

Contrôle module M2207 (POO & Java)

vendredi 31 mars 2017
durée : 1h
responsable : M.Munier

2 pages

Documents autorisés : Pour cette évaluation vous êtes autorisés à utiliser tous les documents pédagogiques qui vous ont été fournis : supports de cours "M2207 & Java" (éventuellement annotés) distribués par M.Munier, corrections des TD/TP, notes de TD et notes de TP. Tout autre document (documentations récupérées sur le web, photocopies de livres, etc. . .) est formellement interdit. Dans le même ordre d'idée, aucune assistance électronique (calculatrice, ordinateur de poche, téléphone portable, PDA, . . .) n'est autorisée. Toutes les notions vues en cours sont supposées acquises et, le cas échéant, doivent être mises en œuvre.

NB : Pour cet examen vous serez évalués sur vos compétences en programmation orientée objets : classes, attributs, constructeurs, méthodes, héritage, . . . L'algorithmique "pure" (boucles, tests, . . .), bien qu'indispensable à la réalisation de votre travail, ne constitue pas le cœur de cette évaluation ! (cf. M1207)

★ ★ ★

Problème : Gestion de Stock

Dans cet exercice on se propose de programmer en Java un ensemble de classes nécessaires à la réalisation d'une application de gestion du stock d'une entreprise. Nous aurons tout d'abord des objets représentant les produits (avec leurs caractéristiques, leur prix unitaire HT, . . .). Pour représenter le stock de notre entreprise, nous utiliserons une seconde classe d'objets. L'état du stock sera matérialisé par une "sorte de tableau" dont chaque ligne contiendra une référence produit, le nom d'un fournisseur et la quantité actuellement en stock. L'exercice se terminera par un programme de test.

Travail demandé :

1. Écrire la classe **Produit** dont chaque instance représentera une catégorie d'articles en stock. Chaque produit est identifié par un code article (une chaîne de caractères), un libellé (une chaîne de caractères) et le prix unitaire HT de cet article. Donc uniquement ces 3 attributs.
Outre les méthodes d'accès classiques, cette classe disposera des méthodes **prixHT()** et **prixTTC()** retournant respectivement (sous forme de réel) le prix Hors Taxes et le prix Toutes Taxes Comprises (avec un taux de TVA par défaut de 20%).
2. Écrire la classe **Stock** dont chaque instance représentera un stock de l'entreprise avec la liste de tous les produits (ainsi que le fournisseur auquel s'est adressé l'entreprise et la quantité actuellement disponible). Pour le moment, vous n'avez vu que les tableaux en Java. A vous de faire avec, tout en

sachant que tous les éléments d'un tableau (quelle que soit sa dimension) sont du même type¹. La classe `Stock` devra disposer des méthodes permettant :

- d'enregistrer un nouveau produit avec (le nom de) son fournisseur et la quantité livrée ; l'entête de la méthode sera `void enregistrer(Produit p, String nomFour, int qte)`
- d'afficher l'état actuel du stock avec, pour chaque produit, les informations sur son fournisseur et la quantité disponible (méthode `void afficher()`)
- de calculer la valeur hors taxes actuelle du stock (méthode `double calculerValeurHT()`)
- **[Bonus]** de calculer la valeur hors taxes actuelle du stock provenant d'un fournisseur dont le nom est passé en paramètre² (méthode `double calculerValeurHT(String nomFour)`)
- de calculer la valeur TTC actuelle du stock (méthode `double calculerValeurTTC()`) en prenant bien évidemment en compte le taux de TVA applicable en fonction des différents articles (cf. question suivante sur l'héritage)

3. (*Héritage* \Rightarrow vous pouvez très bien poursuivre l'exercice même si vous ne faites pas cette question...)

Écrire la classe `ProduitTVA` (héritant donc de `Produit`) permettant de représenter des produits ayant un taux de TVA spécifique (indiqué via le constructeur). Il est donc nécessaire d'ajouter un attribut. À vous de voir s'il est nécessaire d'ajouter et/ou redéfinir certaines méthodes...

4. Pour conclure cet exercice, il vous est demandé d'écrire un petit programme permettant de réaliser les tests suivants sur les classes que vous venez d'écrire. Pour instancier les différents objets, vous coderez les valeurs "en dur" dans le programme de test (i.e. pas de saisie au clavier).

- création d'un stock,
- création de plusieurs produits,
- enregistrement de produits/fournisseurs/quantités dans le stock,
- affichage du stock,
- calcul puis affichage de la valeur de ce stock en HT puis en TTC

Question de Cours

1. Qu'est ce que la liaison dynamique ? Expliquez son fonctionnement à l'aide d'un exemple (en vous inspirant éventuellement de l'exercice précédent... même si vous n'avez pas répondu à toutes les questions...).



Duke store online : everything you need is here...

Document rédigé en L^AT_EX sous Linux.

1. Et souvenez vous que la taille d'un tableau est fixée une fois pour toutes lors de son initialisation \rightarrow vous devez "surdimensionner" le tableau.

2. `String` est une classe. Pour comparer deux chaînes de caractères vous devrez donc utiliser la méthode `equals` et non pas faire une comparaison des références avec `==`. Par exemple, si vous avez deux chaînes `chA` et `chB`, la comparaison se fait via l'instruction `chA.equals(chB)` qui retourne `true` ou `false`.