

# TD 2 sur les threads

---

**Objectifs** : Les `Threads` – Synchronisation avec le mot-clé `synchronized` et les méthodes `wait()` et `notify()` de la classe `Object`.

---

## Exercice 1 : Le dîner des philosophes

Cinq philosophes sont assis en rond autour d'une table ronde.

Chaque philosophe passe sa vie alternativement à penser et à manger. Au centre de la table est déposé un immense plat de nouilles.

Malheureusement, le maître d'hôtel lorsqu'il a dressé la table a déposé sur celle-ci seulement 5 assiettes et 5 fourchettes et tout le monde sait que les philosophes utilisent deux fourchettes pour manger un plat de nouilles.

Une fourchette est placée entre chaque philosophe et comme chaque philosophe est bien élevé, il utilise uniquement la fourchette placée à sa droite et celle placée à sa gauche. Un philosophe ne prend pas une fourchette qui est déjà détenue dans la main d'un autre philosophe.

Une fois qu'il a mangé, un philosophe repose les deux fourchettes qu'il a utilisé.

- 1.1 Identifiez sur un schéma UML les différentes classes de l'application
- 1.2 Quel est le problème de synchronisation et où doit-on le gérer ?
- 1.3 Développer la première version de l'application.
- 1.4 Peut-il y avoir une situation d'inter blocage (aucun philosophe ne mange) ? Comment peut-on la corriger ?

## Exercice 2 : Les lecteurs / rédacteurs

Dans ce problème, des lecteurs et des rédacteurs travaillent sur un texte.

A tout moment, si un rédacteur écrit, il doit être seul (pas d'autre rédacteur en train d'écrire et aucun lecteur en train de lire).

Les lecteurs, eux, peuvent lire en parallèle.

Les lecteurs et les rédacteurs lisent/écrivent pendant 500 ms et attendent pendant 500 ms.

- 1.5 Identifiez sur un schéma UML les différentes classes de l'application
- 1.6 Quel est le problème de synchronisation et où doit-on le gérer ?
- 1.7 Développer la première version de l'application.
- 1.8 Peut-il y avoir une situation de famine des rédacteurs (il y a toujours un lecteur en train de lire et donc les rédacteurs ne peuvent pas écrire) ?