

Conception et programmation orientées objet

Audition au poste de Maîtresse de Conférence
Université Marie et Louis Pasteur

Lison Blondeau-Patissier

ATER à l'École Normale Supérieure de Lyon (LIP)

28/05/2026

Conception et Programmation Orientées Objet

- **Niveau** : Licence, Semestre 3
- **Heures** : 18h CM, 18h TP/TD
- **Langages** : Java, UML
- **Prérequis** : Programmation 1 (Java)
- **Évaluation** : 1 TP noté + 1 partiel de mi-semestre + 1 mini projet de programmation
- **Contenu** : Introduction à la POO et aux diagrammes UML
↪ classes, objets, héritage, polymorphisme, diagrammes UML, documentation, javadoc
- **Compétences** :
 - * Bases POO + UML
 - * Apprendre à réfléchir à la **structure des programmes et du code**

Organisation du cours

Semaine 1	Semaine 2	Semaine 3	Semaine 4
 Intro POO	 Intro UML	 Héritage	 Polymorphisme
TD/TP. Lecture et écriture de codes simples	TD. Lecture et écriture de diagrammes simples	TP. Manipulation de code avec héritage	TP. Manipulation de code avec polymorphisme
Semaine 5	Semaine 7	Semaine 8	Semaine 9
 Interfaces	 Analyse	 Packages	 Documentation
TP. Implémentation et utilisation d'interfaces	TD. Lecture et écriture de diagrammes avancés	TP. Implémentation de packages	TD/TP. Lecture et écriture de javadoc

Partiel sur table et TP noté la semaine 6 + Projet de fin de semestre

Cours 1 : Introduction à la POO

1 – Introduction *30 minutes*

- Introduction historique
- Principes de bases
- Présentation du cours et du fonctionnement sur l'année

2 – Premières définitions *1h30*

- Objets, classes, instanciations
- `private` / `public`
- `getter` et `setter`

TD/TP 1 : Bases de POO en Java

1 – Partie TD *40 minutes*

- À partir de codes exemples, prédire l'affichage et expliquer les résultats

➔ **Objectif :** *Compréhension globale du langage en autonomie*

2 – Partie TP *1h20*

- Petits exercices de programmation

 *Exemple : Catalogue d'une bibliothèque*

➔ **Objectif :** *Mettre en pratique les notions vues en cours*

Cours 2 : Introduction UML

1 – Introduction *30 minutes*

- Introduction historique
- Principes de bases, avantages
- Les différents types de diagrammes

2 – Diagrammes d'objets *30 minutes*

- Définitions + exemples

3 – Diagrammes de classes *1h*

- Définitions + exemples

TD 2 : Introduction UML

1 – Compréhension *40 minutes*

- Explication de diagrammes UML

 *Exemple : Distributeur de boissons*

➔ **Objectif :** *Comprendre des diagrammes simples*

2 – Mise en oeuvre *1h20*

- Dessins de diagrammes

➔ **Objectif :** *Savoir dessiner des diagrammes d'objets et de classes*

Cours 3 : Héritage

1 – Définitions générales 40 minutes

- Intérêt : code réutilisable
- Définitions générales
- Représentation UML
- *Un mot sur l'héritage multiple*

2 – Héritage en Java 1h20

- Classes mère / fille en Java
- protected
- super()
- final

TP 3 : Héritage

1 – Diagrammes UML 40 minutes

- Écrire des classes à partir de diagrammes UML

➔ **Objectif :** *Comprendre des diagrammes représentant l'héritage*

2 – Mise en oeuvre 1h20

- Écrire et manipuler des classes pour divers types de données

 *Exemple : Classer des animaux*

➔ **Objectif :** *Manipulation des notions du cours*

Cours 4 : Polymorphisme

1 – Statique

1h

- Définitions
- Surchage de méthode
- Exemples **Java** et UML
↪ *inclure des exemples d'erreurs*

2 – Dynamique

1h

- Définitions
- Exemples **Java** et UML
↪ *inclure des exemples d'erreurs*

TP 4 : Polymorphisme

- Étendre les exercices de la semaine précédentes au polymorphisme

 *Exemple : Classer des animaux*

 **Objectif :** *Comprendre le polymorphisme en **Java***

Cours 5 : Interfaces

1 – Classes abstraites *40 minutes*

- Définition
- Méthodes abstraites
- Exemples

2 – Interfaces *1h20*

- Définitions **Java**
- Diagrammes UML
- Exemples

TP 5 : Interfaces

- Implémentation d'interfaces
- Utilisation d'interfaces

 *Exemple : Activités dans une salle de sport*

 **Objectif : Savoir écrire et manipuler des interfaces en Java**

 **Fin de la semaine : deadline pour le choix des projets / des groupes**

Semaine 6 : 2h CM (partiel) + 2h TP (noté)

Cours 6 : Partiel

Devoir sur table, pas d'ordinateur.

Programme : Cours 1 à 5.

Contenu :

- Codes à commenter
- Écriture de mini programmes
- Erreurs à trouver dans un code
- Diagrammes UML à expliquer
- Diagrammes UML à dessiner

TP 6 : TP noté

Individuel.

Programme : Cours 1 à 5.

Contenu : Exercices de programmation en **Java** sur les notions des cours 1 à 5.

➔ **Objectif** : *Évaluer les acquis de mi-semester*

Cours 7 : Analyse de problèmes

1 – Diagrammes d'interaction *1h*

- Définitions
- Séquence / Communication
- Exemples

2 – Cas d'utilisation *1h*

- Définitions
- Exemples

TD 7 : Analyse de problèmes

- Compréhension de diagrammes
- Écriture de diagrammes

 *Exemple : Gestion des stocks dans un restaurant*

➔ **Objectif** : Analyser un problème

➔ **Objectif** : Anticiper la structure du code avant de commencer à coder

Cours 8 : Packages

1 – Packages en UML

1h

- Définitions
- Exemples

2 – Packages en Java

1h

- Utiliser des packages existants
- Écrire des packages
- Exemples de code
↪ étendre les exemples précédents

TP 8 : Packages

- Implémentation de packages à partir de consignes écrites
- Implémentation à partir de diagrammes UML

 *Exemple : Calculatrice*

 **Objectif :** Manipuler et écrire des packages en Java

 **Objectif :** Savoir lire des diagrammes UML

Cours 9 : Documentation

- Programmation par contrat
- Javadoc
- Exemples concrets
↪ *code mal commenté à corriger*
- Bonnes pratiques de code en général...

TD/TP 9 : Documentation

1 – Partie TD 40 minutes

- Lecture de javadoc
- Écriture de javadoc
↪ *reprendre les TP précédents*

➔ **Objectif :** *Comprendre comment lire et écrire la javadoc*

2 – TP : Mise en oeuvre 1h20

- Écriture de programmes documentés

➔ **Objectif :** *Écrire un code correctement commenté*

Évaluation : mini-projet de programmation

Mini-projet

📅 À rendre en début de semaine 10

- Un rendu par groupe de deux
 - * Code (en **Java** !)
 - * Rapport de quelques pages avec une étude du problème *via* des diagrammes UML
- Sujet libre mais soumis à validation

 *Exemple : mini-jeu, classification des espèces, catalogue d'une médiathèque ; ...*

Grille d'évaluation

- Présentation et analyse du problème
- Répartition du travail
- Rédaction du rapport
- Structure du code
- Qualité du code
- Propreté du code
- Documentation

➔ **Objectifs** : *Travailler en binôme pour répondre à une demande avec un projet bien structuré, cohérent et justifié*