

LIFAP7

TD6. Diagrammes d'interaction. (correction)

1 Rappels

Les diagrammes d'interaction en UML donnent un point de vue temporel sur les interactions entre instances. Ils apportent un aspect dynamique à la modélisation du système en montrant comment les objets (instances de classes) communiquent pour réaliser une certaine fonctionnalité. UML propose principalement deux diagrammes équivalents pour illustrer une interaction : le diagramme de séquence et le diagramme de communication.

Le **diagramme de séquence** se focalise sur le déroulement dans le temps (chronologique) d'un scénario. L'aspect temporel est représenté explicitement par l'axe vertical et s'écoule de haut en bas.

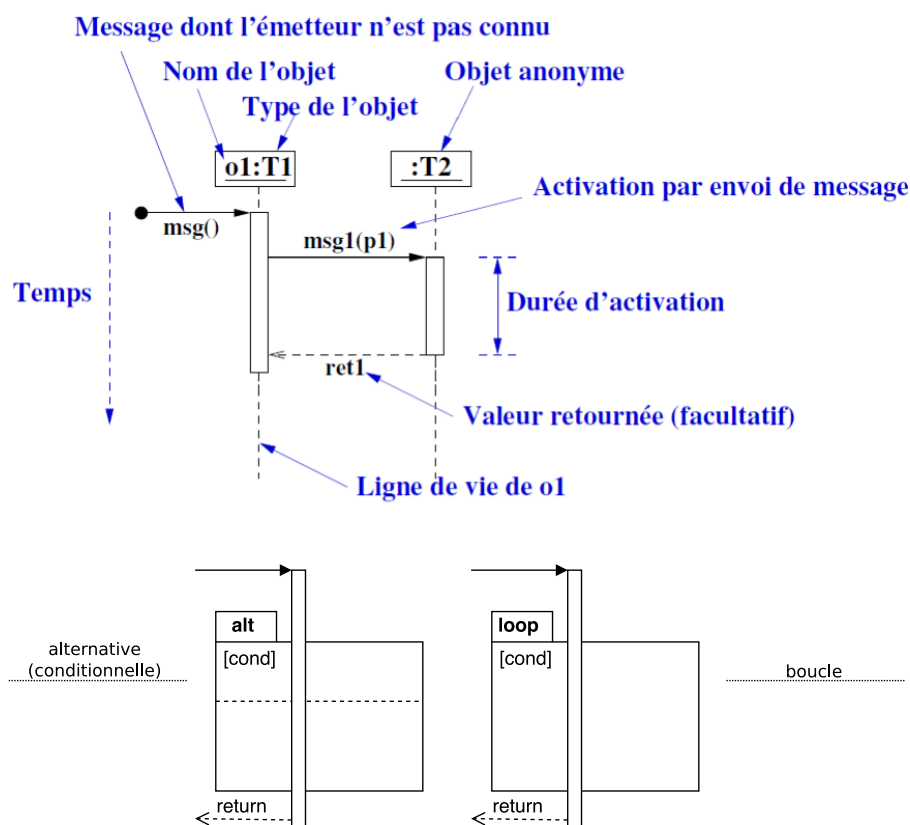


FIGURE 1 – Diagramme de séquence.

Le **diagramme de communication** est un diagramme d'interaction UML2.0 (appelé diagramme de collaboration en UML1). Il met l'accent sur la représentation des collaborations/-communications entre l'ensemble des objets intervenants dans un scénario. C'est un diagramme

équivalent au diagramme de séquence, l'accent est mis sur les relations entre objets plutôt que sur le temps.

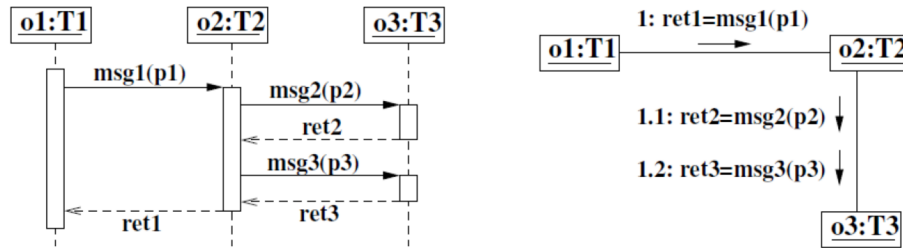
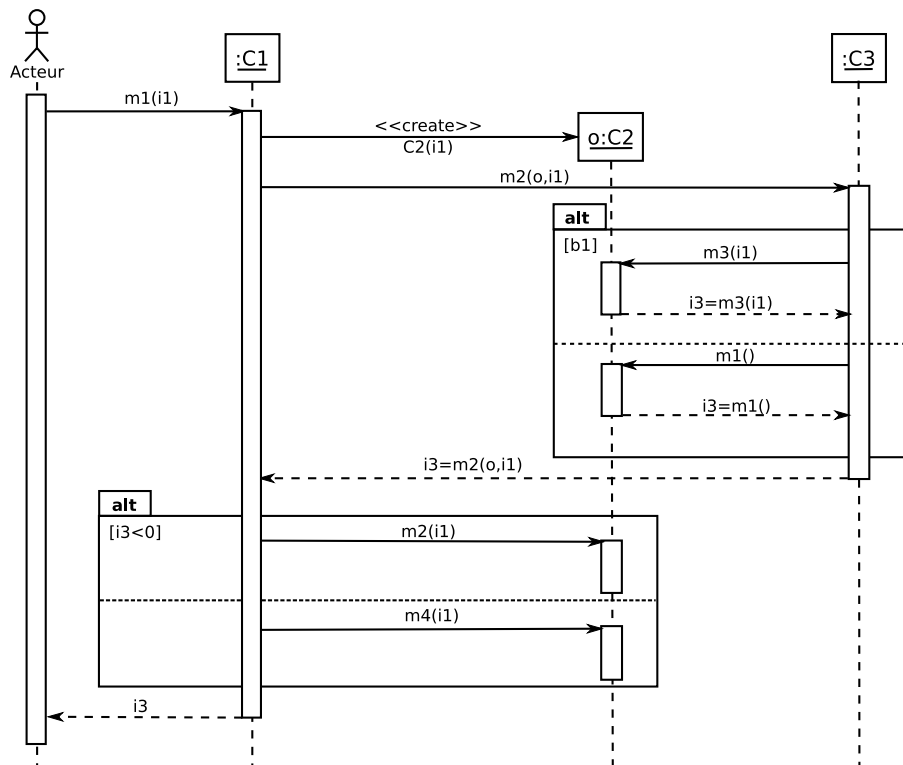


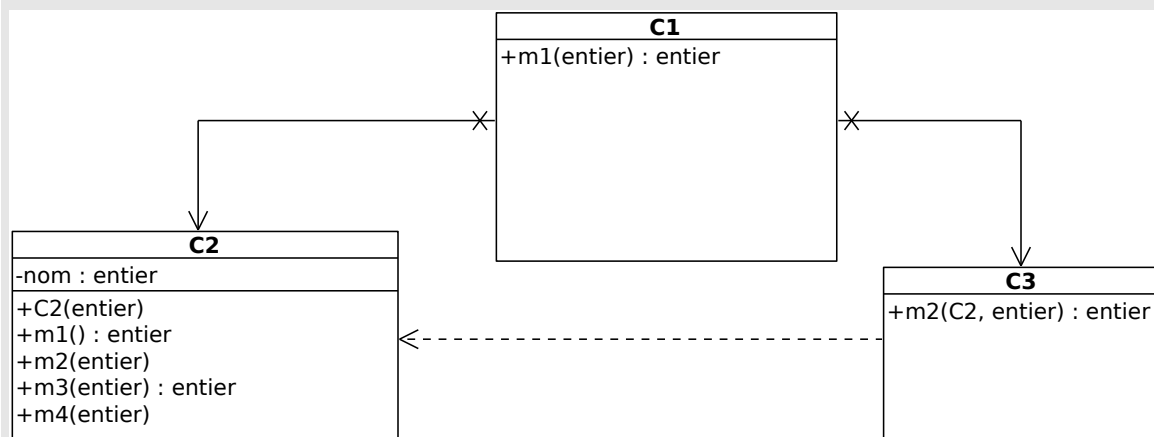
FIGURE 2 – Deux représentations équivalentes : à gauche un diagramme de séquence, à droite le diagramme de communication équivalent.

2 Diagramme de séquences et diagramme de classes

Question 1 *Proposer un diagramme de classes cohérent avec le diagramme de séquence ci-dessous (i_x sont des entiers, b_x des booléens).*

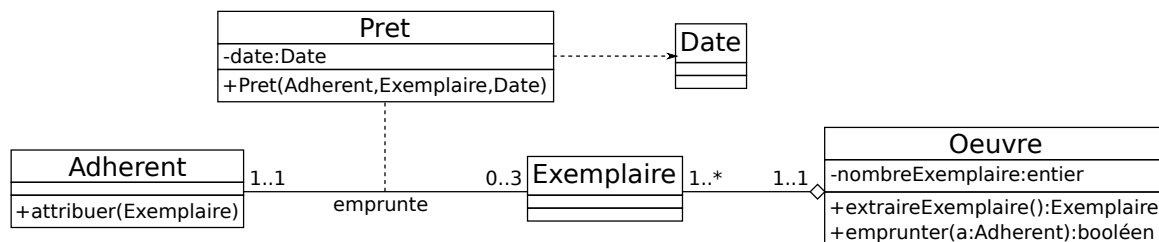


C1 crée une instance de C2 mais ne la détruit pas à la fin de la méthode m1, donc ce n'est probablement pas une variable locale mais C1 a un attribut de type C2. Cette association pourrait éventuellement être une composition (puisque l'instance de C2 est créée par C1). C1 connaît C3 sans l'avoir créé, donc C3 attribut de C1. Les variables i3 de C1 et C3 sont a priori des variables locales (pas nécessaire d'être attributs). C2 a probablement un attribut entier (dont le nom est inconnu) car il prend un entier en paramètre du constructeur.



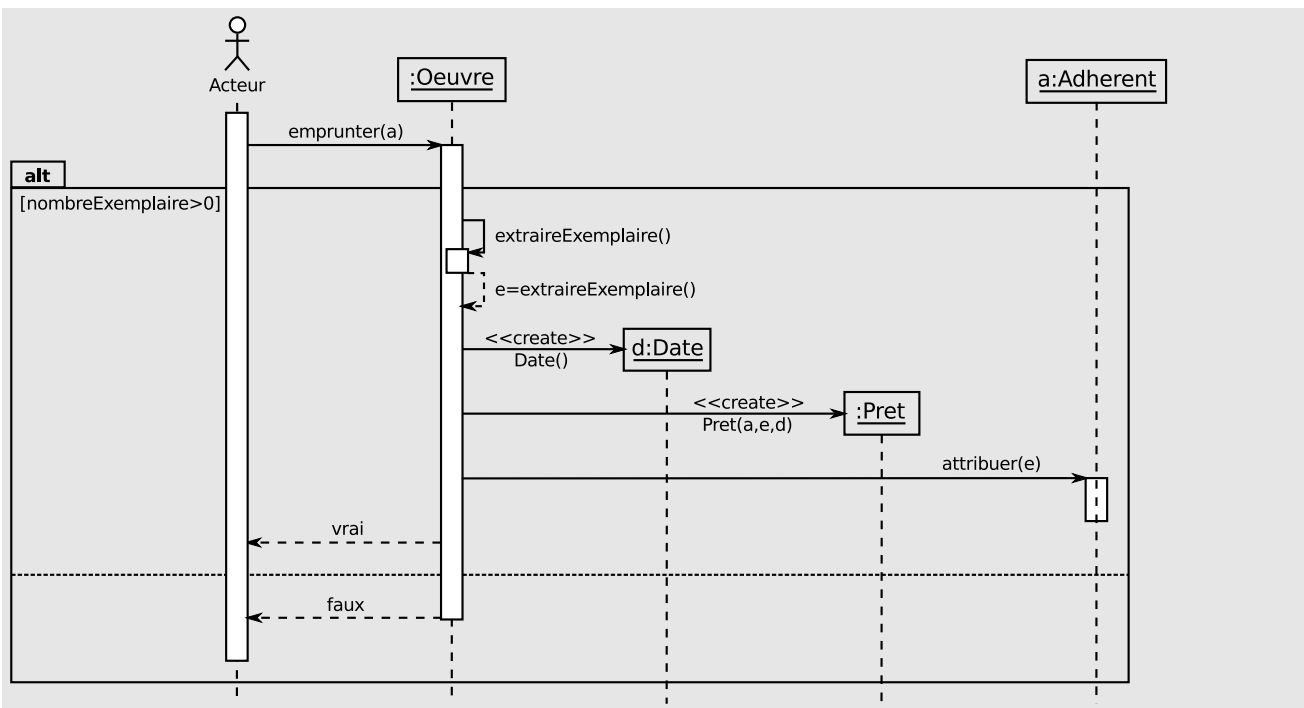
3 Gestion des prêts

Le diagramme de classes présenté ci-dessous modélise une partie de la structure interne du système de gestion des prêts d'une bibliothèque.



Un adhérent peut emprunter un exemplaire 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 **extraireExempleire**, une instance de la classe **Pret** est créée, puis l'exemplaire extrait de la bibliothèque est attribué à l'adhérent grâce à l'opération **attribuer**. S'il restait un exemplaire, la méthode retourne vrai ; dans le cas contraire, elle retourne faux.

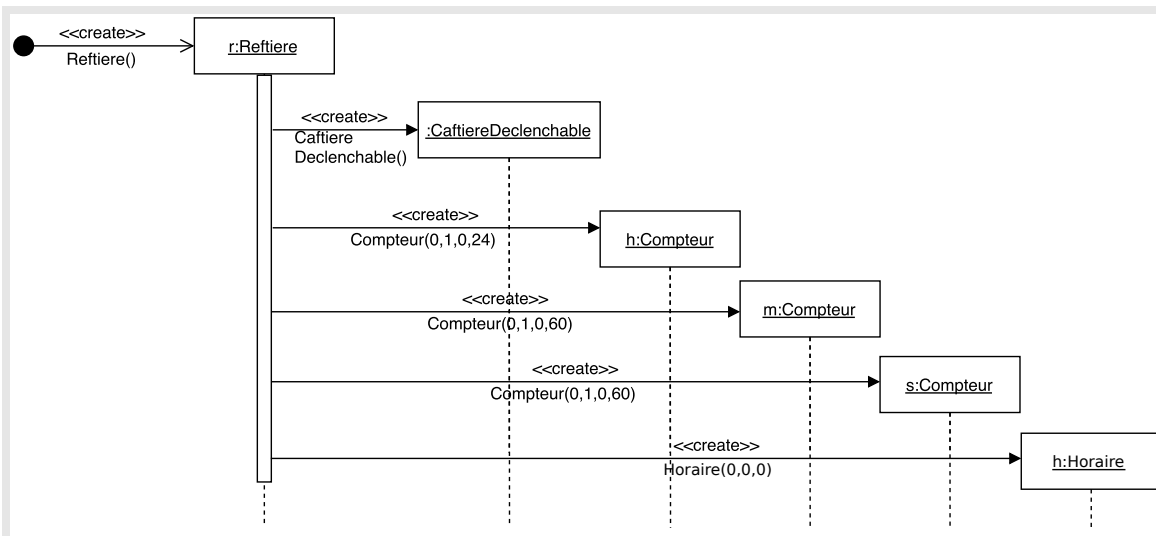
Question 1 Donner le diagramme de séquences du scénario : *Un exemplaire d'une œuvre est emprunté par un adhérent détaillé ci-dessus.*



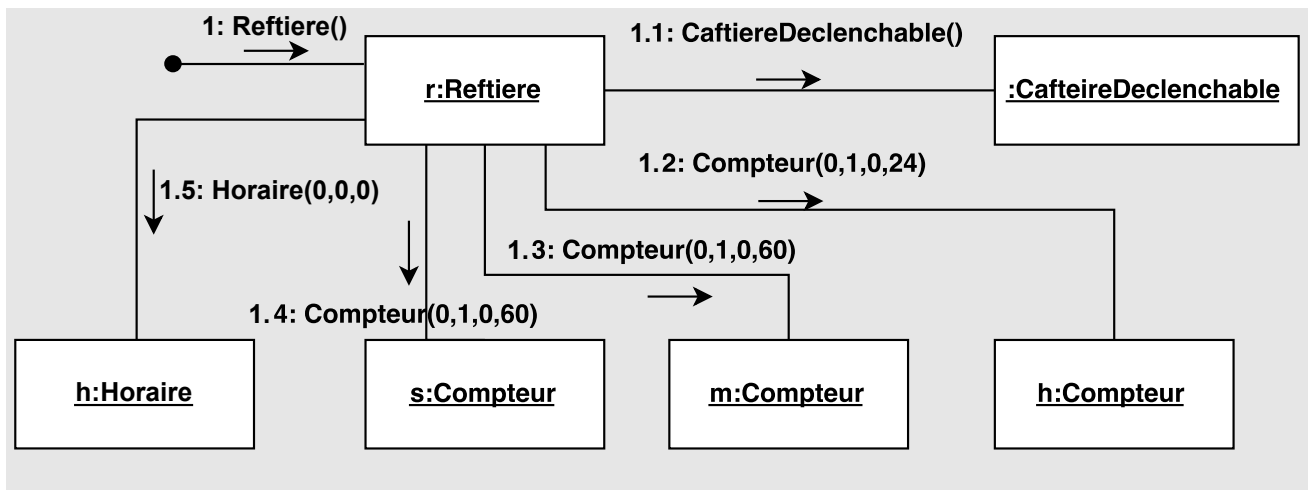
4 Reftière et instantiation

Soit la plébiscité **Reftiere** conçue lors du TD précédent, dont le diagramme est rappelé ci-après.

Question 1 *Illustrer par un diagramme de séquence la création d'une Reftière.*

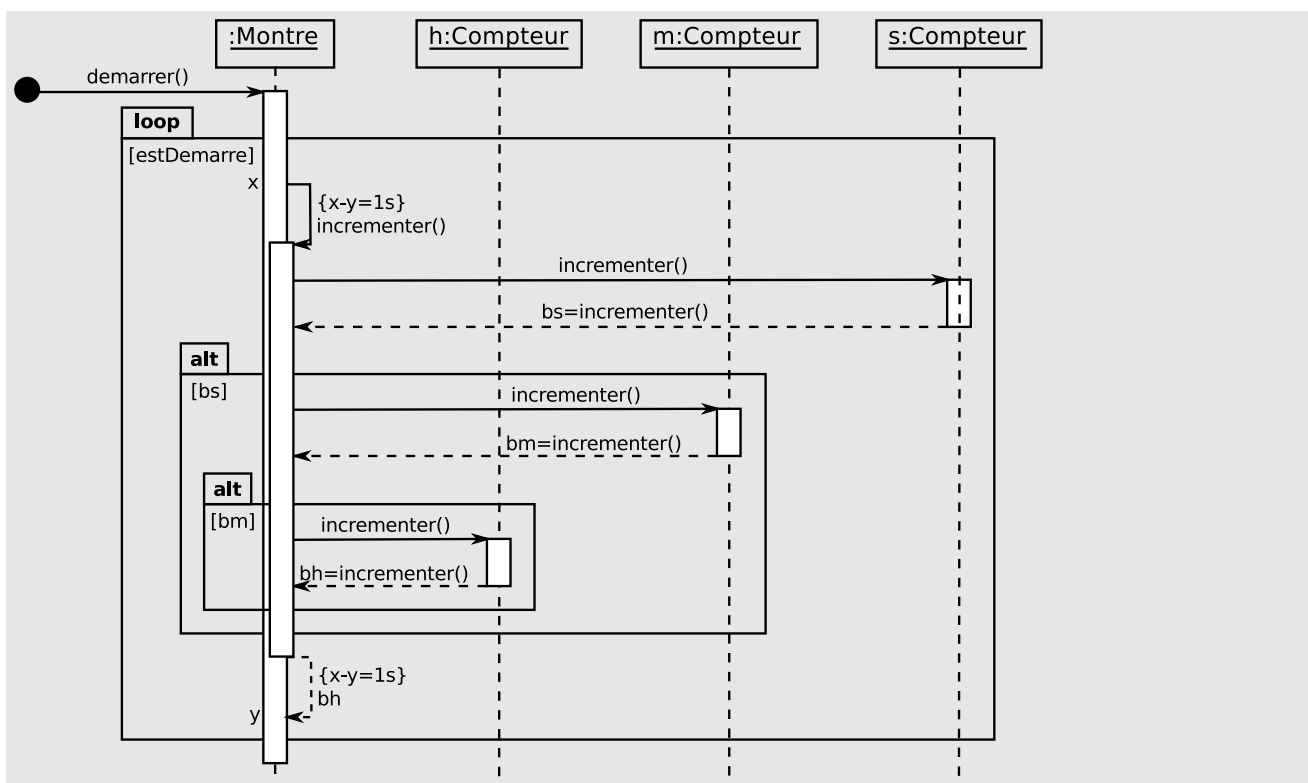


Question 2 *Proposer maintenant un diagramme de communication équivalent.*



5 Incrémentation

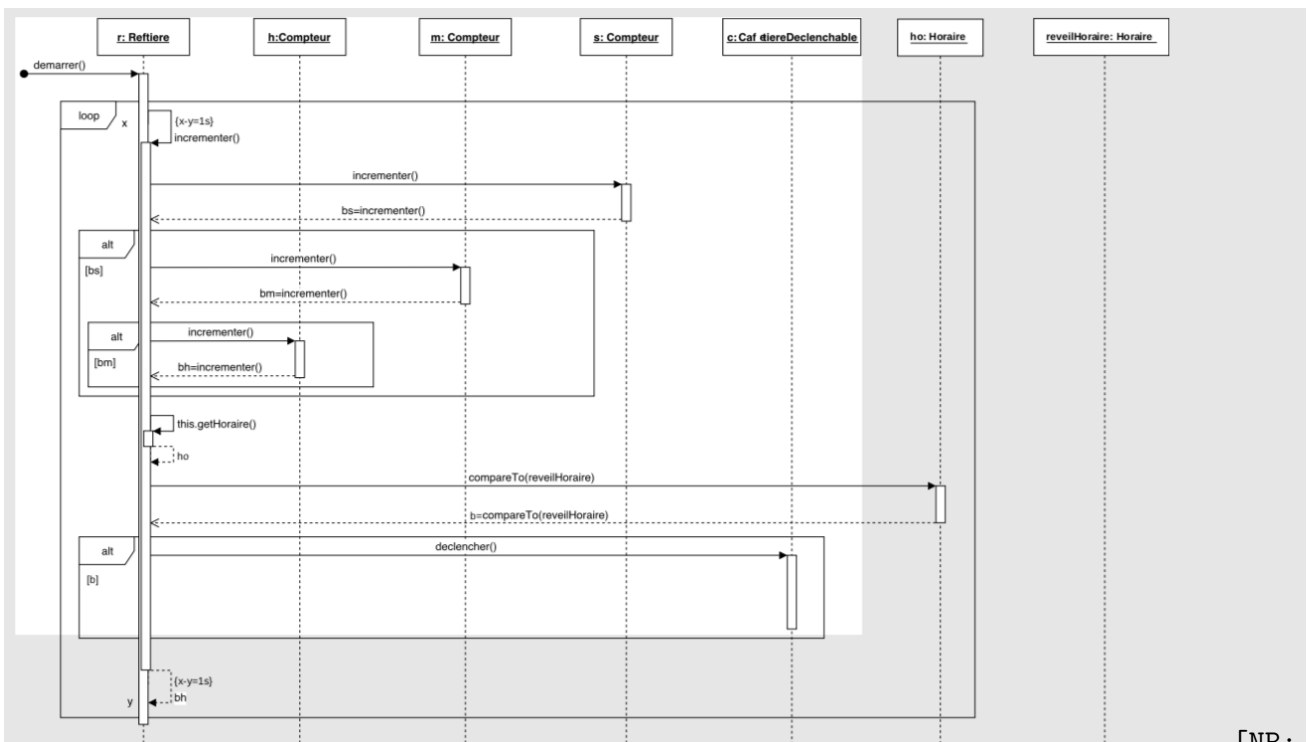
Question 1 Donner le diagramme de séquence du scénario suivant : Une instance de montre démarre. Il faut particulièrement penser à traduire le cas où un compteur atteint sa valeur maximum.



Commentaire :

- À gauche de la barre d'activation de l'objet anonyme montre, x et y représente des jalons temporels ;
- Sur ladite barre, les messages pointant qui, à la fois, partent et pointent vers elle sont dit réflexifs. Un message réflexif ne représente pas l'envoi d'un message, il représente une activité interne à l'objet ;
- Les éléments entre accolades représentent des contraintes temporelles. [NB: formellement la différence entre x et y ne devrait pas être de 1s mais de `s.getPas()` (ou d'un attribut si l'on souhaite éviter l'appel de fonction)]

Question 2 Donner maintenant le diagramme de séquence du scénario suivant : Une instance de reftière démarre. Même remarque que précédemment.



[NB:

formellement la différence entre x et y ne devrait pas être de 1s mais de s.getPas() (ou d'un attribut si l'on souhaite éviter l'appel de fonction)]

Question 3 Proposer un diagramme de communication équivalent.

