

## IDM TP1 - Metadonnées, Métaprogrammation et Ingénierie des modèles

**Objectif :** découverte des concepts les plus avancés en ingénierie logicielle. Méta-connaissance – description de connaissances. Découverte pratique des métadonnées, de l'introspection et de la méta-programmation (Apprendre à apprendre). Tester quelques aspects concrets de l'ingénierie des modèles.

**Notebook :** préparer un fichier avec vos réponses (penser à noter / citer les sources que vous utilisez) ce document sera utile pour l'examen final.

Travail en TP (puis personnel) :

1. En vous aidant des ressources sur Internet, proposer des définitions pour les concepts de métadonnée, d'introspection, de méta-programmation et de réflexivité. Décrire leurs articulations possibles.
2. Rechercher ce que signifie « l'ingénierie des modèles. »
3. En ce qui concerne les métadonnées, regarder dans un 1<sup>er</sup> temps le concept historique des annotations dans le langage Java. L'introduction des annotations s'est effectuée à partir de la version 1.5. Une courte synthèse est donnée dans le lien ci-dessous sur l'ancien site de référence de Sun pour langage Java :

<http://java.sun.com/j2se/1.5.0/docs/guide/language/annotations.html>

Vous complèterez au besoin avec des URLs plus récentes Java 13..16 et suivants ou tout autre site Web que vous jugerez utiles.

Consultez également les sites sur l'Ingénierie Des Modèles, le MDA etc (le site de l'OMG) et en français (Actions spécifiques du CNRS – « Model Driven Engineering » ou d'autres sources que vous jugerez pertinentes).

4. Un article de recherche a utilisé les techniques de meta-programmation dès l'introduction de l'introspection en Java. Etudier la technique proposée (PDF de l'article sur l'url du cours ~.../IDM-ZZ3), elle repose sur l'utilisation de l'ingénierie des modèles. Dès l'introduction des annotations, ce type d'outil a permis la production automatique d'interfaces utilisateur pour des logiciels de simulations et ce grâce à l'introspection de métadonnées (en Java dès l'introduction des annotations).
5. D'autres langages que Java offrent-ils un concept équivalent ? Essayer de trouver des discussions sur leurs atouts et faiblesses.
6. Considérer les limites de la RTTI (Run Time Type Interface) du C++. Regardez ce qu'est la RTTI si vous ne connaissez pas encore, que proposeriez-vous pour intégrer et mettre simplement en œuvre dans le langage C++ par exemple, les concepts plus évolués que vous avez découvert dans les points 1 à 3.

**Rappel :** pour chaque partie, penser à noter vos sources (bibliographie, Webographie, définitions retenues, explications etc...) est une bonne pratique. Les documents de TP sont disponibles pour l'examen final.