

GROUPE	20 VILLEMAIRE, Roger	villemaire.roger@uqam.ca	(514) 987-3000 6744	PK-4615
	Mardi, de 18h00 à 21h00			

DESCRIPTION	Introduction à certaines notations formelles pour décrire les exigences et les spécifications de systèmes logiciels. Méthodes pour les systèmes séquentiels (tel que le langage Z ou la notation de Mills) et pour les systèmes concurrents et réactifs (tels que les machines d'états et les réseaux de Petri avec certaines extensions concernant les données). Utilisation des méthodes formelles pour l'analyse des propriétés et du fonctionnement des systèmes au niveau de la spécification, de la conception ou de l'implantation.
-------------	--

OBJECTIF	<p>Le cours vise à initier les étudiant-e-s aux méthodes formelles et à leur rôle dans le cycle de développement des logiciels; il vise aussi à faire comprendre les avantages et les limites de ces méthodes.</p> <p>À la fin du cours, l'étudiant-e devrait être capable :</p> <ul style="list-style-type: none"> <input type="checkbox"/> d'expliquer le rôle des méthodes formelles dans le cycle de vie des logiciels; <input type="checkbox"/> de lire et écrire des exemples de spécifications formelles dans diverses notations; <input type="checkbox"/> d'identifier les avantages et limites de l'utilisation des méthodes formelles; <input type="checkbox"/> d'évaluer et juger de l'opportunité d'utiliser ou non des méthodes formelles pour divers types d'applications.
----------	--

ÉVALUATION	Description sommaire	Date	Pondération
	Travail de modélisation (proposition)		20 %
	Travail de modélisation (ébauche)		20 %
	Travail de modélisation (rapport final complet)		20 %
	Participation active en classe		10 %
	Examen		30 %

Formules pédagogiques

- Exposés magistraux
- Exercices pratiques en classe
- Travail de modélisation

Les règlements concernant le plagiat seront strictement appliqués. Pour plus de renseignements, consultez le site suivant : <http://www.sciences.uqam.ca/etudiants/integrite-academique.html>

CONTENU	<ul style="list-style-type: none"> <input type="checkbox"/> Introduction aux méthodes formelles en génie logiciel. <input type="checkbox"/> Modélisation, spécification, vérification de systèmes OO. <input type="checkbox"/> Formalisations comme artéfact : descriptions et traitement. <input type="checkbox"/> Vérification à l'exécution. <input type="checkbox"/> Vérification par construction.
---------	--

RÉFÉRENCES	V C MILES, R. and HAMILTON, K. – <i>Learning UML 2.0</i> – O'Reilly Media, 2006. Disponible dans Safari Books Online (accessible aux membres de l'IEEE-CS http://www.computer.org/ ou de l'ACM http://www.acm.org/).
	V C PILONE, D. and PITMAN, N. – <i>UML 2.0 in a Nutshell</i> – O'Reilly Media, 2005. Disponible dans Safari Books Online (accessible aux membres de l'IEEE-CS http://www.computer.org/ ou de l'ACM http://www.acm.org/).
	U C www.info2.uqam.ca/~villemaire_r/7160.html Site web du cours (plan de cours, exemples vus en classes, etc.).
	V C WARMER, J. and KLEPPE, A. – <i>The Object Constraint Language - Second Edition : Getting Your Models Ready for MDA</i> – Addison-Wesley, 2003.
	V C BECKERT, B., HÄHNLE, R. and SCHMITT, P.H. (Eds.) – <i>Verification of Object-Oriented Software: The Key Approach</i> – Springer-Verlag, LNCS 4334. – http://www.springerlink.com/openurl.asp?genre=issue&issn=0302-9743&volume=4334&issue=preprint
	V C WILKE, C., THIELE, M., FREITAG, B. and Lars SCHÜTZE – <i>Dresden OCL, Manual for Installation, Use and Development</i> – Technische Universität Dresden, Faculty of Computer Science, 2012.
V C JACKSON, D. – <i>Software Abstractions: Logic, Language, and Analysis</i> – The MIT Press, 2006.	

^{VC} PELED, D.A. – *Software Reliability Methods* – Springer-Verlag, 2001.

A : article – C : comptes rendus – L : logiciel – N : notes – R : revue –
S : standard – U : uri – V : volume

C : complémentaire – O : obligatoire – R : recommandé