

UE Programmation par objets



Niveau d'étude
Bac +3



ECTS
6 crédits



Crédits ECTS
Echange
0.0



Composante
UFR IM2AG
(informatique,
mathématiques
et
mathématiques
appliquées)



Période de
l'année
Automne (sept.
à dec./janv.)

- > **Langue(s) d'enseignement:** Français
- > **Méthodes d'enseignement:** En présence
- > **Forme d'enseignement :** Cours magistral
- > **Ouvert aux étudiants en échange:** Oui
- > **Crédits ECTS Echange:** 0.0
- > **Code d'export Apogée:** GBIE5U06

Présentation

Description

- programmer de façon modulaire, lisible et maintenable dans un langage orientée objet à partir d'un diagramme UML de classes fourni en utilisant les concepts fondamentaux de l'approche orientée objet (classes, instances, encapsulation, héritage, polymorphisme, généricité, abstraction)

- décrire l'état mémoire d'un programme objet (instances/références/données présentes)

- utiliser les exceptions, les classes abstraites et interface dans un programme en JAVA

Programme résumé du cours

- Révision et approfondissement en algorithmique impérative / passage à l'objet (normalisation des notions / utilisation des structures algorithmiques les plus adaptées, travail sur la lisibilité et maintenabilité du code)
- Description des états mémoires d'un programme (quelles sont les instances/références/données présentes ?)
- Encapsulation

- Héritage et polymorphisme
- Classes Abstraites / Interfaces (au sens Java)
- Généricité / Collections
- Exceptions

Les exemples et les TPs sont donnés en Java, mais les concepts vus sont applicables dans tous les langages objet.

Objectifs

L'objectif de ce cours est l'introduction des paradigmes de programmation objet dans le cadre du génie logiciel.

Heures d'enseignement

CMTD	Cours magistral - Travaux dirigés	16,5h
TP	TP	33h

Pré-requis recommandés

Algorithmique et programmation dans un langage classique, Outils formels.

Période : Semestre 6

Compétences visées

Les compétences visées sont la programmation modulaire, lisible et maintenable dans un langage orientée objet à partir d'un diagramme UML de classes fourni (la conception ne fait pas partie des attendus), en maîtrisant les concepts fondamentaux de l'approche orientée objet : classes, instances, encapsulation, héritage, polymorphisme, généricité, abstraction.

Bibliographie

- B.Meyer, Object Oriented Software Construction, Second Edition. Prentice Hall, 1997
- G.Masini, A.Napoli, D.Colnet, D.Léonard, K.Tombre, Les Langages à objets, Interéditions 1989

Infos pratiques

Contacts

Responsable pédagogique

Celine Fouard

✉ celine.fouard@univ-grenoble-alpes.fr

Lieu(x) ville

> Grenoble

> Valence

Campus

> Grenoble - Domaine universitaire

> Valence - Briffaut