ENSEIRB-MATMECA

Conception orientée objet

École / Prépa
ENSEIRB-MATMECA

Présentation

Code interne : EI9PG301

Description

Intervenants: Georges Eyrolles, Philippe Narbel
Ce cours présentera les principes généraux de conception des architectures logicielles, et tout particulièrement ceux qui s'appliquent aux architectures orientées-objet. Les cinq principes dits SOLID (responsabilité unique, ouvert/fermé, substitutivité de Liskov, ségrégation des interfaces, inversion des dépendances) y seront définis et développés en détail, que ce soit dans leur mise en oeuvre, leur conséquences, leurs qualités et leurs défauts. Ces principes seront par ailleurs illustrés et mis en relation avec despatrons de conception classiques (Design Patterns), comme par exemplel'Adaptateur, le Composite, le Décorateur, le Médiateur, la Stratégie, le Visiteur. Ces derniers seront donc également introduits et définis.
Même si les notions et les concepts y seront présentés avec toute la rigueur qu'ils méritent, ce cours fera la part belle aux expériences, aux exercices pratiques, et aux études de cas.
BIBLIOGRAPHIE : - Agile Software Development. Robert Martin. Prentice Hall. 2003. - Working Effectively with Legacy Code. Feathers, Prentice Hall. 2005. - Design Patterns. Erich Gamma, Richard Helm, Ralph Johnson et John Vlissides. Addison-Wesley. 1995. (livre appelé aussi "GoF" pour "Gang of Four") - Pattern Hatching: Design Patterns Applied. John Vlissides. Addison-Wesley. 1998. - Object-oriented Software Construction. Bertrand Meyer. Prentice-Hall. 1988, 1997. - Object-Oriented Modeling and Design with UML. Blaha and Rumbaugh. Prentice Hall. 1991, 2005. - Large-Scale C++ Software Design. John Lakos. Addison-Wesley. 1996. - Design Patterns for Object-Oriented Software Development. W. Pree. Addison-Wesley. 1995. - Structured Analysis and System Specification. DeMarco. Prentice Hall. 1979. - Structured Design: Fundamentals of a Discipline of Computer Program and Systems Design. Yourdon and Constantine. 1979. - The Practical Guide to Structured Systems Design. Page-Jones. Prentice Hall. 1988
PLAN :
- Architectures logicielles, composants, dépendances. - Substitutivité. - Couplage et cohésion. - Responsabilités, séparation des rôles. - Ségrégation. - Concept de l'ouvert/fermé.

Heures d'enseignement

CI	Cours Intégrés	26h
TI	Travaux Individuels	10h

Pré-requis obligatoires

- Programmation orientée-objet (PG202/203). - Bonne connaissance du langage Java.

Syllabus

- Architectures logicielles, composants, dépendances. - Substitutivité. - Couplage et cohésion. - Responsabilités, séparation des rôles. - Ségrégation. - Concept de l'ouvert/fermé.

Modalités de contrôle des connaissances

Évaluation initiale / Session principale - Épreuves

Type d'évaluation	Nature de l'épreuve	Durée (en minutes)	Nombre d'épreuves	Coefficient de l'épreuve	Note éliminatoire de l'épreuve	Remarques
Contrôle Terminal	Ecrit	120		1		sans document sans calculatrice

Seconde chance / Session de rattrapage - Épreuves

Type d'évaluation	Nature de l'épreuve	Durée (en minutes)	Nombre d'épreuves	Coefficient de l'épreuve	Note éliminatoire de l'épreuve	Remarques
Epreuve terminale	Ecrit	90		1		sans document

Infos pratiques

Contacts

Georges Eyrolles
Georges.Eyrolles@bordeaux-inp.fr