



Qualité et productivité des outils logiciels

MGL7760 Groupe 40 – Hiver 2008

Louis Martin – UQAM

Version 0.01
2008-01-08

Historique des révisions

| Version | Date | Description | Auteurs |
|----------------|-------------|--------------------|----------------|
| 0.01 | 2008-01-08 | Version initiale | Louis Martin |

Table des matières

| | |
|--|----------|
| Historique des révisions | 2 |
| Sigles et abréviations | 4 |
| 1. Plan de cours | 5 |
| 1.1 Description du cours selon l'annuaire | 5 |
| 1.2 Objectifs | 5 |
| 1.3 Évaluation | 5 |
| 1.4 Calendrier | 6 |
| 1.5 Logistique | 6 |
| 1.6 Approche pédagogique | 7 |
| 1.7 Description des travaux | 7 |
| 1.7.1 Projet 1 – Mise en place et extension du framework Hibernate Validator | 7 |
| 1.7.2 Projet 2 – Mise en place et utilisation de AspectJ | 7 |
| 1.7.3 Projet 3 – Réalisation d'un langage dédié | 7 |
| Bibliographie | 8 |

Sigles et abréviations

| | |
|------------|------------------------------------|
| AOP | Aspect-Oriented Programming |
| DSL | Domain Specific Language |
| IDE | Integrated Development Environment |
| MOP | Meta-Object Protocol |
| S/O | sans objet |

1. Plan de cours

1.1 Description du cours selon l'annuaire

*Aperçu des outils pour assister au développement et à l'entretien des logiciels.
Plates-formes d'intégration des outils. Environnements de développement. Outils pour
la rétro-ingénierie des logiciels. Critères et stratégies d'évaluation des outils.*

1.2 Objectifs

- Initier les étudiants à la programmation par aspect.
- Comprendre la problématique des préoccupations transversales en orientée objet.
- Initier les étudiants à la méta programmation et au protocole méta-objet.
- Comprendre les caractéristiques des langages dynamiques.
- Familiariser les étudiants aux concepts des langages dédiés (domain specific language).
- Initier les étudiants à l'utilisation des annotations comme outil.
- Utiliser les annotations lors de l'exécution d'un programme.
- Approfondir l'utilisation des environnements intégrés de développement.
- Comprendre les tendances émergentes dans le domaine.
- Initier les étudiants aux nouvelles approches dans le domaine.

1.3 Évaluation

| Description sommaire | Date de remise | Pondération |
|---|---------------------------|-------------|
| Projet 1 – Mise en place et extension du framework Hibernate Validator | Jeudi, le 14 février 2008 | 20 % |
| Projet 2 – Mise en place et utilisation de AspectJ | Jeudi, le 13 mars 2008 | 20 % |
| Projet 3 – Réalisation d'un langage dédié | Jeudi, le 10 avril 2008 | 20 % |
| Présence et participation active en classe | Continu | 10 % |
| Examen final | Jeudi, le 24 avril 2008 | 30 % |

Un travail remis en retard reçoit la note zéro à moins d'avoir fait l'objet d'une entente **préalable** avec le professeur.

Le détail des conditions de réalisation de chaque travail est précisé avec la description du travail.

Au plus tard le mercredi de chaque semaine, les étudiants devront avoir transmis par courriel au professeur une feuille de temps décrivant les activités accomplies lors de la semaine précédente (la

date et l'heure de réception du courriel seront utilisées pour contrôler le respect de la date de tombée). La forme et le contenu de cette feuille de temps seront précisés en classe. Pour tout retard, 1 % sera retranché à la note finale de l'étudiant.

La qualité du français fait partie intégrante des critères d'évaluation des travaux et des examens jusqu'à un maximum de 25%.

La politique de tolérance zéro du département d'informatique sera appliquée à l'égard des infractions de nature académique.

La note de passage du cours est de 60% pour l'ensemble de l'évaluation et de 50% pour l'examen final.

1.4 Calendrier

Le contenu de chaque période dépendra des besoins des étudiants au fur et à mesure de la réalisation de leurs projets.

Il n'y a pas de livre obligatoire pour le cours. Cependant des articles disponibles sur le WEB devront être lus et seront des sujets couverts à l'examen.

| Période | Date | Contenu | Lectures et remises des travaux |
|---------|------------------|---------------------------|---|
| 1 | 2008-01-10 Jeudi | Introduction | |
| 2 | 2008-01-17 Jeudi | | |
| 3 | 2008-01-24 Jeudi | | |
| 4 | 2008-01-31 Jeudi | | |
| 5 | 2008-02-07 Jeudi | | |
| 6 | 2008-02-14 Jeudi | | Remise du Projet 1 – Mise en place et extension du framework Hibernate Validator |
| 7 | 2008-02-21 Jeudi | | |
| * | 2008-02-28 Jeudi | Semaine de relâche | |
| 8 | 2008-03-06 Jeudi | | |
| 9 | 2008-03-13 Jeudi | | Remise du Projet 2 – Mise en place et utilisation de AspectJ |
| 10 | 2008-03-20 Jeudi | | |
| 11 | 2008-03-27 Jeudi | | |
| 12 | 2008-04-03 Jeudi | | |
| 13 | 2008-04-10 Jeudi | | Remise du Projet 3 – Réalisation d'un langage dédié |
| 14 | 2008-04-17 Jeudi | Conclusion | |
| 15 | 2008-04-24 Jeudi | Examen final | |

1.5 Logistique

Les cours ont lieu les jeudis de 18:00 à 21:00 au PK-4665.

1.6 Approche pédagogique

Le cours se base sur une approche par projet. L'approche par projet s'inscrit dans l'esprit de la formation par compétence. Il permet la mobilisation des ressources de l'étudiant dans la réalisation d'une tâche authentique.

Les étudiants devront travailler en équipe (taille: de 1 à 4 étudiants maximum). Il est fortement conseillé de ne pas effectuer les projets seul. Les critères de correction et les attentes ne seront pas modifiés en fonction du nombre d'étudiants dans l'équipe. Chaque équipe formera un groupe de consultants à qui seront confiés des mandats à réaliser. Chaque projet sera réalisé par l'ensemble des équipes.

Chaque période de cours sera divisée en deux. La première partie abordera des thèmes choisis préalablement par les étudiants pour les aider dans la réalisation de leurs projets. Si aucun thème n'est proposé par les étudiants, un thème contemporain sera présenté par le professeur à son choix. La deuxième partie consistera à reviser avec chaque équipe le déroulement des travaux entrepris dans le but d'éviter des dérives éventuelles.

1.7 Description des travaux

1.7.1 Projet 1 – Mise en place et extension du framework Hibernate Validator

Étude des annotations comme outil de productivité, élaboration des critères d'évaluation, preuve de concept, normes et recommandations concernant les meilleures pratiques dans l'utilisation des annotations. Étude de cas avec Hibernate Validator. Utilisations possibles des annotations, autres que les utilisations courantes actuelles.

1.7.2 Projet 2 – Mise en place et utilisation de AspectJ

Étude du concept de la programmation par aspect, élaboration des critères d'évaluation, preuve de concept, normes et recommandations concernant les meilleures pratiques dans le domaine. La preuve de concept touche l'implémentation de deux préoccupations: la journalisation et la sécurité des accès.

1.7.3 Projet 3 – Réalisation d'un langage dédié

Étude du concept des langages dynamiques et des langages dédiés, élaboration des critères d'évaluation, preuve de concept, normes et recommandations concernant les meilleures pratiques dans le domaine. La preuve de concept touche l'implémentation de deux volets: une extension pour les unités de mesure et la réalisation d'un assembleur (builder) pour un état de compte.

Bibliographie

Livres

- [BL-1] CARLSON, David
Eclipse Distilled
2005, Addison Wesley – ISBN: 0321288157
- [BL-2] D'ANJOU, Jim; FAIRBROTHER, Scott; KEHN, Dan; KELLERMAN, John; MCCARTHY, Path
The Java Developer's Guide to Eclipse (2ième édition)
2005, Addison Wesley – ISBN: 0321305027
- [BL-3] LADDAD, Ramnivas
AspectJ in Action – Practical Aspect-Oriented Programming
2003, Manning – ISBN: 1930110936

Sites Web

- [BW-1] <http://www.eclipse.org/>
Eclipse
Page d'accueil
- [BW-2] <http://groovy.codehaus.org/>
Groovy
Page d'accueil
- [BW-3] <http://martin.louis.uqam.ca/>
Louis Martin
Portail