

# Programmation orientée objet et interfaces homme-machine 1

Niveau d'étude  
**Bac +3**

ECTS  
**6 crédits**

Composante  
**Sciences Fondamentales  
et Appliquées**

Période de l'année  
**Semestre 5**

## En bref

- # **Langue(s) d'enseignement:** Français
- # **Méthode d'enseignement:** En présence
- # **Ouvert aux étudiants en échange:** Oui

## Présentation

## Description

### Programme résumé

- \* Éléments historiques de la programmation orientée objet
- \* Classes et objets : abstraction, encapsulation, constructeurs, visibilité
- \* Associations de classes : agrégation, composition
- \* Héritage et ses implications : redéfinition et surcharge de méthodes, contrôle de l'héritage, polymorphisme, sur-classement et sous-classement, classes abstraites et interfaces
- \* Étude du langage Java et ses spécificités (utilisation d'un environnement de développement intégré, création et utilisation de bibliothèques, manipulation de collections, création et gestion des exceptions, manipulation de flux

pour les entrées/sorties, sérialisation, duplication d'objets (copie de surface et copie profonde)

- \* UML : diagrammes de classes, diagrammes de séquences et diagrammes d'états

Réalisation d'un projet à effectuer par petits groupes, où les étudiants doivent concevoir, spécifier, réaliser, tester et présenter leur application.

## Objectifs

Connaître les fondements du paradigme de programmation orientée objet

## Heures d'enseignement

Programmation Orientée Objet et IHM - CM	CM	20h
Programmation Orientée Objet et IHM - TD	TD	12h
Programmation Orientée Objet et IHM - TP	TP	12h
Programmation Orientée Objet et IHM - P-Proj	Pédagogie par projet	6h

## Pré-requis nécessaires

Niveau d'algorithmique et de programmation des 2 premières années de licence

## Compétences visées

- \* Maîtriser l'environnement de travail et de développement
- \* Choisir les structures de données adéquates pour un problème donné
- \* Proposer un algorithme adapté à un problème donné
- \* Développer dans différents paradigmes de programmation (objet, fonctionnel, impératif, événementiel...)
- \* Développer un logiciel (codage, vérification et documentation) dans un contexte scientifique et technique maîtrisé
- \* Utiliser des techniques de compilation séparée ou des bibliothèques pour augmenter la réutilisabilité du développement

---

## Infos pratiques

### Lieu(x)

# Futuroscope