

TP n°1

Découverte de l'environnement de travail

1 Accès au site de l'UE (Moodle)

Toutes les ressources de l'UE seront mises en ligne sur Moodle. Outre la mise à disposition des documents et la publication des actualités de l'UE, ce site fonctionne comme un forum et vous permet de poser toutes vos questions à l'équipe enseignante de manière publique ou privée.

1. Connectez-vous sur le site accessible depuis <https://moodle-sciences.upmc.fr/>
2. Vérifiez que vous avez accès à l'ue Fosyma (code d'ue 4I202).

Le sujet de ce premier TP ainsi que les documents associés sont disponibles sur le site.

2 Prise en main de l'environnement de travail

L'ensemble des TPs de l'ue seront réalisés en JAVA sur la plateforme multi-agents JADE, et préparés pour l'ide Eclipse.

La présentation de la plateforme est disponible ici : jade.tilab.com

Vous trouverez dans l'onglet "Documentation/Tutorial" du site deux documents clés :

- **Jade Programming Tutorial**
- **Programmer's Guide**

Ces documents vous permettront de vous familiariser avec les différentes fonctionnalités de JADE et la façon de programmer les agents. Vous devez impérativement en prendre connaissance afin d'être en mesure de réaliser le projet.

Exercice 1 Première exécution

1. Récupérez le projet eclipse du TP1 disponible sur gitlab à l'adresse suivante :
<https://gitlab.com/herpsonc/startJade>
2. — Sur votre machine personnelle : Importez le dans Eclipse en tant que projet Maven¹ comme indiqué ici : <https://gitlab.com/herpsonc/startJade/wikis/home>.

1. Maven est un outil facilitant la gestion et la production de projets Java. Il vous permettra (notamment) de télécharger automatiquement les différentes bibliothèques dont vous aurez besoin, leur code source et la javadoc par l'appel de la commande *maven install*. Si Maven n'est pas installé dans votre version d'eclipse, faire *Help/Install new software/work with all available sites*, filtrer avec le mot clé "Maven" et installer m2e.

- Sur les machines de l'université : La version d'eclipse supportant le java (à lancer depuis un terminal via la commande *eclipse &E* et non depuis la liste des applications) n'est pas à jour et les étudiants ne disposent des droits suffisants pour corriger cela, rendant maven inopérant. Téléchargez l'archive de la version standalone du projet depuis le site de l'UE (moodle). Dézippez là dans votre workspace puis importez la comme un projet Java.

Ce projet contient un readme, des bibliothèques, une classe permettant de démarrer la plateforme et trois exemples simples :

exemple 1 Constitué de 3 classes, une pour chaque type d'agent, cet exemple présente le fonctionnement élémentaire des agents

exemple 2 Plus complexe, cet exemple sépare les classes définissant les comportements des classes d'agents. Cette organisation est à privilégier.

exemple3 Ce dernier exemple touche à la migration des agents, qui sera abordé en fin de semestre.

3. Lisez le readme et le code source pour en comprendre la structure puis exécutez l'exemple 1 comme indiqué sur le wiki (gitlab). Le sniffeur qui vous est présenté ici vous permettra de vérifier manuellement (dans un premier temps) le comportement de vos agents.
4. Exécutez ensuite l'exemple 2 en suivant les messages échangés par les agents grâce au sniffeur.

3 Protocoles, rôles et comportements

En vous aidant du code fourni dans le cadre des exercices 1 et 2 et de la documentation de Jade, vous allez maintenant implémenter quelques actions simples vous permettant de vous familiariser avec les notions de Protocoles, rôles et comportements.

Exercice 2 Communication 2-1 : Envoi de 10 et 15 entiers

En vous inspirant de l'exemple 2, écrivez un protocole permettant à un agent A et un agent C d'envoyer respectivement 10 et 15 messages à l'agent B (chaque message contenant un entier tiré aléatoirement). L'agent B doit recevoir les différents entiers, réaliser la somme propre à chaque émetteur puis lui retourner son résultat (l'agent A reçoit donc la somme des 10 entiers, et l'agent C la somme de ses 15 entiers). Les agents A et C doivent afficher le résultat obtenu dans le terminal lors de sa réception.

Exercice 3 Communication n-1 : Envoi de k entiers

Écrivez un protocole permettant à un agent *somme* de demander à l'ensemble des autres agents de lui envoyer des nombres entiers (tirés aléatoirement). L'agent *somme* doit réceptionner les différents entiers jusqu'à atteindre *k* entiers reçus, puis demander aux agents d'arrêter de transmettre de nouvelles valeurs.

Exercice 4 Communication n-n

Réfléchissez à la généralisation de chaque protocole des exercices de façon à ce qu'il soit possible d'ajouter autant d'agents de n'importe quel rôle sans que cela ne pose problème. Quels sont les éléments à prendre en compte, que faudrait-il modifier dans vos différents protocoles ?