LICENCE MENTION INFORMATIQUE

PARCOURS CONCEPTION ET DÉVELOPPEMENT D'APPLICATIONS

Semestre 5

Java et Conception d'Applications

Présentation

6 crédits ECTS

UE constituée de deux EC obligatoires : Java 2 et Conception d'Applications.

Java 2

Présentation

4 crédits ECTS

Volume horaire

Travaux Pratiques : 12h
Cours Magistral : 12h
Travaux Dirigés : 12h

Ce cours revient sur les concepts fondamentaux de la programmation objet : l'encapsulation, la composition et l'héritage. En complément, ce cours aborde les concepts de classe enfouie, de classe abstraite, d'interface, d'introspection, de clonage, d'exception, de généricité et de lambda-expression. Le langage Java sert de support à l'apprentissage de ces concepts. Le cours aborde également les éléments indispensables de l'API pour le développement d'applications en Java comme les collections, les flots de données ou les fils d'exécution. Enfin, le cours présente les principaux outils de développement associés à Java : outils de tests unitaires et de non-régression, générateurs de documentation, outils de contrôle de la qualité du code, et outils de travail collaboratif.

L'UE est organisée en 10 leçons comportant chacune un cours, une à deux séances de travaux dirigés et une à deux séances de travaux pratiques :

1. Rappels de Java : fondements impératifs, fondements Objet (encapsulation, composition, héritage)
2. Classes enfouies, classes abstraites, interfaces
3. Introspection, clonage, égalité, représentation littérale
4. Exceptions : principes généraux et mise en oeuvre en Java
5. Généricité : polymorphisme et types génériques
6. Objets fonctionnels : lambda-expressions
7. Collections : API standard, collections simples et tableaux associatifs
8. Flots de données : flots d'octets et de caractères, gestion de fichiers et analyse lexicale
9. Fils d'exécution : principes élémentaires de programmation concurrente
10. Outillage pour le développement d'applications : JUnit, Javadoc, Doxygen, Checkstyle, SVN, Git

Objectifs

L'objectif est quadruple :

- savoir tirer parti de la programmation objet pour la conception et la maintenance de programmes modulaires, réutilisables, évolutifs et sûrs
- compléter les éléments méthodologiques d'analyse, de conception et de programmation orientées objet
- renforcer la maîtrise du langage Java, de ses constructions élémentaires, et de son API standard
- découvrir l'outillage essentiel au développement d'une application d'envergure

- maîtriser le langage Java : sa syntaxe et ses fondements sémantiques
- connaître les différentes phases de la compilation d'un programme Java
- connaître le mécanisme d'exécution d'un programme Java et le fonctionnement de la machine virtuelle
- maîtriser les rudiments de la programmation objet
- connaître quelques bonnes pratiques de conception objet

D'autre part, concernant l'outillage pour la conduite des projets, les étudiants sont amenés à expérimenter et à comprendre l'intérêt et les principes des outils :

- de validation par tests unitaires
- de documentation logicielle
- de contrôle de la qualité du code
- de gestion de versions et travail collaboratif

Pré-requis nécessaires

Pour aborder ce cours, il est nécessaire d'avoir déjà programmé en Java ou dans un langage de programmation Objet similaire, et d'avoir un minimum de connaissances sur :

- les aspects procéduraux de Java (types, variables, opérateurs, appels, boucles, conditionnelles) et l'algorithmique élémentaire (tri, recherche de max, calcul, etc.)
- les fondements de la programmation Objet (encapsulation, composition, héritage)

> les principes élémentaires de conception Objet (identification des classes, assignation de responsabilités, structure, composition et comportement des objets)

L'UE Java 1 de L2 et l'UE Introduction à l'objet et aux systèmes d'exploitation et microprocesseur en L3 fournissent ces pré-requis.

Compétences visées

Traduire la spécification d'un programme en système d'objets en assignant des responsabilités.

Concevoir un programme à la fois ouvert à l'extension et fermé à la modification.

Mobiliser à bon escient les constructions du langage Java et de son API.

Savoir utiliser les outils de tests unitaires, les générateurs de documentation, les outils de contrôle de la qualité du code, et les outils de travail collaboratif.

Bibliographie

> Mickaël Kerboeuf : Fondements de la programmation orientée objet avec Java 8. Références Sciences, Ellipses, décembre 2016, ISBN 9782340014824.

> Mickaël Kerboeuf : Algorithmique et programmation objet. Références Sciences, Ellipses, mai 2020, ISBN 9782340037984.

> James Gosling, Bill Joy, Guy Steele et Gilad Bracha : The Java(TM) Language Specification (3rd Edition). Addison-Wesley Professional, 2005, ISBN 0321246780.

> Erich Gamma, Richard Helm, Ralph Johnson et John Vlissides : Design Patterns, Elements of Reusable Object-oriented Software. Addison-Wesley Longman Publishing Co., 1995, ISBN 0-201-63361-2.

Modalités de contrôle des connaissances

Session 1 ou session unique - Contrôle de connaissances

Nature de l'enseignement	Modalité	Nature	Durée (min.)	Coefficient	Remarques
UE	CT	Ecrit - devoir surveillé	120	2/3	
UE	CC	Travaux Pratiques		1/3	

Session 2 : Contrôle de connaissances

Nature de l'enseignement	Modalité	Nature	Durée (min.)	Coefficient	Remarques
UE	CT	Ecrit - devoir surveillé	120	1	Pas de report de CC, même favorable.

Conception d'applications

Présentation

Cette UE a pour objectif la maîtrise des outils du développeur (codage, test et déploiement) pour réaliser une application déployable et utilisable par un utilisateur final. L'application visée couvre plusieurs technologies (développement objet, développement Web, base de données).

L'UE est organisée sous forme de projet à réaliser en groupe de 4 à 5 étudiants selon une méthode itérative basée sur plusieurs Sprints.

2 crédits ECTS

Volume horaire

Cours Magistral : 2h
Travaux Dirigés : 6h
Travaux Pratiques : 12h

Pré-requis nécessaires

Programmation Java. Programmation Web. Programmation Base de données. Mise en place de tests unitaires.

Compétences visées

Savoir développer une application. Savoir spécifier et mettre en place un IHM. Savoir décrire et mettre en place une base de données. Savoir décrire et documenter le code. Maîtriser la définition et la mise en place de tests unitaires. Maîtriser la qualité du code. Maîtriser les outils du développeur (IDE, suivi de version, vérification de la qualité du code, spécification du code, mise en œuvre de tests unitaires).

Modalités de contrôle des connaissances

Session 1 ou session unique - Contrôle de connaissances

Nature de l'enseignement	Modalité	Nature	Durée (min.)	Coefficient	Remarques
EC	CC	Travaux Pratiques		1	note reportée en session 2

Session 2 : Contrôle de connaissances

Nature de l'enseignement	Modalité	Nature	Durée (min.)	Coefficient	Remarques
EC	Report de notes	Travaux Pratiques		1	report de session 1