

Mini-projet

Déroulement du mini-projet :

Le mini-projet s'étale sur les deux mardis matin du 6 avril et du 13 avril (soit 6h de TP).

L'énoncé ci-dessous est prévue pour ces 2 séances. Toutefois un complément "bonus" ayant pour but de gérer le menu avec une interface graphique en utilisant la classe Swing est envisagé pour le 13 avril pour les étudiants les plus rapides et désirant découvrir cet outil.

Le projet est à rendre pour le vendredi 16 avril minuit.

Le rendu est une archive contenant votre code accompagné d'un fichier (txt ou pdf) contenant les réponses principales aux questions (les dessins peuvent être des photos jpg à part).

Le projet est à rendre en binôme ou en monôme.

Nomenclature de pièces

L'entreprise JLFB propose des meubles ou des ensembles de meubles en kit livrés par lot de cartons. Afin de réduire les coûts, des mêmes composants se retrouvent entre deux meubles différents : ainsi une même vis peut être un composant d'une chaise ou d'un bureau, également deux cuisines différentes peuvent être composées des mêmes éléments de placards.

Nous appellerons ici *pièce* les produits vendus par JLFB. Pour gérer cette grande diversité de meubles et d'ensembles de meubles, l'entreprise JLFB utilise une *nomenclature* permettant de donner à chaque pièce un *numéro de référence* (un nombre entier), une *dénomination* (un petit texte) et un poids (un flottant). Une pièce peut éventuellement être en fait une composition d'autres pièces de la nomenclature. Une pièce a un poids *poids total* qui est le poids de la pièce plus la somme de tous les poids des pièces qui composent la pièce. La pièce composite correspondant à un meuble en kit, ce sera le poids de l'emballage vide plus le poids des pièces placées dedans ; dans le cas d'une cuisine, elle a un poids nul et un poids total qui le poids de tous les cartons de meuble et d'électroménager qui la compose.

Il est important de noter qu'une pièce est composée de pièces qui peuvent elles-même être composées de pièces (une cuisine est composée de placards qui sont eux-mêmes composés de vis et de planches). On parlera alors de sous-composants d'une pièce. Considérons l'exemple suivant : un tabouret est composé d'une planche et de quatre pieds ; un pied est une pièce emballée à part, composé d'une barre de bois et de 3 vis : ainsi un tabouret est composé de 1 planche, 4 pieds, 4 barres de bois et 12 vis, soit 21 sous-composants.

Dans ce projet, nous allons créer un logiciel proposant un menu en mode texte afin de manipuler cette nomenclature.

Il est fortement recommandé de tester au fur et à mesure chaque classe et méthode implémentée. Dans un tel projet, une fonction erronée peut ne pas produire d'erreurs à première vue mais être pourtant la source d'erreurs par la suite.

A la fin de ce projet, des tests seront effectués de manière à vérifier chacune des fonctions demandées, en utilisant des nomenclatures de très grande taille.

Exercice 1 – Hiérarchie de classes

Pour modéliser le problème, nous allons considérer deux types de pièces : les pièces de base (qui ne sont pas composées d'autres pièces) et les pièces composites qui correspondent à des pièces obtenues par assemblage d'autres pièces (composites ou de base).

Q 1.1 Donner le schéma de hiérarchie des classes modélisant le problème en utilisant une classe abstraite `Piece` et deux classes `PieceDeBase` et `PieceComposite`. N'oubliez pas le nombre d'occurrences des pièces dans une pièce.

Q 1.2 Implémentez ces classes en pensant à l'encapsulation des données et en utilisant des structures de données adaptées.

Q 1.3 Pour la classe `PieceDeBase`, écrire un constructeur avec les 2 paramètres nécessaires et une méthode `toString`.

Q 1.4 Pour la classe `PieceComposite`, écrire un constructeur qui construit une pièce composite avec une liste de pièces vide. Ajouter une méthode permettant d'ajouter une pièce dans la liste des pièces.

Q 1.5 Ajouter une méthode retournant une chaîne de caractères contenant la liste des noms des pièces composant une pièce composite. Utiliser cette méthode dans la surcharge de la méthode `toString` de la classe `PieceComposite`.

Cette méthode pourra appeler une méthode `affiche_tous_les_sous_composants(int decal)` qui affiche tous les composants et sous-composants d'une pièce. Cet affichage doit permettre de bien visualiser l'arborescence des sous-composants. Pour cela, le paramètre `decal` permet d'indiquer que la pièce à afficher doit être décalée de `decal` espaces du bord de la fenêtre.

Q 1.6 Créer une classe `MainPiece` qui crée quelques pièces de base et quelques pièces composites. En particulier, on considère une pièce composite p_1 . Montrer comment l'on ajoute des pièces de bases et des pièces composites dans la liste des pièces composant la pièce p_1 .

Q 1.7 Écrire une méthode permettant d'afficher le poids d'une pièce. Donner un exemple dans votre classe `Menu_essai`. Afficher par la méthode `toString`.

Exercice 2 – Détection d'erreurs

Q 2.1 Que se passe-t-il si l'on utilise la méthode `toString` d'une pièce composite p_1 , qui possède la pièce p_1 dans sa liste de pièces ?

Q 2.2 Proposer une méthode qui détecte si une pièce est une composante (ou sous-composante) d'une pièce. Cette méthode doit être utilisable pour toutes les types de pièces.

Q 2.3 Utiliser l'algorithme implémenté dans la méthode précédente pour lever une exception en cas d'ajout d'une pièce créant le problème détecté ci-dessus : (l'exception sera traitée à l'extérieur).

Exercice 3 – Nomenclature

Q 3.1 Ajouter une classe **Nomenclature** qui contient son nom (texte) et une structure de données contenant toutes les pièces de l'entreprise JLFB. Ajouter une méthode permettant d'ajouter une pièce dans la nomenclature.

Q 3.2 Donner une méthode `toString` affichant toutes les pièces de la nomenclature. **Q 3.3** Ajouter une exception levée par en cas d'ajout d'une nouvelle pièce ayant même numéro de référence.

Q 3.4 Ajouter une méthode de recherche d'une pièce dans la liste des pièces à partir de son numéro de référence.

Q 3.5 Ajouter une méthode affichant toutes les pièces (composites) contenant une pièce passée en paramètre.

Q 3.6 Ajouter une méthode permettant de supprimer une pièce de la nomenclature. On n'autorise cette suppression que seulement si cette pièce n'entre pas dans la composition d'une autre pièce.

Q 3.7 Ajouter une méthode `Menu` qui sera lancée par `MainMenu`. Le menu texte affiché est comme ceci. 0-Exit

1-Nouvelle nomenclature

2-Ajouter une pièce sans ces composants

3-Afficher la nomenclature

4-Liste les pièces de même dénomination

5-Ajouter un composant à une pièce

6-Afficher tous les détails d'une pièce

7-Supprimer une pièce

Relier le menu avec les méthodes vues précédemment : les pièces seront saisies au clavier en saisissant leur numéro de référence. Pensez à lever/traiter toutes les exceptions nécessaires !

Exercice 4 – Pièces particulières

Q 4.1 Créer une pièce **Lampe** qui possède une puissance en unité Lumen. De même créer une pièce composite **Cadres** qui peut contenir un ensemble de cadres et de plaques de verres pour l'affichage de photos.

Q 4.2 A l'image des deux pièces précédentes, des pièces fragiles à stocker dans une zone sécurisée de l'entrepôt afin de pas risquer de les fragiliser par des vibrations. Proposer une solution pour pouvoir stocker dans la nomenclature une liste des pièces fragiles, sans changer la hiérarchie des classes. On veut pouvoir produire à tout moment un document donnant la liste des pièces fragiles en indiquant comment les tenir pour les déplacer.

Exercice 5 – Gestion du stock

Q 5.1 Ajouter un item 8-Trier qui trie la nomenclature par numéro de références des pièces. Pour cela, vous utiliserez la méthode `sort()` de la classe `Collections`.

Exercice 6 – Chargement et sauvegarde d'une nomenclature

On veut ajouter les items
9-Sauvegarder la nomenclature
10-Lire une nomenclature

Pour cela, nous allons manipuler le format “ad hoc” suivant :

```
test1 8
34 boulon 0.1
    -1 -1
56 table 0.5
    77 1
    21 4
    -1 -1
77 planche 7.2
    -1 -1
21 pied 0.2
    34 3
    37 1
    -1 -1
37 barre 3.4
    -1 -1
43 vase 1.2
    -1 -1
88 tabouret 0.2
    37 3
    19 1
    -1 -1
19 planche 3.4
    -1 -1
```

La première ligne contient le nom de la nomenclature et la seconde le nombre total de pièces. Chacun des blocs de lignes suivants correspond à une des pièces.

Une pièce est donnée par une première ligne comportant son numéro de référence, sa dénomination, qui est un mot sans espace, puis son poids. Les lignes suivantes donnent les composants de la pièce sous la forme d'une liste des paires (références, nombre d'occurrence), cette liste se termine par deux fois l'entier -1.

Q 6.1 Implémentez la sauvegarde d'une nomenclature.

Codez le chargement d'une nomenclature. Attention, lors du chargement, les données peuvent être telles qu'une pièce apparaît d'abord en tant que composant avant d'apparaître en tant que pièce ! Pensez éventuellement à lire 2 fois le fichier si nécessaire.

Exercice 7 – Tests expérimentaux

Q 7.1 Vous testerez toutes vos questions en utilisant l'exemple `test1.nom` et en produisant vos propres exemples permettant de vérifier que votre programme fonctionne dans tous les cas. Vous testerez également votre code en utilisant l'instance de grande taille `test2.nom` fournie.

Q 7.2 Comparez les temps nécessaires au chargement et à la sauvegarde de cette grande instance. Comment pouvez-vous l'expliquer ? Comment modifier le format du fichier afin d'accélérer ce chargement.