Projet

Algorithmique et programmation : notions de base

§ 1 Objet du projet

L'objet de ce devoir est d'écrire un programme composé de plusieurs classes permettant de manipuler des factures pour une entreprise de vente de produits de bureau. À l'issue de ce projet, vous aurez trois classes à rendre :

- Catalogue servant à manipuler la liste des produits vendus par l'entreprise avec leur prix,
- Facture servant à collecter les différentes factures de l'entreprise,
- Menu, le menu principal qui permettra de réaliser quelques opérations.

Pour vous aider, les squelettes des classes Catalogue et Facture vous sont donnés sur Moodle.

§ 2 Notion de variable de classe

Pour limiter le nombre de paramètres des différentes méthodes que nous allons écrire, nous utilisons des variables dites de classe. Ces variables sont partagées par toutes les méthodes de la classe. Dans certains langages de programmation, on parle parfois de « variables globales ». Ce n'est pas exactement le cas en Java, mais le principe en est très proche.

Téléchargez puis importez dans Eclipse le fichier « VariableDeClasse.java » fourni sur Moodle et qui est un exemple de création et d'utilisation de ces variables de classe. Ce fichier est fortement documenté pour vous expliquer ce concept et son utilisation.

§ 3 La classe Catalogue

Pour chaque produit, nous gardons le nom du produit et son prix.

Pour mémoriser le catalogue de l'entreprise, nous utiliserons deux variables de classe :

- `tabNoms` un tableau de `String` qui contiendra le nom des produits,

- `tabPrix` un tableau de `double` qui contiendra le prix des produits.

Chaque produit sera identifié par un numéro qui sera en fait l'indice de la case des deux tableaux qui contient l'information concernant le produit. Par exemple, `tabNoms[1]` contiendra le nom du produit numéro 1 alors que `tabPrix[1]` contiendra son prix.

Le squelette de cette classe `Catalogue` vous est fourni (fichier Catalogue.java) avec 4 produits prédéfinis. Il y a aussi un petit main qui vous permettra de vérifier vos codes. Le résultat attendu par ce programme est le suivant :

```
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
Catalogue initial :
------------------
Stylo bleu = 1.2€
Stylo rouge = 1.25€
Cahier petit format = 2.0€
Cahier grand format = 3.0€
------------------
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
Catalogue avec classeur A4 en plus :
------------------
Stylo bleu = 1.2€
Stylo rouge = 1.25€
Cahier petit format = 2.0€
Cahier grand format = 3.0€
Classeur A4 = 1.55€
------------------
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
Catalogue avec classeur A4 et ramette A4 en plus :
------------------
Stylo bleu = 1.2€
Stylo rouge = 1.25€
Cahier petit format = 2.0€
Cahier grand format = 3.0€
Classeur A4 = 1.55€
Ramette A4 = 2.55€
------------------
Produit 0 : Stylo bleu = 1.2€
Produit 4 : Classeur A4 = 1.55€
Pas de trousse
```

À vous de compléter cette classe en suivant les directives ci-dessous.

3.1 Lire les informations d'un produit

Dans la classe fournie, modifiez la méthode `getNom(int numProduit)` qui permet de récupérer le nom du produit dont le numéro est `numProduit`. Faites en de même avec `getPrix(int numProduit)` pour en récupérer le prix.

3.2 Afficher le catalogue complet

Toujours dans la classe fournie, écrivez la méthode `afficher()` qui a pour but d'afficher à l'écran le contenu complet du catalogue. Le résultat doit être à peu près de la forme :

```
------------------
Stylo bleu = 1.2€
Stylo rouge = 1.25€
Cahier petit format = 2.0€
Cahier grand format = 3.0€
------------------
```

Ne cherchez pas à embellir le résultat pour l'instant. Si vous souhaitez le faire, finissez d'abord le projet et, seulement quand vous aurez tout fait, vous pourrez revenir sur votre code.

3.3 Ajout d'un produit au catalogue

Le but de la méthode `ajouter(String nom, double prix)` est, comme son nom l'indique, d'ajouter un produit au catalogue. En commentaire dans la classe fournie, vous sont données quelques indications pour y arriver.

3.4 Chercher un produit par son nom

Écrivez la méthode `chercher(String nom)` dont le rôle est de trouver le numéro d'un produit à partir de son nom qui est pris en paramètre. Si le produit n'est pas trouvé, le résultat sera -1.

§ 4 Le facturier

Comme pour le catalogue, le squelette d'une classe Facture vous est fourni. Utilisez le comme base de votre travail. Vous y trouverez aussi un autre main dont le résultat attendu peut être de la forme :

```
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
Facture n°0
----------------------
Stylo bleu (1.2€ par pièce) x 10 pièces = 12.0€
Cahier grand format (3.0€ par pièce) x 2 pièces = 6.0€
----------------------
Montant total : 18.0€
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
Facture n°1
----------------------
Stylo rouge (1.25€ par pièce) x 1 pièces = 1.25€
Stylo bleu (1.2€ par pièce) x 5 pièces = 6.0€
Cahier petit format (2.0€ par pièce) x 5 pièces = 10.0€
----------------------
Montant total : 17.25€

-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
Facture n°0
----------------------
Stylo bleu (1.2€ par pièce) x 10 pièces = 12.0€
Cahier grand format (3.0€ par pièce) x 2 pièces = 6.0€
----------------------
Montant total : 18.0€

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
Facture n°1
----------------------
Stylo rouge (1.25€ par pièce) x 1 pièces = 1.25€
Stylo bleu (1.2€ par pièce) x 5 pièces = 6.0€
Cahier petit format (2.0€ par pièce) x 5 pièces = 10.0€
----------------------
Montant total : 17.25€

-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.
```

4.1 La structure de données

Le facturier est composé d'un certain nombre de factures. Chaque facture contient un certain nombre de lignes. Chaque ligne est composée du numéro du produit et de la quantité achetée.

La solution proposée pour conserver ces informations consiste à créer deux tableaux à deux dimensions : un tableau pour mémoriser le numéro des produits de chaque facture, l'autre la quantité. Une dimension identifiera la facture, la seconde la ligne de la facture. Il faut noter que le nombre de lignes est variable d'une facture à l'autre.

Définissez une variable de classe pour chacun des deux tableaux. Les deux tableaux seront initialisés à un tableau vide (tableau de taille 0 pour dire qu'il n'y a pas de facture au préalable).

Expliquez en commentaire dans le code quelle dimension servira de numéro de facture, laquelle servira de numéro de lignes de facture et pourquoi ce choix.

4.2 Créer une facture vide

Écrivez la méthode `nouvelleFacture()` pour ajouter une nouvelle facture à la structure de données. La facture sera initialement vide (aucune ligne). La question suivante permettra de compléter la facture. La méthode retournera le numéro de la facture ainsi créée.

Vous pouvez vous inspirer de l'aide de la question 3.3 pour agrandir vos deux tableaux.

4.3 Ajouter une ligne à une facture

Complétez la méthode `ajouterProduit(int numFacture, int numProduit, int quantite)` qui a pour but d'ajouter une ligne à la facture de numéro `numFacture`. Le paramètre `numProduit` est le numéro du produit acheté, `quantite` en est la quantité achetée.

4.4 Affichage à l'écran d'une facture.

Complétez la méthode `afficherFacture(int numFacture)` qui a pour but d'afficher une facture. La présentation de la facture peut être celle donnée ci-dessous.

De nouveau, attendez d'avoir fini complètement le projet avant de chercher à faire du « beau » comme, par exemple, un tableau bien propre. Allez à l'essentiel en priorité.

---

xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

```
Facture n°1
----------------------
Stylo rouge (1.25€ par pièce) x 1 pièces = 1.25€
Stylo bleu (1.2€ par pièce) x 5 pièces = 6.0€
Cahier petit format (2.0€ par pièce) x 5 pièces = 10.0€
----------------------
Montant total : 17.25€
```

4.5 Affichage du facturier

Prévoir l'affichage de toutes les factures.

§ 5 Menu

Concevez un programme qui permette de manipuler le catalogue et le facturier de l'entreprise.

Ce menu proposera comme choix :

- affichage d'un produit (numéro, nom et prix) à partir de son nom que l'utilisateur entrera au clavier,
- ajout d'un produit dont les données seront lues au clavier,
- affichage du catalogue complet,
- affichage d'une facture par son numéro,
- ajout d'une facture (la liste des produits achetés et leur quantité seront lus au clavier),
- affichage du facturier,
- sortie du programme (pas de sauvegarde demandée).