

Nom :
Prénom :
Numéro d'étudiant :
Groupe :

Contrôle continu 1

Tous documents sur support papier autorisés. Durée : 1h30.

L'ensemble des réponses sera à donner sur les feuilles d'énoncé. Ne pas dégrapher les feuilles.

On s'intéresse à un magasin de vente de produits surgelés. Un produit est décrit par une date limite d'utilisation optimum (DLUO) et d'un numéro de lot. Un produit est muni d'une méthode permettant de calculer (en jours) le temps restant avant la péremption du produit. Il est de plus muni d'une méthode calculant une chaîne descriptive du produit qui est la concaténation du nom de la référence cliente avec le numéro de lot. Un produit a une référence de produit.

Une référence de produit a un nom, un taux de TVA qui est soit normal soit réduit, un poids, et un prix hors taxe. Une référence produit a de plus une analyse nutritionnelle. Une référence produit est munie d'accesssieurs non modélisés mais dont on supposera disposer, et d'une méthode qui calcule le prix TTC (i.e. incluant la TVA).

Une analyse nutritionnelle est décrite par une énergie (en KJoules), un poids de matières grasses (en g), un poids d'acides gras saturés (en g), et d'autres informations comme le taux de sel, de protéines et de glucides qu'on ne mentionnera pas ici pour simplifier. La classe **AnalyseNutritionnelle** possède de plus un attribut de classe **analyseNutritionnelleRef** qui représente l'analyse nutritionnelle de référence, c'est-à-dire correspondant aux apports journaliers recommandés par les normes européennes.

On donne un modèle partiel de la classe **AnalyseNutritionnelle** à la figure 1.

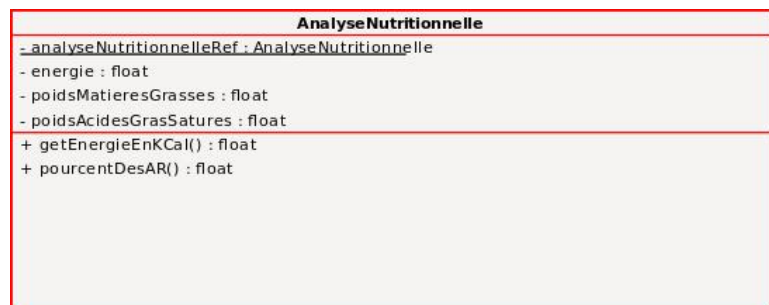


FIGURE 1 – Diagramme de classes à compléter pour l'analyse nutritionnelle

Question 1. Représentez sous forme d'un diagramme d'objet une instance d'**AnalyseNutritionnelle** qui représente les apports journaliers recommandés : l'énergie est 8400 KJoules, les matières grasses 70g, les acides gras saturés 20g.

Question 2. Complétez sur la figure 1 la classe **AnalyseNutritionnelle** afin d'y ajouter un constructeur paramétré adéquat, qui ne s'occupera pas de la création de l'analyse nutritionnelle de référence.

Question 3. Complétez ci-dessous le diagramme de classes pour introduire les produits et les références de produits. On ne fera figurer ni les accesseurs ni les constructeurs.

Question 4. Ecrivez en Java le code de la méthode `getEnergieEnKCal` qui retourne l'énergie en KCalories (sachant que $1\text{kJoule}=0,239006\text{KCal}$).

Question 5. Ecrivez en Java le code de la méthode `pourcentDesAR` qui retourne le pourcentage d'énergie apportée par l'énergie d'une analyse nutritionnelle par rapport à celle de l'analyse nutritionnelle de référence. Par exemple, si une analyse nutritionnelle a pour énergie 840 KJoules, elle couvre 10% des apports journaliers recommandés (qui sont de 8400), donc la méthode `pourcentDesAR` retournerait 10.

Question 6. On donne à la figure 2 un modèle de la classe **Commande** qui représente une commande de références de produits. Cette classe a un attribut référençant les références de produits commandés. Elle est munie d'une méthode qui calcule le poids total de la commande, une qui calcule la capacité requise pour stocker la commande dans un congélateur, une méthode qui calcule le prix des produits commandés, et une méthode qui calcule le prix total de la commande. Les accesseurs ne sont pas mentionnés mais on suppose en disposer.

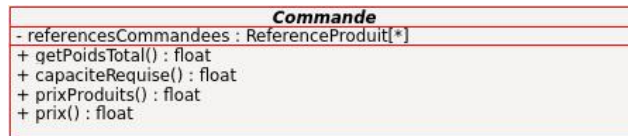


FIGURE 2 – Diagramme de classes à compléter pour les commandes

a- Complétez le diagramme de classes de la figure 2 afin de représenter les classes **CommandeAvecLivraison** et **CommandeRetraitMagasin**. Le prix d'une commande livrée est majoré d'un prix forfaitaire de livraison. Les commandes avec retrait en magasin spécifient le nom du magasin dans lequel la commande doit être retirée. On ne fera pas figurer les accesseurs ni les constructeurs.

b- Ecrivez en Java la méthode **capaciteRequise** de la classe **Commande**. La capacité requise est calculée en litres en multipliant par 5 le poids en kg de la commande.

c- Ecrivez en Java la méthode **prix** dans la classe **Commande** ainsi que si vous le jugez nécessaire dans ses sous-classes. Le prix de toute commande est donné par le prix des produits commandés majoré d'un euro pour les frais de préparation.

