

# Mini-projet en programmation concurrente Java

2017-2018

## Travail à rendre

Un compte-rendu est attendu pour ce TP. Il est à fournir sous forme électronique sur M@doc. Le travail à rendre est sous forme d'archive zip, qui comprendra :

- le code source **commenté** de vos programmes ;
- un script ant ou bash commenté permettant **de construire et lancer vos programmes** ;
- un dossier détaillant les réponses aux questions, au format **pdf**.

L'archive est à déposer sur madoc avant le vendredi 11 novembre 2017 à 23h55. Le serveur de dépôt sera bloqué après cette date.

## 1 Exercice 1

Un problème classique de la programmation concurrente est celui du « *Dîner des Philosophes* » qui s'énonce comme suit :

*Cinq philosophes sont assis autour d'une table ronde. Devant chaque philosophe se trouve un bol de riz. À gauche de chaque assiette se trouve une baguette. Ces philosophes occupent la majorité de leur temps à réfléchir. Mais entre deux périodes d'intense réflexion, un philosophe a besoin de se nourrir. Pour manger, il doit commencer par prendre deux baguettes : celle qui est à gauche de son bol et celle qui est à gauche du bol de son voisin de droite. Il peut alors manger un peu de riz. Une fois rassasié, il repose les deux baguettes. Faisons l'hypothèse que chaque philosophe mange pendant un temps fini et que le bol contient tellement de riz qu'il en restera toujours. Expliquez comment les philosophes peuvent s'assurer qu'aucun d'entre eux ne mourra jamais de faim (pas de famine, donc).*

1. Étudiez, construisez et exécutez le programme `prog2.concur.exercice1.Philosophe`. Que constatez vous ?
2. Expliquez la cause du problème et proposez une correction.
3. Implémentez et testez votre solution.

## 2 Exercice 2

1. Étudiez le code de la classe `prog2.concur.exercice2.AbstractFileBloquanteBornee`.
2. Décrivez les principes d'une réalisation complète de cette classe.
3. Implémentez une réalisation utilisant l'API bas niveau. Accompagnez cette implémentation d'un programme principal permettant de tester votre solution.
4. Implantez une réalisation utilisant l'API haut niveau. Vous accompagnez cette implantation d'un programme principal permettant de tester votre solution.