

Série 7

Exercice 1

Nous voulons écrire un programme en Python servant à déterminer les prix de vente pour le compte d'une entreprise qui achète des produits élémentaires pour les utiliser dans la fabrication d'autres produits dits composés.

Dans ce système d'information, un produit ne peut être qu'élémentaire ou composé. Pour chaque produit composé, il faut connaître sa composition (ou constitution), c'est-à-dire les produits élémentaires qui entrent dans sa fabrication avec les quantités utilisées.

Exemple : P1 est élémentaire, P2 est élémentaire, P3 est constitué de 2 unités de P1, et 4 unités de P2, P4 est constitué de 3 unités de P2 et de 2 unités de P1

1. Définissez les classes suivantes :

Classe composition ayant :

- Deux attributs privés __produit et __quantité
- Un constructeur à deux paramètres
- les getters et setters pour les deux attributs en utilisant le décorateur @property

Classe produit ayant :

- Deux attributs privés __nom et __code
- Un constructeur à deux paramètres
- les getters pour les deux attributs en utilisant le décorateur @property
- une méthode abstraite getPrixHT

Classe produit élémentaire ayant :

- un attribut privé __prixAchat
- Un constructeur
- une méthode __str ()__ qui retourne les informations sur un produit élémentaire
- une méthode getPrixHT qui retourne le prixAchat d'un produit élémentaire

Classe produit composé ayant :

- un attribut privé __fraisFabrication, qui représente le coût engendré par la fabrication d'un produit à partir de ses composants (la valeur des composants n'est pas considérée)
- un attribut tauxTVA qui est le même pour tous les produits composés (18%)
- un attribut privé __listeContituants qui représente un tableau de compositions.
- Un constructeur
- les getters pour les attributs en utilisant le décorateur @property
- une méthode __str ()__ qui retourne les informations sur un produit composé
- une méthode getPrixHT qui calcule et retourne le prix hors taxe qui est la somme des prix hors taxe de tous les constituants majoré par les frais de fabrication du produit en question.

2. Dans un fichier main.py, testez toutes les classes et méthodes que vous avez implémentées.

Exercice 2

Cette activité est une suite de l'activité précédente

1. Créer une liste ListeProduit contenant la liste de tous les produits (élémentaires/ composés) créés
2. Pour chaque produit de ListeProduit, créer un objet Description de type namedtuple

Description= namedtuple('Description', ['Produit', 'Détail'])

Le champ Produit: correspond au nom du produit

Le champ Détail: présente une description textuelle du produit

Par exemple:

- Si le produit P1 est un produit élémentaire, le champ nom ="P1 " et le champ Détail=" P1 est un produit Elémentaire"
- Si le produit P3 est un produit composé de P1 et P2, le champ nom="P3" et le champ Détail est" P3 est composé de P1 et P2 "

3. Convertissez chaque instance de Description créée en un dictionnaire

4. Créer ListeDescription un objet de type defaultdict contenant toutes les descriptions des produits

5. Créer un objet DeqDescription de type deque. Ajouter les éléments de ListeDescription dans DeqDescription