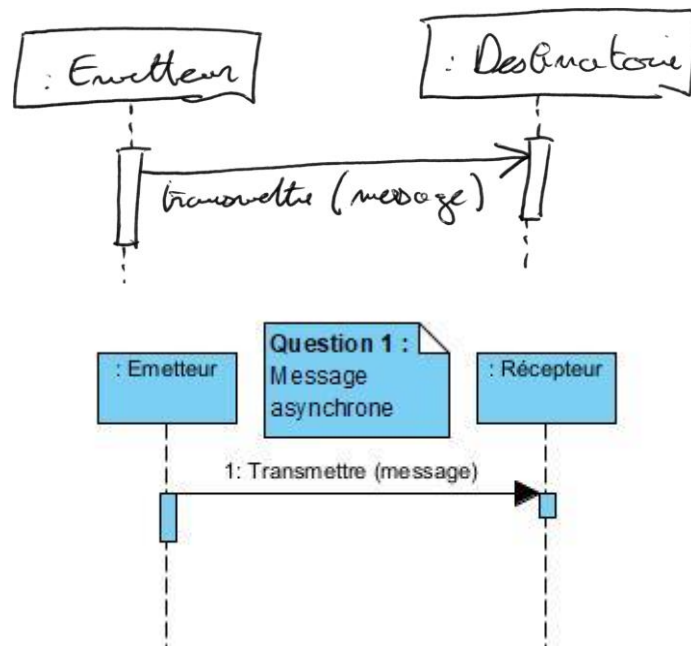
1. Le guichetier saisit le numéro de compte du client ;
2. L'application valide le compte auprès du système central ;
3. Le guichetier demande un retrait de 100 euros ;
4. Le système « guichet » interroge le système central pour s'assurer que le compte est suffisamment approvisionné ;
5. Le système central effectue le débit du compte ;
6. En retour, le système notifie au guichetier qu'il peut délivrer le montant demandé.

**Question :** Donner le diagramme de séquences associé à cette description textuelle.

## Correction exercice 1 : Types de messages

**Question 1 :** Quand un courrier électronique est envoyé par l'émetteur, celui-ci ne veut pas attendre que le destinataire l'ait reçu et il n'y a pas d'intermédiaire. Peut-on utiliser un message synchrone ? Complétez la figure ci-dessous par des flèches représentant des messages

*On doit utiliser un message asynchrone, puisque l'envoi de message n'est pas bloquant pour l'émetteur.*



**Question 2 :** Est-ce que transmettre est une opération ou un signal ? Dans tous les cas, donnez des Éléments d'un diagramme de classe cohérent avec le diagramme de séquence.

### Différence entre opération et signal

Source : <http://www.pharmasen-emplois.com/tdr/sarr21939.pdf>

Les principales informations contenues dans un diagramme de séquence sont les messages échangés entre les lignes de vie, présentés dans un ordre chronologique.

Un message définit une communication particulière entre des lignes de vie (objets ou acteurs).

Plusieurs types de messages existent, dont les plus courants :

- l'envoi d'un signal ;
- l'invocation d'une opération (appel de méthode) ;
- la création ou la destruction d'un objet.

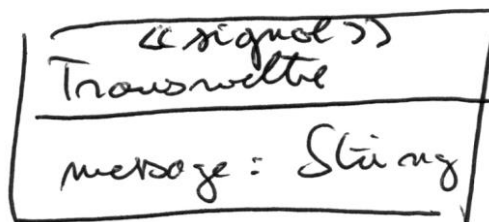
Un **message synchrone** bloque l'expéditeur jusqu'à la réponse du destinataire. Le flot de contrôle passe de l'émetteur au récepteur. Typiquement, il s'agit d'un appel de **méthode** : Si un objet A invoque une méthode d'un objet B, A reste bloqué tant que B n'a pas terminé. On peut associer aux messages d'appel de méthode un message de retour (en pointillés) marquant la reprise du contrôle par l'objet émetteur du message synchrone. Les messages synchrones correspondent à des **opérations** dans le diagramme de classes.

L'invocation d'une opération est le type de message le plus utilisé en programmation objet. L'invocation peut être asynchrone ou synchrone. Dans la pratique, la plupart des invocations

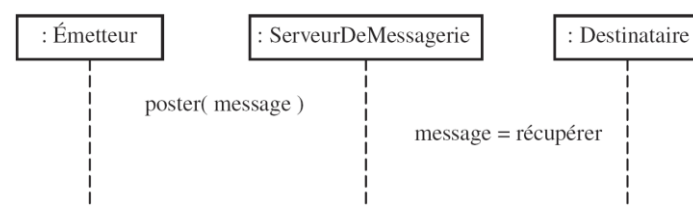
sont synchrones, l'émetteur reste alors bloqué le temps que dure l'invocation de l'opération.  
(<https://laurent-audibert.developpez.com/Cours-UML/?page=diagrammes-interaction>)

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. Typiquement : envoi de **signal** (voir stéréotype de classe « signal »). Les messages asynchrones correspondent à des **signaux** dans le diagramme de classes. Les signaux sont des objets dont la classe est stéréotypée « signal » et dont les attributs (porteurs d'information) correspondent aux paramètres du message.

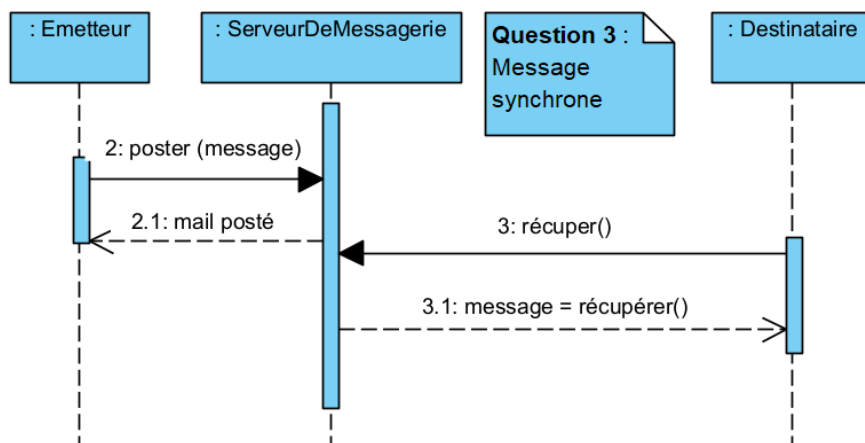
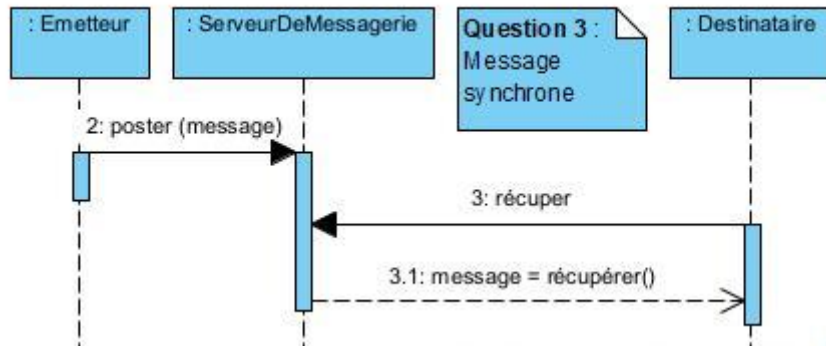
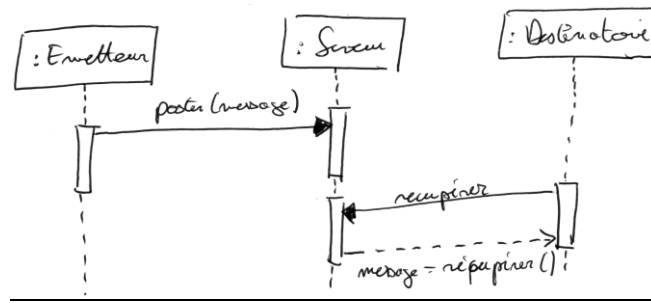
**Il s'agit donc ici d'un signal représenté comme ceci dans un diagramme de classe (TD03-Exo01).**



**Question 3 :** Un serveur de messagerie sert d'intermédiaire entre l'émetteur et le récepteur d'un email. Le serveur est toujours en fonction. Est-ce qu'on peut utiliser des messages synchrones pour l'envoi et la récupération de emails ? Complétez la figure ci-dessous par des flèches représentant des messages.

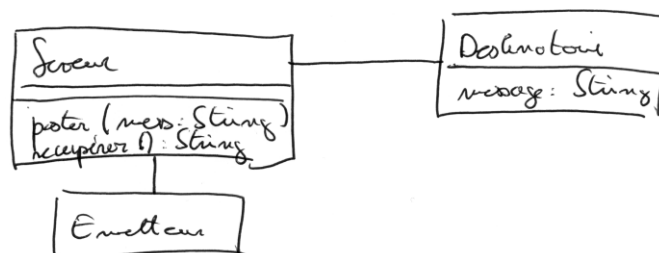


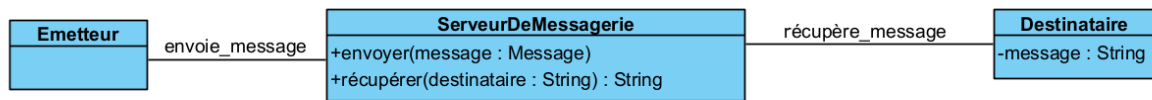
**Un message synchrone est possible ici et c'est donc préférable : si on a le choix, il vaut mieux utiliser des messages synchrones, qui s'implémentent facilement par des opérations.**



**Question 4 :** Est-ce que poster est une opération ou un signal ? Dans tous les cas, proposez un diagramme de classe cohérent avec le diagramme de séquence.

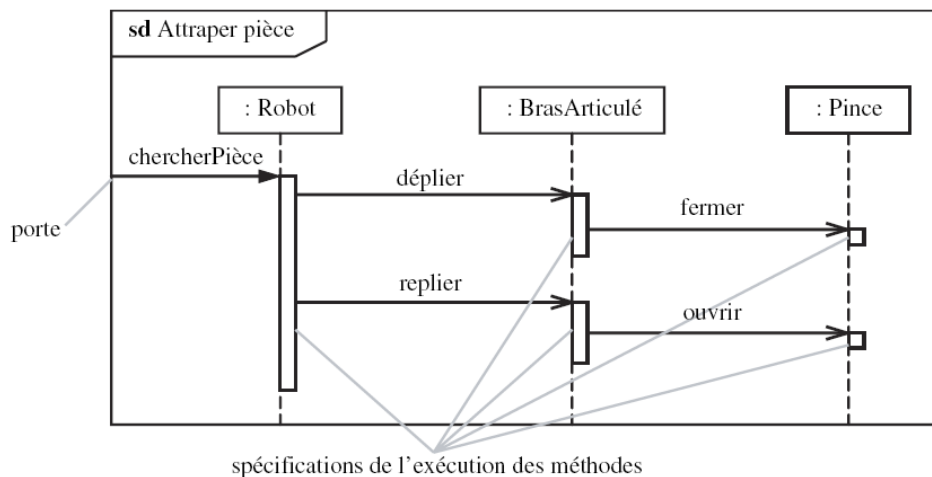
**Poster est une opération.**





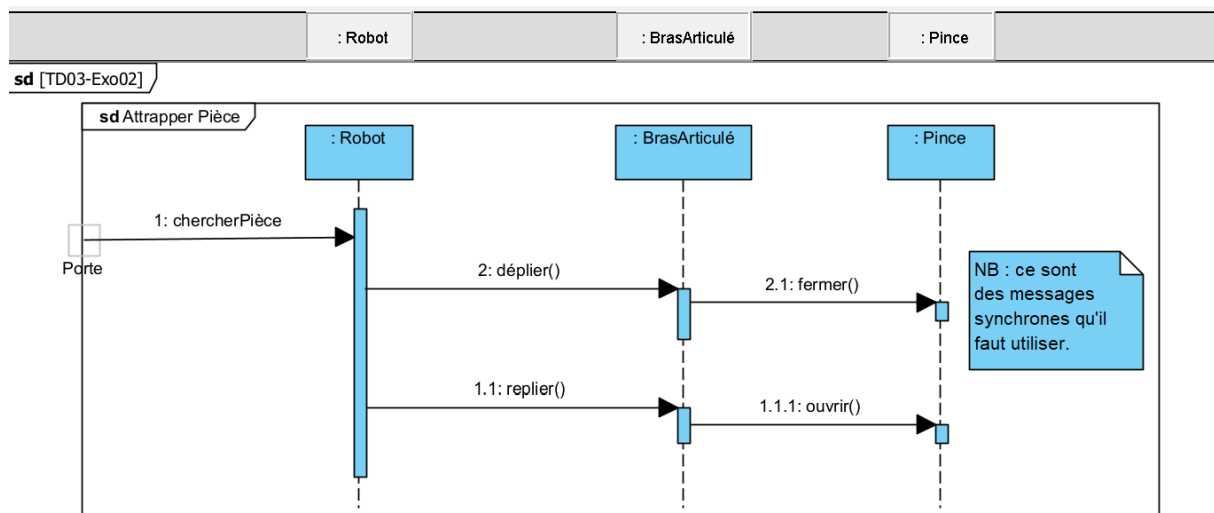
## Correction exercice 2 : Modélisation de la dynamique de l'interaction entre classes

**Question :** Donnez un diagramme de séquence représentant l'échange de messages résultant de l'exécution du programme principal.

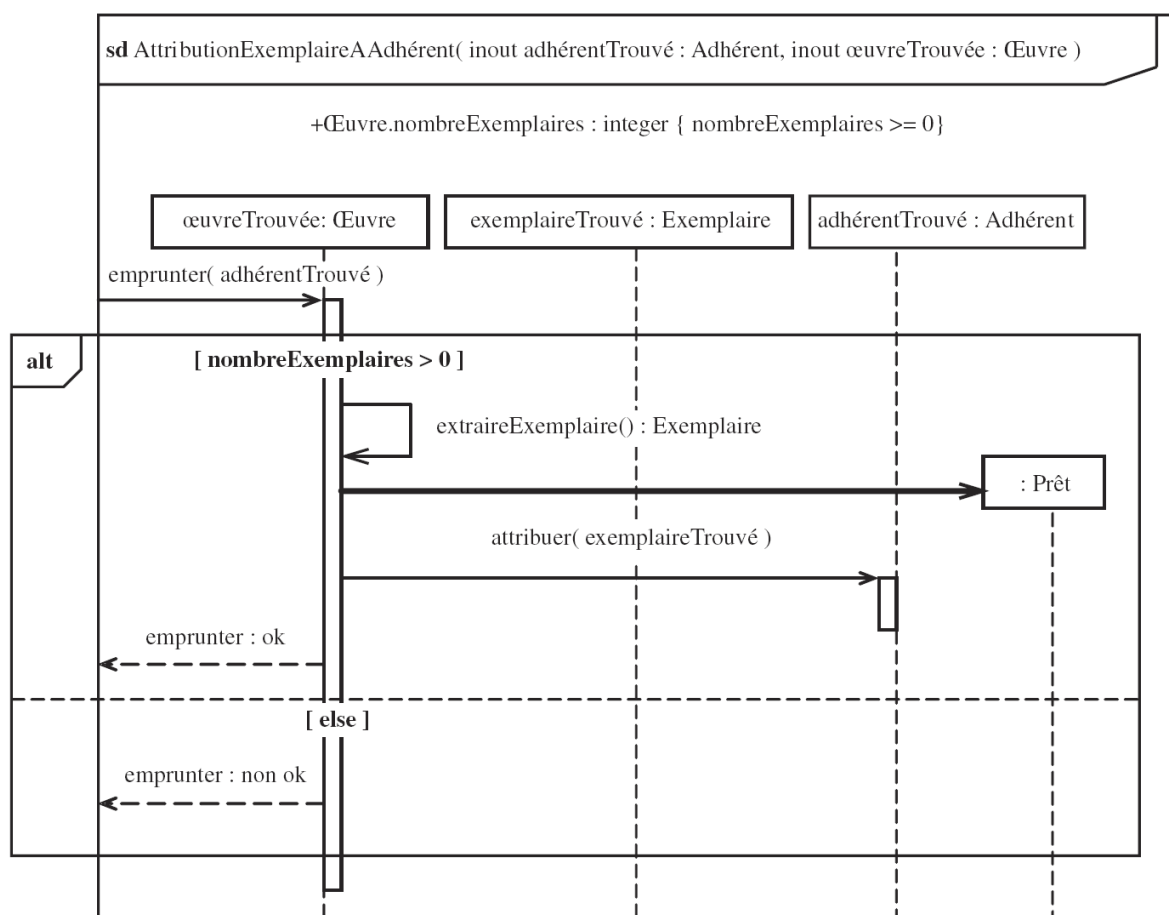


**NB :**

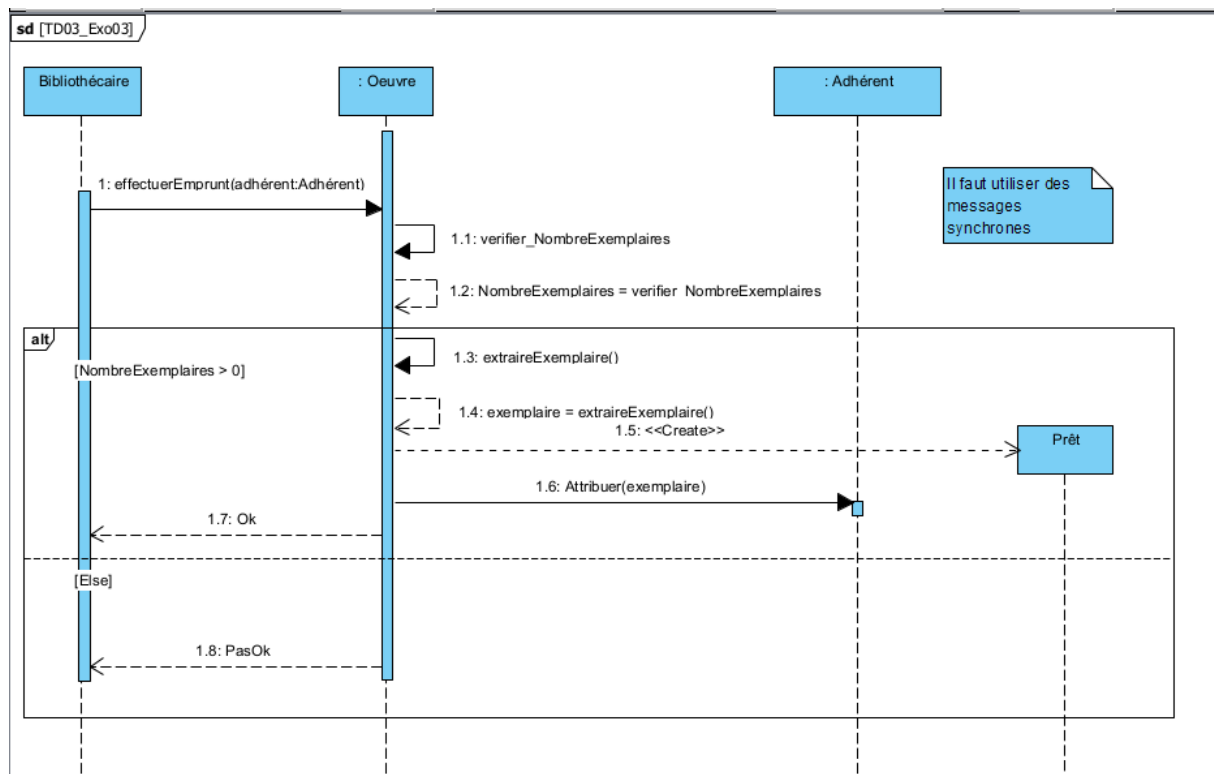
- **Ce sont des messages synchrones qu'il faut utiliser.**
- **A la place de la porte, on peut aussi avoir un objet « utilisateur » qui envoie le message au robot.**



### Correction exercice 3 : Modélisation d'une interaction interne







## Correction exercice 4 : Documentation d'un cas d'utilisation

La rubrique « enchaînement nominal » du cas d'utilisation retrait d'espèces contient les éléments suivants :

1. Le guichetier saisit le numéro de compte du client ;
2. L'application valide le compte auprès du système central ;
3. Le guichetier demande un retrait de 100 euros ;
4. Le système « guichet » interroge le système central pour s'assurer que le compte est suffisamment approvisionné ;
5. Le système central effectue le débit du compte ;
6. En retour, le système notifie au guichetier qu'il peut délivrer le montant demandé.

**Question :** Donner le diagramme de séquences associé à cette description textuelle.

