

# Introduction au cours PROJET1

En trois étapes :

- Séance de préparation : Ryan (2 x 2h = 4h)
- Séance de renforcement : Lou (5 x 2h = 10h)
- Travail personnel et évaluation : Lou - Ryan

## Objectifs principaux du cours :

- Rappel et remise à niveau sur la P.O.O avec Java;
- Approfondir la compréhension de la P.O.O avec Java :  
<https://www.amazon.com/Effective-Java-Joshua-Bloch/dp/0134685997>
- Faire une étude de cas réel.

=> Si vous ne passez pas cette UE, vos notes PROG3 et PROG4 vont être **AJOURNÉE**, et vous devriez aller en rattrapage même si vous avez une moyenne générale > 10 pour le semestre S3.

## Partie 1 : rappel et remise à niveau sur la P.O.O avec Java

Pré-requis : lire <https://www.jmdoudoux.fr/java/dej/chap-poo.htm>

**Mini-projet sur le thème :** Gestion des stations service d'une compagnie pétrolière

### Contexte sur le thème :

La compagnie pétrolière X prévoit de mettre en place un système d'information dans ces stations. Le système à mettre en place a pour objectif de faire le suivi des entrées-sorties des produits liés aux produits pétroliers (gasoil, essence, pétrole). Voici les informations à savoir :

1. Chaque station possède son propre système, et il existe un système central géré par une station centrale pour pouvoir centraliser les données de chaque station. Autrement dit, on peut **lire** les données de vente de toutes les stations depuis le central, et chaque station peut **mettre à jour** les ventes et l'approvisionnement de celle-ci.  
Par exemple, voici une des vues attendues par la centrale pour suivre les ventes dans un intervalle de date donnée, c'est à dire par exemple entre le 01/05/2024 à 31/05/2024.

ID Station	Qté vendue essence	Montant essence	Qté vendue gasoil	Montant gasoil	Qté vendue pétrole	Montant vendue pétrole	Montant total de la station	Qté restante essence	Qté restante gasoil	Qté restante pétrole
1	10	59.000	10	49.000	10	21.300	129.300	2.500	4.000	1.000
2	0	0	20	98.000	5	2130	10.650	4.500	2.000	800

**Tableau 1 : Lecture par la station centrale des données de vente et stock effectuée par chaque station**

**NB:** prix unitaire ou p.u essence = 5.900 Ar - p.u du gasoil est de 4900 - p.u pétrole : 2130

2. Chaque produit a un prix qui peut varier en fonction du temps.  
Par exemple, le litre d'essence est actuellement à 5.900 Ar, mais dans 1 an, il se peut que le prix passe à 6.900 Ar, or notre outil doit toujours être utilisable sans intervenir dans le code pour modifier la hausse du prix.
3. Aussi, il faut prendre en compte un paramètre important, l'évaporation. Lorsqu'on stocke du carburant au bout d'un certain temps, celui-ci se transforme en gaz et s'évapore. De ce fait, le stock diminue petit à petit en fonction du temps.
  - Ce taux d'évaporation, dans notre cas, est exprimé en litre (l) par jour (j).
  - Chaque produit possède un taux d'évaporation différent, par exemple,
    - Le taux d'évaporation de l'essence est de 100 l/j
    - Le taux d'évaporation du gasoil est de 50 l/j
    - Le taux d'évaporation du pétrole est de 10 l/jCes données peuvent être différentes pour chaque station, car les facteurs qui sont à l'origine de cette évaporation sont liés aux conditions d'entreposition et de conservation de chaque station.

### Fonctionnalités attendues :

#### EPIC 1 : Approvisionnement et vente de produits pétroliers pour chaque station

##### - F1 : Approvisionnement

Lorsqu'un camion citerne approvisionne la station, les responsables doivent mettre à jour le stock en ajoutant à la date d'approvisionnement et la quantité approvisionnée pour chaque produit.

##### - F2 : Vente de produits pétroliers

Lorsqu'un client veut acheter un produit pétrolier, que ce soit de l'essence, du gasoil ou du pétrole, il peut soit acheter en litres ou soit à travers le montant de son carburant.  
Par exemple, le client peut demander au pompiste l'achat de 10L d'essence, ce qui fait un montant à payer de 59.000 Ar, ou dire au pompiste qu'il veut faire acheter 50.000 Ar d'essence, soit 8.47 litres.

Notez bien que durant la vente :

- Il faut que le logiciel vérifie que le stock restant est encore suffisant;
  - Il faut que le logiciel vérifie que la quantité demandée n'est pas supérieure à la quantité maximale supportée par les pompes (200 litres par vente uniquement). C'est une contrainte qui ne peut pas être changée, mais qui va rester dans le temps.
  - Le stock restant doit être réduit de la quantité vendue;
- ##### - F3 : Gestion du stock
- Journalièrement, chaque station peut consulter les mouvements de stock, et le stock restant de tous les produits pétroliers qu'elle vend, en comprenant le taux d'évaporation en vigueur pour la station. Pour faciliter le suivi, on peut spécifier une plage de date, par exemple du 06 mai au 13 mai 2024 pour lire les données sur le tableau 3 ci-dessous.

- a) Ci-dessous un format des données attendues pour connaître le détail de chaque mouvement pour chaque produit. Notez bien que les quantités initiales pour chaque produit sont de zéro (0). Les premiers approvisionnements déterminent ainsi les quantités réelles.

Date et heure	Produit	Type de mouvement	Qté
01/05/2024 08:00	Essence	Entrée	1.000
01/05/2024 08:00	Gasoil	Entrée	2.000
01/05/2024 08:00	Pétrole	Entrée	500
01/05/2024 08:15	Essence	Sortie	50
01/05/2024 08:17	Gasoil	Sortie	10
02/05/2024 08:01	Pétrole	Sortie	90

**Tableau 2 : Lecture par une station des détails de chaque mouvement de stock**

- b) Étant donné que la dernière vue est trop détaillée et pas assez synthétique pour pouvoir suivre les quantités restantes, la station a besoin d'une vue plus synthétique par jour des ventes effectuées et des quantités restantes par produit.

Date	Qté ajoutée Essence	Qté ajoutée Gasoil	Qté ajoutée Pétrole	Qté vendue Essence	Qté vendue Gasoil	Qté vendue Pétrole	Qté restante Essence	Qté restante Gasoil	Qté restante Pétrole
01/05/2024	0	0	0	50	10	0	950*	1990	500
02/05/2024	0	0	0	0	0	90	750**	1940**	390***

**Tableau 3 : Lecture par une station des données de vente et stock restant pour une intervalle de temps**

\*La quantité initiale d'essence était de 1.000, soustrait des 50 litres de vente, il reste 950. Or la date d'approvisionnement était le 01/05/2024, donc nous supposons qu'aucun produit ne s'est évaporé et que la quantité restante initiale correspond directement aux quantités initiales - quantité vendues.

\*\*À partir du 02/05/2024, le lendemain de l'approvisionnement, l'évaporation s'est effectuée. De ce fait, même sans vente d'essence, la quantité a été réduite de 100 litres car le taux d'évaporation journalier pour cette station est de 100 litres par jour. Pareil pour le gasoil, qui a été déduit de 50 litres même si aucune vente n'a été effectuée.

\*\*\*Pour le cas du pétrole, la vente de 90 litres du 02/05/2024 a été additionnée du taux d'évaporation de 10 litres de la veille, ce qui a causé la quantité restante du pétrole à 390.

## EPIC 2 : Vue globale des données de ventes et stock de toutes les stations

Fonctionnalité unique : Obtenir les données de vente et de stock de toutes les stations, telles qu'indiquées au tableau 1.

ID Station	Qté vendue essence	Montant essence	Qté vendue gasoil	Montant gasoil	Qté vendue pétrole	Montant vendue pétrole	Montant total de la station	Qté restante essence	Qté restante gasoil	Qté restante pétrole
1	10	59.000	10	49.000	10	21.300	129.300	2.500	4.000	1.000
2	0	0	20	98.000	5	2130	10.650	4.500	2.000	800

### Rappels des consignes techniques :

- Langage : Java, P.O.O
- Framework : Spring pour rendre les services développées disponibles à travers une API
- Pas de modules de Spring, telles que JPA ou autres.  
**Seul** Lombok est autorisé.
- N'oubliez pas d'ajouter également la librairie JUnit dans votre projet pour pouvoir écrire des tests.