

Master Informatique

Programmation objet concurrente

Responsable	Descriptions	Informations
Remi MORIN remi.morin@univ-amu.fr	Code : S51IN1M4 Nature : Unité d'enseignement Domaines : Sciences et Technologies	Composante : Faculté des Sciences

CONTENU

Depuis la généralisation des architectures multi-cœurs, la programmation parallèle est devenue incontournable pour développer des applications exploitant pleinement les capacités de traitement offertes par les ordinateurs actuels. Elle est aussi un moyen de simplifier la structure du logiciel en l'organisant sous la forme de tâches distinctes qui interagissent entre elles afin de répondre aux requêtes de l'utilisateur (ou de l'environnement).

Cette UE présente aux étudiants du M1 les principales difficultés de la programmation multi-thread et les techniques classiques permettant de développer un code **correct et performant**. Tout d'abord, les instructions de base liées à la classe Thread en Java sont introduites avec les concepts sous-jacents de **verrou** et de **variable de condition**. L'effet de ces instructions sur l'état d'un thread est illustré sur des exemples simples s'appuyant sur des diagrammes de séquence. Les notions d'indépendance et d'**atomicité** permettent ensuite de spécifier précisément les problèmes classiques de **synchronisation** en séparant le besoin d'exclusion mutuelle des contraintes d'attente particulières. L'approche recommandée pour débuter est celle de la conception et de la programmation de **moniteurs** (à la Hoare), étudiée sur plusieurs exemples classiques. Les outils dédiés à la programmation parallèle en Java sont également présentés en cours et exploités en Travaux Pratiques: locks divers, pools de threads, collections synchronisées ou concurrentes, objets atomiques, etc. Les difficultés propres à la **programmation sans verrou** sont illustrées par la construction de structures de données simples et de verrous performants.

Enfin, un aperçu du modèle mémoire Java permet d'initier les étudiants à la notion de programme « **bien synchronisé** » et aux risques d'exécutions **inconsistantes séquentiellement** du fait des optimisations de codes réalisées lors de la compilation ou de l'exécution.

Description en cours d'actualisation.

VOLUME HORAIRE

- Volume total: 26 heures
- Cours magistraux: 10 heures
- Travaux dirigés: 8 heures
- Travaux pratiques: 8 heures

CODES APOGÉE

- SINAU05L [ELP]
- SINAU05J [ELP]

POUR PLUS D'INFORMATIONS

[Aller sur le site de l'offre de formation...](#)

