IUT des Pays de l'Adour - Dpt RT - BUT2
R308 - Consolidation de la programmation

Évaluation de Travaux Pratiques

décembre 2024
durée : 1h
responsable : M.Munier

Sujet 4 - Réserve

Documents autorisés : Pour cette évaluation de travaux pratiques vous êtes autorisés à utiliser tous les documents pédagogiques qui vous ont été fournis : support de cours "consolidation de la programmation" (module R308) (éventuellement annotés) distribué par M.Munier, corrections des TD/TP, notes de TD et notes de TP. Vous pouvez également utiliser tout autre document (documentations récupérées sur le web, photocopies de livres, etc...) que vous jugerez utiles pour... vous faire perdre du temps. Par contre aucune assistance électronique (calculatrice, ordinateur de poche, smartphone, mail, Google+, Facebook, Twitter, Google Glass,...) n'est autorisée pour communiquer avec qui que ce soit. Toutes les notions vues en cours sont supposées acquises et, le cas échéant, doivent être mises en œuvre.

Restitution de vos travaux : Vous développerez toutes les classes de cette évaluation dans un seul et unique répertoire (éventuellement un seul fichier) dont le nom est **votre nom de famille en minuscules**. À la fin de la séance votre "copie d'examen" sera une archive (formats `zip` ou `tar` uniquement) contenant ce répertoire et dont le nom sera votre nom de famille en mininuscules, sans accents et sans espaces. Vous enverrez ensuite votre archive par mail à l'adresse manuel.munier@gmail.com

Conseils : Durant cette évaluation de TP (qui ne dure que 2h), votre but n'est pas de tout faire, mais plutôt de nous prouver que vous êtes capables de vous débrouiller en programmation objet en Python devant une machine. Cela signifie bien évidemment savoir écrire des classes correctes (constructeurs, méthodes, attributs,...), mais également savoir interpréter les erreurs de syntaxe et/ou d'exécution, les corriger, tester vos classes, etc... Dit de manière un peu plus directe, **il est plus "intéressant" pour vous d'avoir un programme fonctionnel validant une partie des classes que d'avoir écrit toutes les classes mais sans en avoir testé aucune !**

Tout manquement sur la procédure de restitution de votre travail sera immédiatement sanctionné sur votre note !

⋆ ⋆ ⋆

Exercice POO

L'objectif de ce sujet est de programmer en Python des classes nécessaires à la réalisation d'une application de gestion d'une réserve de produits : des produits, une armoire de stockage,...

1. Écrire la classe `Produit` représentant un produit. Cette classe aura 2 attributs `description` et `PU` correspondant respectivement au nom du produit (une chaîne de caractères) et son prix unitaire en € (un réel). Outre le constructeur et les méthodes d'accès aux attributs, vous devrez implémenter la méthode `valeurStock(qte)` qui retourne la valeur en stock (prix total en €) pour `qte` exemplaires de ce produit.

2. Écrire maintenant la classe `Armoire` permettant de gérer une réserve de produits. Dans l'idée, un objet de cette classe contiendra une collection de références à des produits avec, pour chacune, la quantité actuellement en stock. À vous de choisir quels attributs utiliser pour cette classe. Outre le constructeur et les méthodes d'accès aux attributs, vous devrez implémenter les méthodes suivantes :
   - `ajouter(refProduit,qte)` qui ajoute dans l'armoire un nouveau produit avec la quantité indiquée
   - `valeur()` qui retourne la valeur totale de ce stock en €

3. Afin de pouvoir valider votre travail, il vous est naturellement demandé d'écrire un programme de test créant plusieurs produits, de les ajouter à une armoire, et d'invoquer les différentes méthodes.

4. Écrire la classe `Vrac`, sous-classe de la classe `Produit`, permettant de représenter des produits conditionnés en vrac. Cette sous-classe ajoutera un nouvel attribut `unite` indiquant la quantité unitaire. Prenons un exemple : supposons que notre produit soit de la `"farine de blé complet"` en vrac au prix de `15` € (valeur de `PU`) les `5` kg (valeur de `unite`). L'exécution sur cet objet de l'appel de méthode `valeurStock(2)` devra retourner `6`, c'est-à-dire le prix en € de 2kg de farine.

5. Valider votre code par un programme de test mélangeant dans un même objet `Armoire` différents objets des classes `Produit` et `Vrac`.

Exercice SDD

Sur la base du code fournit dans le cours et testé durant les TP à propos des listes (simplement) chaînées, ajouter à la classe `LinkedList` une méthode `impairs()` qui, dans cette liste, va supprimer tous les nœuds dont la valeur est paire. Pour ne garder donc que les nœuds dont la valeur est impaire. Illustrons ceci par un exemple :

- si la liste initiale est `1` $\rightsquigarrow$ `10` $\rightsquigarrow$ `5` $\rightsquigarrow$ `11` $\rightsquigarrow$ `6` $\rightsquigarrow$ `2` $\rightsquigarrow$ `8` $\rightsquigarrow$ `5` $\rightsquigarrow$ `null`
- le résultat sera la liste `1` $\rightsquigarrow$ `5` $\rightsquigarrow$ `11` $\rightsquigarrow$ `5` $\rightsquigarrow$ `null`

* * *

Document rédigé en LaTeX sous Linux.