

Niveau : B2_IT

Enseignant : M. AJAVON

TRAVAUX PRATIQUES DE POO JAVA

1. Présentation générale

1.1. Contexte

Une petite structure de restauration (restaurant, snack, fast-food) souhaite disposer d'une application de bureau pour gérer :

- son stock de produits (boissons, plats, ingrédients, etc.) ;
- les commandes des clients ;
- les entrées et sorties ;
- quelques statistiques de base (chiffre d'affaires, produits les plus vendus).

L'application sera développée en Java SE avec une interface graphique en Java Swing.

1.2. Objectifs pédagogiques

Ce projet doit permettre aux étudiants de :

- Mettre en pratique la programmation orientée objet en Java (classes, héritage, encapsulation, etc.).
- Appliquer le modèle MVC dans une application graphique.
- Gérer la persistance de données (base de données).
- Manipuler des interfaces graphiques (fenêtres, tableaux, formulaires, événements).
- Gérer correctement les exceptions et la validation des saisies.

1.3. Périmètre du projet

Le projet couvre :

- Gestion des produits et catégories.
- Gestion des stocks (entrées, sorties, alertes).
- Gestion des commandes clients.
- Consultation de quelques états / statistiques simples.

2. Spécifications fonctionnelles

2.1. Gestion des produits et catégories

Description : Permet de définir les produits vendus dans le restaurant.

Données principales :

- Catégorie : id, libellé (ex. "Boissons", "Plats", "Desserts").
- Produit : id, nom, catégorie, prix de vente, stock actuel, seuil d'alerte.

Fonctionnalités :

1. Ajouter une catégorie (saisir libellé, vérifier qu'elle n'existe pas déjà).
2. Modifier / supprimer une catégorie (gérer le cas des produits rattachés).
3. Ajouter un produit (catégorie, prix, stock initial, seuil d'alerte).
4. Modifier / supprimer un produit (gérer les dépendances avec les commandes).

Règles de gestion :

- Le libellé de catégorie doit être unique.
- Le prix de vente doit être strictement positif.
- Le stock ne peut pas être négatif.

2.2. Gestion du stock

Description : Permet d'enregistrer les entrées et sorties de stock (autres que les ventes).

Données :

- Mouvement de stock : id, type (ENTRÉE/SORTIE), produit, quantité, date, motif (achat, perte, inventaire, etc.).

Fonctionnalités :

5. Enregistrer une entrée de stock.
6. Enregistrer une sortie de stock.
7. Consulter l'historique des mouvements.
8. Lister les produits avec stock inférieur au seuil d'alerte.

Règles de gestion :

- Quantité de mouvement > 0.
- Interdire une sortie si la quantité demandée dépasse le stock disponible.

2.3. Gestion des commandes clients

Description : Permet de créer et suivre les commandes des clients.

Données :

- Commande : id, date, état (EN_COURS, VALIDÉE, ANNULÉE), total.
- Ligne de commande : id, commande, produit, quantité, prix unitaire, montant ligne.

Fonctionnalités :

9. Créer une nouvelle commande.
10. Ajouter un produit à la commande (vérifier la disponibilité du stock).
11. Modifier la quantité / supprimer une ligne (recalcul du total).
12. Annuler une commande (gérer la remise en stock si nécessaire).
13. Valider la commande.

Règles de gestion :

- Au moins une ligne pour valider une commande.
- Le total de la commande est la somme des montants de chaque ligne.

2.4. États et statistiques

Fonctionnalités possibles :

- Chiffre d'affaires par jour (somme des ventes pour une date).
- Chiffre d'affaires sur une période.
- Top produits vendus sur une période (en quantité).
- Produits en rupture ou en dessous du seuil.

2.7. Gestion des utilisateurs (simplifiée)

Données :

- Utilisateur : id, login, mot de passe.

Fonctionnalités :

14. Écran de connexion (login, mot de passe).
15. Redirection vers le menu principal si authentification réussie.

Règles de gestion :

- Mot de passe masqué à l'écran.
- Login unique.

3. Exigences non fonctionnelles

3.1. Ergonomie

Interface claire, adaptée à un poste de travail (souris + clavier).

- Navigation via un menu principal ou une barre d'outils.
- Utilisation de tableaux pour afficher les listes (produits, commandes, ventes).
- Boutons clairement nommés : Ajouter,Modifier, Supprimer, Valider, Annuler.

3.2. Performance

Application fluide pour un volume raisonnable (quelques milliers de lignes).

- Recherches quasi instantanées dans ce contexte.

3.3. Qualité du code

Respect de la POO : classes claires, responsabilité unique.

- Architecture MVC.
- Code commenté de manière minimale mais suffisante.
- Noms de classes et méthodes explicites.

3.4. Sécurité & robustesse

Validation des champs (pas de texte vide pour les noms, pas de quantités négatives, etc.).

- Gestion des exceptions avec messages d'erreur compréhensibles.
- Mot de passe non affiché en clair dans l'interface.

4. Spécifications techniques

4.1. Environnement

- Langage : Java SE.
- Interface graphique : Java Swing.
- IDE : NetBeans.

4.2. Modèle objet (entités principales)

Exemples de classes : (les attributs à compléter au besoin)

- Categorie : id, libelle.
- Produit : id, nom, categorie, prixVente, stockActuel, seuilAlerte.
- MouvementStock : id, produit, type, quantite, dateMouvement, motif.
- Commande : id, dateCommande, etat.
- LigneCommande : id, produit, quantite, prixUnitaire, montantLigne.
- Utilisateur : id, login, motDePasse.

4.4. Persistance des données

Les données doivent être persistées dans une base de données MySQL.

4.5. Interface graphique - écrans minimum

- Écran de connexion.
- Menu principal.

- Gestion des produits.
- Gestion du stock.
- Prise de commande.
- Statistiques.

5. Critère de correction

Les exigences techniques suivantes seront rigoureusement prises en compte pour la correction du présent sujet de travaux pratiques:

- Respect des bonnes pratiques de POO (classes claires, code réutilisable, bonne nomenclature, etc...).
- Fonctionnalités implémentées conformément à l'énoncé
- Qualité de l'interface graphique
- Gestion correcte des interactions avec la base de données (MySQL).
- Validation des données et gestion des erreurs utilisateur.

Bonne chance !