

## Support de cours

---

# *PROGRAMMATION ORIENTEE OBJETS*

---

**Niveau :** Troisième niveau de la section informatique

**Option :** Informatique Industrielle

Réalisé par : SGHAIER Imene

Année Universitaire : 2007-2008

# Table des Matières

INTRODUCTION A LA POO .....	1
0.1 Rappel : Les Types Abstraites de données.....	1
0.2 Exemple : concept POO .....	2
CONCEPTS DE BASE DE LA POO .....	4
1.1 De la programmation classique vers la programmation orientée objet .....	4
1.2 Définitions .....	5
LES CONCEPTS ORIENTES OBJETS EN JAVA.....	10
2.1 Notion de Classe.....	10
2.2 Attributs et Méthodes .....	12
2.3 Objets et Instanciation.....	17
2.4 Membres statiques et Autres mots clés (final, this, null) .....	22
2.6 Les packages .....	26
3.1 L'encapsulation .....	29
3.2 Modificateurs de visibilité et Accès .....	29
3.3 Surcharge des Méthodes.....	33
L'HERITAGE .....	38
4.1 Le principe de l'héritage.....	38
4.3 L'accès aux propriétés héritées .....	40
4.4 Construction et initialisation des objets dérivés.....	42
4.5 Dérivations successives .....	44
4.6 Redéfinition et surcharge de membres .....	44
4.7 Autres mots clés «final » et « super ».....	47
4.8 Des conseils sur l'héritage .....	48
LE POLYMORPHISME.....	49
5.1 Concept de polymorphisme.....	49
5.2 Exemples et interprétations .....	49
5.3 Conversion des arguments effectifs .....	52
5.4 Les conversions explicites de références .....	53
CLASSES ABSTRAITES ET INTERFACES .....	57
6.1 Les classes abstraites .....	57
6.2 Les interfaces .....	60

# Présentation du cours

Ce cours est conçu comme une introduction aux paradigmes de la programmation orientée objet. Il présente les notions de base de cette technologie : type abstrait de données, classe, objet, héritage simple et multiple, objet complexe. Mécanisme d'abstraction, surcharge, généricité, polymorphisme. Les concepts seront illustrés avec le langage JAVA.

## **Niveaux cibles :**

Étudiants du niveau 3, toutes filières (Réseaux Informatique, Informatique de gestion, informatique industrielle)

## **Pré-requis :**

Algorithmique, Structures de données, Programmation I et Programmation II

## **Formule pédagogique :**

- Exposé informel
- Laboratoire

## **Moyens pédagogiques :**

- Tableau
- Support de cours

## **Méthodologie :**

- Cours intégré
- Travaux dirigés (réalisation et correction d'exercices)
- Travaux pratiques (JDK 1.5 + console MS-DOS)

## **Volume Horaire :**

- 22.5 heures de cours
- 45 heures de travaux pratiques

## **Objectifs généraux :**

- Comprendre les origines et l'intérêt d'un nouveau style de programmation orientée objet
- Maîtriser les concepts orientés objet en termes de définitions, syntaxe Java et usage
- Savoir utiliser cette nouvelle approche pour modéliser des problèmes
- Faire la liaison entre les différents éléments du cours (classes, objets, encapsulation, héritage, polymorphisme, classe abstraite et les interfaces)
- Traiter la gestion des exceptions afin de permettre une production du code efficace

Objectifs généraux	Conditions de réalisation de la performance	Critères d'évaluation de la performance
1- S'initier aux concepts orientés objets et leurs intérêts pratiques.	A partir des notes du cours, des travaux dirigés et des références bibliographiques, l'étudiant doit être capable de dégager les lacunes de l'approche classique de programmation et de distinguer les principales caractéristiques de POO	Aucune confusion entre les principes de la programmation classique et la programmation objet
2- Comprendre les concepts orientés objet en termes de concept, syntaxe Java et utilisation.	A partir des notes du cours, des travaux dirigés et des références bibliographiques, l'étudiant doit être capable de concevoir des classes et des objets en Java	Les exercices relatifs à l'implémentation d'une classe en Java doivent être réussis
3- Comprendre l'utilité et l'application des concepts d'encapsulation et de surcharge.	A partir des notes du cours, des travaux dirigés et des références bibliographiques, l'étudiant doit maîtriser les concepts d'encapsulation et surcharge	Les exercices relatifs à l'accès aux membres privés doivent être maîtrisés
4- Comprendre la technique d'héritage et ses intérêts pratiques	A partir des notes du cours, des travaux dirigés et des références bibliographiques, l'étudiant doit être capable de concevoir et implémenter des classes dérivées.	Aucune erreur n'est permise au niveau de la définition et la manipulation d'une classe dérivée
5- Comprendre le concept de polymorphisme et ses intérêts pratiques	A partir des notes du cours, des travaux dirigés et des références bibliographiques, l'étudiant doit comprendre l'intérêt pratique de polymorphisme et la relation avec l'héritage	Aucune erreur n'est permise au niveau de la liaison entre l'héritage et le polymorphisme
6- Comprendre et distinguer entre les 2 concepts de classes abstraites et interfaces	A partir des notes du cours, des travaux dirigés et des références bibliographiques, l'étudiant doit comprendre l'utilité pratique des classes abstraites et interfaces et distinguer les nuances entre ces 2 concepts	Aucune confusion entre les 2 concepts n'est permise

7- Faire découvrir la notion de gestion des exceptions	A partir des notes du cours, des travaux dirigés, l'étudiant doit comprendre l'utilité pratique des exceptions.	Les exercices relatifs à l'implémentation des mécanismes des exceptions en Java doivent être réussis
--	---	--

**Évaluation** : 1 test, 1 DS et un examen final écrits.

Nature évaluation	Contenu	Pourcentage	Date (environ)
<b>Test</b>	Semaine 1 à 4 ou Semaine 8 à 15	10%	Semaine 1 à 4 ou Semaine 8 à 15
<b>Devoir surveillé</b>	Semaine 1 à 8	30%	Semaine 8
<b>Note TP</b>	Semaine 2 à 14	30%	Semaine 15
<b>Examen</b>	Semaine 1 à 15	40%	Semaine 16

**Bibliographie** :

- <http://java.sun.com>
- Lemay L, Le Programmeur JAVA 2, Campus Press.
- Bailly C, Chaline J.F., Ferry H.C & al, Les langages orientés objets, Cepadues éditions.