

IFT 785 Approches orientées objets

Plan de cours

Professeur : Sylvain Giroux

Session : hiver 2014

Horaire : lundi, 13h30 à 16h20

Local : laboratoire DOMUS.

Objectif général

- Connaître les concepts fondamentaux des approches orientées objets;
- Concevoir le développement de systèmes informatiques dans une optique orientée objets;
- Comparer des méthodes et outils orientés objets afin de pouvoir choisir ceux qui conviennent le mieux dans le développement de systèmes particuliers.

Objectifs spécifiques

A la fin du cours, l'étudiant devra être capable de :

- Maîtriser les concepts fondamentaux de différents paradigmes liés à la programmation par objets, en particulier Java, la programmation par acteurs et la programmation par aspects;
- Décrire, utiliser et comparer des méthodes et outils de conception objets, par exemple UML et la programmation agile;
- Décrire et appliquer les principaux patrons de conception (*design patterns*);
- Décrire et appliquer les procédures de factorisation de programmes orientés objets;
- Appliquer et intégrer les procédures de tests dans les programmes objets;
- Utiliser et concevoir des *frameworks* orientés objets;
- Manipuler la réflexivité et le concept de métaclasses;

Contenu

- Concepts de base : objet, acteur, agent, classe, message, héritage, délégation, instanciation, clonage, spécialisation, différenciation, classe abstraite, généricité, polymorphisme, persistance;
- Concepts avancés: *frameworks*, métaclasses, réflexivité, introspection;
- Méthodes d'analyse et de conception orientées objets: UML, programmation extrême;
- Étude de langages orientés objets, tels que Smalltalk, Java, ObjVlisp, ABCL, ACTORS;
- Outils pour le développement orienté objet: JUnit
- Outils pour la programmation par aspects : AspectJ

Organisation du cours semaine par semaine

Durée (sem.)	Thème	Contenu	Références
1	Variations en POO	Classe, instance, héritage, méthode, message.	[1]
1	Factorisation	Différenciation, classe abstraite.	[3]
2	Patrons de conception	Observateur, délégation, clonage, généricité, polymorphisme.	[4]
1	Frameworks orientés-objets		[5]
2	Méthodes de conception objets	Programmation extrême; Tests unitaires	[6]
2	Réflexivité, métaclases	ObjVLisp, Réflexivité en Java	[7]
1	Acteurs	Actalk, Actors, ABCL/1	[8] [9] [10]
1	Programmation par aspects	AspectJ	[11]
1	Méthodes de conception objets	UML, modélisation et diagrammes.	[2]
2	Projet		
1	Examen		

Evaluation

Projet	/35	Projet individuel ou projet en équipe de 6 à 10
Présentation orale	/15	Analyse et présentation d'un patron de conception
Examen final	/50	Durée : 3 jours

Matériel pédagogique

- Aucun livre obligatoire.
- Les transparents seront disponibles sur Moodle avant le cours.
- Les fichiers .pdf de certaines références seront aussi rendus disponibles.

Quelques références

- [1] A. Goldberg, *Smalltalk-80, The Language*, Addison Wesley Professional; 1989.
- [2] M. Fowler, *UML Distilled: A Brief Guide to the Standard Object Modeling Language*, Addison-Wesley, 2003.
- [3] Martin Fowler, et al., *Refactoring: Improving the Design of Existing Code*, Addison-Wesley; 1999.
- [4] Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides, *Design Patterns*, Addison-Wesley;1995.
- [5] Roberts, D., Johnson, R. (1997) Evolving Frameworks A Pattern Language for Developing Object-Oriented Frameworks, in R. Martin, et al. eds, *Pattern Languages of Program Design*, vol.3, Addison-Wesley, 1997.
- [6] Kent Beck, *Extreme Programming Explained: Embrace Change*, Addison-Wesley Pub Co; 1999.
- [7] P. Cointe, *Metaclases are First Class: the ObjVLisp model*, OOPSLA '87 Proceedings, 1987, Orlando, Florida, SIGPLAN Notices, vol. 22, no 12, pp. 156-167.
- [8] Agha, Gul, *Actors: A Model of Concurrent Computation in Distributed Systems*. MIT Press, 1986.
- [9] Briot, Jean-Pierre, *From Objects to Actors: Study of a Limited Symbiosis in Smalltalk-80*, LITP 88-58 RXF, Université Pierre et Marie Curie, France, sept. 1988.
- [10] A. Yonezawa, et al., *Object-Oriented Concurrent Programming in ABCL/1*, OOPSLA 86 Proceedings, SIGPLAN Notices, vol. 21, no 11, pp. 258-268.
- [11] T. Elrad, et al., *Discussing aspects of AOP*, Comm. of the ACM Vol. 44 , no 10, Oct. 2001, pp. 33 -38.