

# PROG2 — Gestion de Dépenses

---

Durée : 1h30 — Rattrapage pratique, Internet autorisé, IA interdite, Lombok interdit.

## 1 Objectif :

Modéliser un système de gestion de dépenses avec héritage, puis manipuler les données via les API Stream de Java.

## 2 Contexte :

On souhaite modéliser des dépenses dans un système de gestion budgétaire.

Chaque dépense a un libellé, un montant et une date.

Certaines sont remboursables (par exemple les frais professionnels), d'autres sont récurrentes (abonnements mensuels ou annuels).

## 3 Travail demandé :

### ✓ Partie 1 — Classe de base Expense (8 points)

1. Implémentez la classe Expense avec :

- String label
- double amount
- LocalDate date

N'oubliez pas les constructeur(s) et les getters, qu'il faut.

2. Méthodes :

- boolean isLargeExpense() → retourne vrai si le montant > 100.0.
- String toString() → affiche proprement la dépense.

### ✓ Partie 2 — Héritage (8 points)

1. Créez RefundableExpense dans les règles de l'encapsulation avec :

- Attribut boolean refunded.
- Méthode refund() qui passe *refunded* à true.

2. Créez RecurringExpense :

- Attribut recurrence qui peut prendre les valeurs "monthly" ou "yearly" uniquement.
- Redéfinir toString() pour inclure la récurrence.

### ✓ Partie 3 — Manipulations avec Streams (8 points)

Créer une méthode dans une classe ExpenseDAO qui fait les opérations suivantes

1. Retourner uniquement les dépenses remboursables non remboursées.
2. Retourner le total de toutes les dépenses récurrentes.
3. Retourner une liste des libellés (label) des dépenses strictement supérieures à 100.
4. Retourner le montant total de toutes les dépenses (les remboursables, et les dépenses récurrentes, sans notion de durée ou de date).

### ✓ Partie 4 — Test (6 points)

1. Dans des tests unitaires JUnit: Incluez au moins une RefundableExpense non remboursée, une RecurringExpense, et une grosse dépense.
2. Proposez au moins un test unitaire pour chaque méthode montrée dans la partie 3.



#### Conseils :

- Pour la date : utilisez `LocalDate.now()` ou [LocalDate.of\(\)](#) si besoin.