

# Thème 11 - TD

## Objectifs

— Barrière de synchronisation

## Exercice

### Exercice 17 – Petit Java Noël...

Noël approche, il est temps pour les 9 rennes du Père Noël de rentrer de leurs vacances dans le Pacifique Sud... Comme toujours, ils ont attendu la dernière minute et le Père Noël est impatient. Dès qu'il en voit arriver un, il s'empresse de l'atteler au traineau. Les rennes sont polis : ils vont saluer le Père Noël pour lui signaler leur retour. Mais ils sont ardents : il faut empêcher un renne de commencer à tirer le traineau avant que tous soient attelés.

#### Question 1

Comment garantir qu'un renne ne commence pas à tirer le traineau avant que tous soient attelés ?

#### Question 2

Proposez une implémentation de la classe `Renne` (un renne appelle la méthode `sayHello` de la classe `PereNoel` pour signaler son retour) et écrivez une classe permettant de tester l'exécution de ce système.

Le froid rend le Père Noël maladroit, il a du mal à atteler les rennes et cela prend du temps pour chacun. Les rennes peuvent donc se retrouver à faire la queue. Un des lutins propose au Père Noël d'exécuter les opérations suivantes :

```
import java.util.concurrent.locks.*;

public class PereNoel implements Runnable {
    private final int NB_TOTAL_RENNES;
    private int nbRennesAtteles = 0;
    private boolean busy = false;

    public PereNoel(int nb) {
        NB_TOTAL_RENNES = nb;
    }

    public synchronized void sayHello() throws InterruptedException {
        while (busy) {
            wait();
        }
        busy = true;
        notify();
    }

    private synchronized void attelerRenne() throws InterruptedException {
        while (! busy) {
            wait();
        }
        Thread.sleep(300); // Ca prend du temps d'attacher le bestiau !
        nbRennesAtteles++;
        System.out.printf("Encore un renne attelé. ");
        System.out.printf("Il y en a maintenant %d\n", nbRennesAtteles);
        busy = false;
        notify();
    }

    public void run() {
```

```
try {
    while (nbRennesAtteles != NB_TOTAL_RENNES) {
        attelerRenne();
    }
    Thread.sleep(100); // Le Pere Noel a du mal a monter !
    System.out.println("Je monte dans le traineau !");
}
catch (InterruptedException e) {
    System.out.println("Pere Noel interrompu en plein travail !");
}
}
```

### Question 3

Le Père Noël est fâché parce qu'il se trouve coincé (il y a 9 rennes à atteler et l'exécution produit l'affichage ci-dessous) :

```
Encore un renne attele. Il y en a maintenant 1
Encore un renne attele. Il y en a maintenant 2
```

Aidez le lutin et expliquez-lui pourquoi.

### Question 4

Proposez deux implémentations alternatives des méthodes `sayHello` et `attelerRenne`, l'une à base de `Lock` et l'autre à base de `Semaphore`. Vous devez garantir que le Père Noël attèle tous les rennes.

### Question 5

Tous les rennes sont-ils bien attelés au moment du démarrage du traineau ? Si nécessaire, modifiez les méthodes `sayHello` et `attelerRenne`.

Le Père Noël a beaucoup mangé pendant sa période d'inactivité, il est donc un peu lourd et a du mal à monter dans le traineau.

### Question 6

Modifiez le programme pour garantir que les rennes ne partent pas avant que le Père Noël se soit hissé dans le traineau.