

# Programmation Orientée Objets avec Java

## Projet POO

**Stéphane Malandain / Yassin Rekik**  
**Semestre d'automne 2024**

### Introduction

Dans le cadre du cours de Poo / Java, vous allez réaliser un projet entier. Ce travail se divise en 4 étapes. Chaque étape fait l'objet d'un rendu à une date fixe, avec une note. La moyenne des 4 notes vous donne une note de projet.

Le projet est individuel. Plagiat ou copie seront sévèrement puni ! Les modalités de dépôt vous seront communiquées sur le Git ultérieurement.

### **1 Première étape : les bases d'un gestionnaire de tâches : Terminé !**

Ce projet consiste en la création d'un gestionnaire de tâches. Par définition, une tâche est un travail qui doit être effectué dans un temps donné. Les tâches peuvent être donc très différentes et très variées, réalisées par une, 2 ou plusieurs personnes.

### **2 Seconde étape : Terminé !**

### **3 Troisième étape : Programmation fonctionnelle, tests unitaires, exceptions et interface graphique.**

Le but de cette dernière étape est de finaliser une version de ce projet. Il s'agit donc de consolider votre code en effectuant les tests unitaires et en gérant les exceptions, de sauvegarder les tâches localement, et de réaliser une interface graphique appropriée.

#### Partie A : Programmation fonctionnelle

Le but de cette partie est de profiter des streams Java pour implémenter des fonctionnalités supplémentaires sur vos collections de tâches. Par exemple, sélectionner les tâches entre deux dates, sélectionner les tâches ayant un statut donné, sélectionner les tâches dont la description contient certains mots clefs. L'idée pour réaliser ceci est d'implémenter quelques interfaces fonctionnelles (au moins 1 de chaque type : un prédicat, un consumer, un supplier, et une fonction) et de les utiliser pour ajouter 4-5 fonctionnalités sur les listes de tâches, en utilisant des streams.

## Partie B : Tests unitaires et gestion des exceptions

La deuxième tâche de cette partie est d'améliorer la qualité de votre code. Pour cela, vous devez effectuer deux choses. D'abord, vous devez gérer les exceptions quand ceci s'impose. Il s'agit surtout de gérer les exceptions lors de la manipulation des fichiers, lors de vérification des choix des utilisateurs, etc.

Ensuite, vous devez implémenter des tests unitaires. Normalement, les tests unitaires auraient dû être implémentés dès le départ. Mais comme nous avons abordé ce thème récemment, nous vous demandons juste de sélectionner une de vos classes essentielles, et de lui implémenter une classe de tests unitaires permettant une bonne couverture des cas pertinents.

## Partie C : La persistance des données

La troisième tâche de cette étape est de gérer la persistance des données. Il s'agit de donner la possibilité de sauvegarder / restaurer les tâches au début et à la fin de chaque utilisation. Le choix du support de stockage (fichier txt ou csv) est libre. Un exemple de manipulation de fichier CSV est sur le [Git](#).

## Partie D : Interface graphique

La dernière tâche du projet consiste à réaliser une interface graphique avec JavaFX qui permet de rendre la gestion des différentes tâches plus facile. Le but n'est pas non plus d'avoir des interfaces complètes et sophistiquées. Nous vous demandons au minimum une interface graphique permettant à l'utilisateur de créer une nouvelle tâche et de visualiser la liste des tâches. Comme bonus, vous pouvez ajouter les autres opérations pour une tâche : suppression, consultation et modification.