

# Atelier 8 : travail perso (facultatif)

Une série d'exercice pour continuer de pratiquer d'ici à Java avancé.

## 1. Jeu de l'oie

Les règles du jeu de l'oie sont disponibles sur le site suivant : <https://www.regles-de-jeux.com/regle-du-jeu-de-l-oie/>

Après avoir réfléchi aux classes nécessaires pour représenter un jeu de l'oie, développez le programme permettant d'y jouer dans la console.

Vous pouvez ajouter des variantes supplémentaires sur certaines cases, en plus des règles existantes (ex : relancez votre dé, faites reculer tous les autres de 1...)

## 2. Yahtzee

Les règles du Yams sont disponibles sur le site suivant : <https://www.regles-de-jeux.com/regle-du-yams/>

Après avoir réfléchi aux classes nécessaires pour représenter un jeu de Yams, développez le programme permettant d'y jouer dans la console.

Afin de garder un historique de vos parties, enregistrez les scores dans un fichier pour permettre aux joueurs de le consulter même après plusieurs parties.

## 3. MonPetitERP (Exercice SQL)

Un ERP est un logiciel permettant de gérer les ressources d'une entreprise. Il est souvent utilisé pour gérer les stocks, saisir les achats et les ventes, et déclenche des alertes lorsque les stocks sont trop bas. On y gère également les données des clients et des fournisseurs, et toutes ces informations permettent de gérer les facturations.

Nous allons nous intéresser à la mise en place des différentes tables d'un ERP, que l'on pourrait utiliser par la suite dans une application Java pour réellement mettre en place notre ERP personnel.

Pour cela, créez les tables suivantes, avec les types qui vous semblent judicieux :

- articles
  - id
  - nom
  - id\_categorie
  - prix\_unitaire
  - description
- categorie
  - id
  - libelle
- clients
  - id
  - nom
  - id\_adresse
- fournisseurs
  - id
  - nom
  - id\_adresse
  - id\_coordonnes\_bancaires
- coordonnees\_bancaires
  - id
  - iban
  - bic
- adresses
  - id
  - numero
  - type\_rue
  - nom\_rue
  - ville
  - code\_postal
- stocks
  - id
  - id\_article
  - quantite
- factures\_achat
  - id
  - id\_fournisseur
  - date\_facture
- ligne\_achat
  - id
  - id\_article
  - id\_facture\_achat
  - quantite
- factures\_ventes
  - id
  - id\_client
  - date\_facture
- ligne\_vente
  - id
  - id\_article
  - id\_facture\_vente
  - quantite

Pensez à mettre en place les contraintes d'intégrité de vos table (foreign key), pour vous assurer que je ne peux pas affecter une facture à un client inexistant par exemple.

Insérez des données dans chaque table. Essayez de prévoir au moins un cas de figure pour chacun des requêtes demandées ci-après.

Proposez ensuite des requêtes vous permettant de récupérer les informations suivantes (une requête par ligne) :

- La liste de tous les articles dont le nom commence par un V (si vous n'avez pas d'article qui commence par un V, vous avez le droit de chercher un article qui commence par une autre lettre, je vous pardonnerai)
- La liste de tous les clients, avec leurs informations personnelles (nom, adresse)

- La liste de tous les fournisseurs, avec leurs informations personnelles (nom, adresse, coordonnées bancaires)
- La liste de toutes les factures fournisseurs
- La liste de toutes les factures client dont la somme totale dépasse 2000€ (on souhaite connaître le détail des articles commandés et en quelle quantité)
- Le résultat du mois de Mars 2022 (résultat = le montant total des factures clients – le montant total des factures fournisseurs)