

Méthodes et Techniques du Développement du Logiciel

Conf. dr. Maria-Iuliana Dascalu

mariaiuliana.dascalu@gmail.com

www.mariaiulianadascalu.com

Faculté d'Ingénierie en Langues Étrangères, UPB

Informations administratives

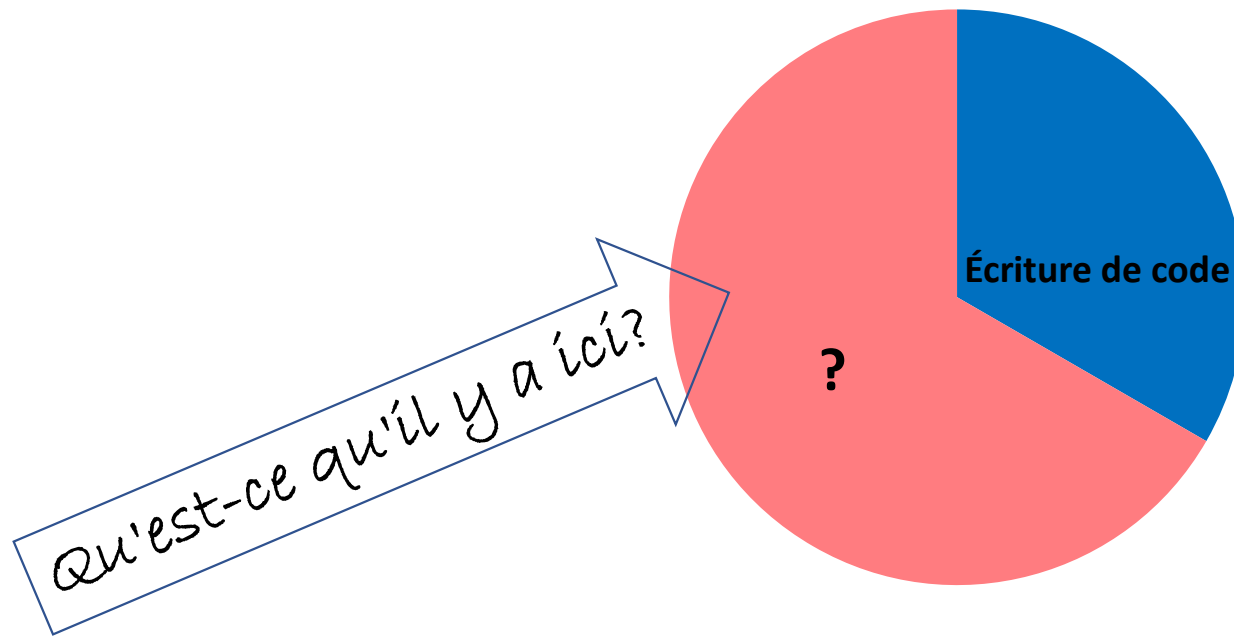
- 4 ECTS
- Si vous avez des questions:
 - mariaiuliana.dascalu@gmail.com (cours)
 - iulia.stanica@gmail.com (TP, Projet)
 - le sujet de e-mail doit être: [MTDL] [nom de student]; example:
MTDL Popescu Ion

Évaluation

- La note finale $E = E1 + E2 + E3 + E4 + E5 + E6$
 - E1 - examen écrit pendant la session: 4p
 - E2 - activité pendant les travaux pratiques: 1p
 - E3 - étape 1 projet (pendant les heures du projet): 0.5 p
 - E4 - étape 2 projet (pendant les heures du projet): 1.5 p
 - E5 - projet final (code et documentation, présentés **au prof du cours et du TP**): 2p
 - E6 - points attribués aux activités de FILS liées au MTDL: 3p (e.g. meilleur projet – 0.5 p, participation à la session de communication FILS – 0.5p, prix à la session de communication FILS – 1p, participation à POLIFEST au demo corner 1 jour – 1p, participation à POLIFEST au stand 1 jour – 0.5p, présentation pendant le cours sur un sujet donné – 0.75p, tests inopinés pendant le cours – 0.25)
 - Note maximale: 12p

<https://docs.google.com/spreadsheets/d/189c3wkbPmMyS0JO4pr0R9ythl2iGZbfbUnjtGyb4jMg/edit#gid=1494738508>
- Pour passer:
 - $E \geq 4.5$ (sur 12)
 - $E1 \geq 4.5$ (sur 10)
 - $E3 \geq 2$ (sur 10)
 - $E4 \geq 2$ (sur 10)
 - $E5 \geq 4.5$ (sur 10)
 - présence au moins 7 TP et présence au moins 7 cours (spécialement pour les cours avec invités de l'industrie)

Distribution de ressources en développement logiciel



Objectifs du cours

- Intro: du développement de systèmes au développement de logiciels, activités dans le développement de logiciels – 1 semaine
- Modèles du processus de développement logiciel (e.g. Waterfall, V model, iterative, spiral, Agile) – 1 semaine
- Analyse et définition des besoins (e.g. Business Modeling vs Software Modeling, User Requirements vs Software Requirements) – 1 semaine
- Modélisation de systèmes logiciels (e.g. modèles orientés objet, UML) – 2 semaines
- Conception architecturale (types d'architectures, styles architecturaux – e.g. Client – Server, 3-Tier, Model - View - Controller etc, modèles de conception/ Design Patterns) – 3 semaines
- Tests de programme et assurance qualité – 1 semaine
- Présentations d'étudiants – max 8 (chaque présentation doit durer 15 minutes)
- https://docs.google.com/spreadsheets/d/19qT6R9BRThPE_-vVafokoVrxthZa7EAu3-PSaeEVQYA/edit#gid=0 – 1 semaine
- Révision, préparation pour l'examen – 1 semaine
- Présentations des projet final – 1 semaine
- Présentation de spécialistes de l'industrie

Objectifs du TP

- 1. Révision – Java classes, interfaces, collections
- 2. Interfaces graphiques pour desktop applications en Java – Java Swing, Java FX
- 3. Threads, travaux avec des fichiers en Java
- 4. Travaux avec bases de données en Java
- 5 . Représentations graphiques UML
- 6. Modèles de conception/ design patterns
- 7. Moyennes pour TP et projet

Objectifs du Projet

- développer une application (desktop, mais pas obligatoire) complexe en Java par équipes de 2 étudiants maximum, en fonction d'exigences données
- développer la documentation associée:
 - Identification des acteurs, opérations (fonctionnalités) du système, règles métier (business rules)
 - Diagrammes UML : cas d'utilisation, diagrammes d'activités (au moins 3), diagrammes de séquence (au moins 3), diagramme de classes, diagramme de paquets
 - Description détaillée de chaque cas d'utilisation
 - Description du modèle architecturale (composantes, design patterns – au moins 2, diagramme de classes et – si nécessaire – diagramme de composantes)
 - Diagramme (schema) de la base de données
- présentation de l'application aux clients/profs pendant **la dernière semaine**
- il y a 3 types de projets: A (pour la note maximale 10), B (pour la note maximale 8), C (pour la note maximale 5): plus de détails pendant les heures du projet

References

- Alin Moldoveanu , Florica Moldoveanu, Maria-Iuliana Dascalu et al., **UML Practic**, Editura Matrix Rom (COD CNCIS: 39), ISBN 978-606-25-0118-1. 2014
- L.D. Serbanati, **Méthodes et Techniques du Développement du Logiciel - Notes de cours**, 2015
- Ian Sommerville, **Software Engineering**, Eight Edition, Addison-Wesley, 2007, ISBN 13 : 978-0-321-31379-9
- Roger S. Pressman, **Software Engineering, A Practitioner's Approach**, Fifth Edition, European Adaptation, McGraw-Hill, 2000, ISBN 0-07-709677-0.
- Hans van Vliet, **Software Engineering, Principles and Practice**, Second Edition, Wiley, 2000, ISBN 0-471-97508-7.
- Tom Pender, **UML Bible**, John Wiley & Sons, 2003, ISBN:0764526049
- Grady Booch, James Rumbaugh, Ivar Jacobson, **Unified Modeling Language User Guide**, 2nd edition, , Addison-Wesley Professional, 2005
- Erich Gamma, Richard Helm, Ralph Jonsopn si John Vlissides, **Design Patterns: Elements of Reusable Object-Oriented Software**, Addison-Wesley, 1994.
- Grady Booch, James Rumbaugh, Ivar Jacobson, **Software Testing and Quality Assurance: Theory and Practice**, Grady Booch, James Rumbaugh, Ivar Jacobson, Wiley&Sons, 2008, ISBN: 978-0-471-78911-6.
- C.Mazza, J.Fairclough, B.Melton, D. de Pablo, A.Scheffer, R.Stevens, M.Jones, G.Alvesi, **Software Engineering Guides**, Prentice Hall, 1996, ISBN 0-13-449281-1 (The European Space Agency (ESA) Software Engineering standards).
- <http://www.softreq.com/>
- <http://users.tkk.fi/~mlauerma/contextmupe/requirements.html>
- <http://www.agilealliance.com/>
- <http://www.extremeprogramming.org/>