TP1 : Les Bases de Java

Exercice 1 : Manipulation de types primitifs et tableaux

1. Déclarez un tableau nombres de 5 entiers et initialisez-le avec les valeurs 10, 20, 30, 40, 50.
2. Créez une variable somme de type int et calculez la somme des éléments du tableau.
3. Créez une variable moyenne de type double et calculez la moyenne des éléments du tableau.
4. Affichez la somme et la moyenne.

Exercice 2 : Opérations sur les chaînes

1. Déclarez une variable prenom de type String et initialisez-la avec votre prénom.
2. Déclarez une variable nom de type String et initialisez-la avec votre nom.
3. Créez une variable nomComplet qui concatène prenom et nom avec un espace entre les deux.
4. Affichez la longueur de nomComplet.
5. Affichez nomComplet en majuscules.

Exercice 3 : Calcul de Prime d'Ancienneté

Contexte : Une entreprise souhaite calculer la prime d'ancienneté de ses employés selon certains critères.

1. Déclarez les variables suivantes :

1. anciennete (int) : nombre d'années d'ancienneté
2. salaire (double) : salaire mensuel en euros
3. performance (char) : notation de performance (A, B, C ou D)

2. Calculez la prime selon les règles suivantes :

1. Si l'ancienneté est inférieure à 5 ans : pas de prime
2. Si l'ancienneté est entre 5 et 10 ans :
    1. 2% du salaire pour une performance A
    2. 1% du salaire pour une performance B
    3. Pas de prime pour C ou D
3. Si l'ancienneté est supérieure à 10 ans :
    1. 5% du salaire pour une performance A
    2. 3% du salaire pour une performance B
    3. 1% du salaire pour une performance C
    4. Pas de prime pour D

3. Affichez le montant de la prime calculée.

Exercice 4 : Calculateur de Frais de Livraison

Contexte : Une boutique en ligne calcule ses frais de livraison en fonction du pays de destination et du montant de la commande.

1. Déclarez les variables suivantes :

1. paysLivraison (String) : code du pays de livraison (FR, BE, DE, IT, ES)
2. montantCommande (double) : montant de la commande en euros
3. estPremium (boolean) : indique si le client est un membre premium

2. Utilisez une structure switch pour déterminer les frais de base selon le pays :

1. FR : 5€
2. BE, DE : 8€
3. IT, ES : 10€
4. Autre : 15€

3. Appliquez ensuite les règles suivantes :

1. Si le montant de la commande est supérieur à 100€, réduisez les frais de 2€
2. Si le client est premium, divisez les frais par 2
3. Si le montant de la commande dépasse 200€, les frais sont gratuits

4. Affichez les frais de livraison finaux.

Exercice 5 : Analyse Météo

Contexte : Vous allez analyser les données météorologiques d'une semaine pour 3 villes. Utilisez les données fournies ci-dessous pour répondre aux questions suivantes. N'utilisez que des tableaux et des boucles (for, while, do-while) pour résoudre ces problèmes. Créez une méthode séparée pour chaque question.

```
// Températures journalières pour 3 villes sur 7 jours (en °C)
double[][] temperatures = {
   {12.5, 14.0, 13.5, 15.0, 16.5, 15.5, 14.0}, // Ville 1
   {11.0, 12.5, 13.0, 14.5, 15.0, 14.0, 13.5}, // Ville 2
   {13.0, 14.5, 15.0, 16.0, 17.5, 16.5, 15.5}  // Ville 3
};
// Précipitations journalières pour 3 villes sur 7 jours (en mm)
double[][] precipitations = {
   {0.0, 5.5, 2.0, 0.0, 0.0, 1.5, 3.0}, // Ville 1
   {2.0, 0.0, 0.0, 1.0, 3.5, 2.5, 0.0}, // Ville 2
   {1.0, 0.5, 0.0, 0.0, 4.0, 3.0, 0.5}  // Ville 3
};
String[] villes = {"Paris", "Lyon", "Marseille"};
String[] jours = {"Lundi", "Mardi", "Mercredi", "Jeudi", "Vendredi", "Samedi",
"Dimanche"};
```

1. En utilisant une boucle for :

a) Calculez et affichez la température moyenne de la semaine pour chaque ville.
b) Trouvez et affichez le jour le plus chaud de la semaine, toutes villes confondues.

2. En utilisant une boucle while :

a) Pour chaque ville, comptez et affichez le nombre de jours sans pluie (précipitations = 0).
b) Trouvez et affichez la plus longue séquence de jours consécutifs sans pluie pour chaque ville.

3. En utilisant une boucle do-while :

a) Calculez la température moyenne de même jour pour les 3 villes.
b) Affichez les jours où la température moyenne dépasse 15°C.

Exercice 6 : Listes de Courses

Contexte: Vous allez analyser une liste de courses en tenant compte des allergies
Utilisez les données fournies ci-dessus pour répondre aux questions suivantes. N'utilisez que des tableaux et des boucles (for ou while)

```
String[][] listes = {
        {"pommes", "pain", "fromage", "dattes", "oeufs", "poisson", "raisins"},
        {"lait", "beurre", "yaourt", "fromage", "crème"},
        {"carottes", "brocolis", "épinards", "tomates", "concombres"}
    };
    String[] allergies = {"dattes", "poisson", "brocolis"};
```

1. Affichez les articles de la première liste, en sautant ceux dans allergies.
2. Trouvez le premier produit laitier dans la première liste. Arrêtez la recherche après l'avoir trouvé. :
3. Utilisez break. Comptez les légumes de la troisième liste jusqu'à "tomates" inclus. Utilisez break.

Exercice 7 : Création de la Classe Personne

1. Créez une classe nommée Personne avec les attributs suivants :
- nom (String)
- age (int)

2. Incluez deux constructeurs :
- Un constructeur qui prend nom et age en paramètres.
- Un constructeur par défaut qui initialise nom à "Inconnu" et age à 0.

3. Créez une méthode afficherInfo qui imprime le nom et l'âge de la personne.
4. Dans la méthode main, créez une instance de Personne en utilisant le constructeur par défaut et appelez afficherInfo.

Exercice 8 : Gestion des Comptes Bancaires

Contexte: Vous travaillez pour une banque qui souhaite moderniser son système de gestion des comptes clients. On vous demande de créer une classe CompteBancaire qui permettra de gérer les opérations de base sur un compte.

Variables d'instance :

- **numeroCompte** : le numéro du compte (int)
- **solde** : le solde actuel (double)
- **nomProprietaire** : le nom du propriétaire du compte (String)

Méthodes :

- **Un constructeur** : pour initialiser le numéro de compte, le nom du propriétaire et le solde initial.
- **deposer(double montant)** : pour ajouter de l'argent au compte
- **retirer(double montant)** : pour retirer de l'argent du compte
- **getSolde()** : pour obtenir le solde actuel
- **afficherInfosCompte()** : pour afficher toutes les informations du compte

1. Créez une variable de classe (statique) tauxInteret de type double pour représenter le taux d'intérêt commun à tous les comptes. Quelle différence y a-t-il entre l'accès à cette variable et l'accès à la variable d'instance solde ?

2. Ajoutez une méthode statique modifierTauxInteret(double nouveauTaux) pour changer le taux d'intérêt. Pourquoi cette méthode devrait-elle être statique ? Comment l'appelleriez-vous depuis une autre classe ?

3. Implémentez une méthode statique **transferer(CompteBancaire source, CompteBancaire destination, double montant)** pour transférer de l'argent entre deux comptes. Pourquoi cette méthode peut-elle être statique, contrairement à deposer() ou retirer() ?