

projet POO JAVA GL1

Sujet 1

Proposer une classe Equation2d avec trois propriétés représentant les coefficients de l'équation de

type double et disposant des méthodes suivantes:

- Un constructeur avec trois arguments de type double
- Un constructeur par copie d'un objet de type Equation2d
- Une méthode d'instance resoudre() qui renvoie la solution sous forme de chaîne de caractère.
- Créer une classe TestEquation2d permettant de créer deux instances de la classe Equation2d et de renvoyer leurs solutions.

Sujet 2

Dans cet exercice, on s'intéresse à la gestion des tickets de caisse concernant les achats de clients dans un supermarché. Le supermarché propose un ensemble d'articles aux clients. Un article est décrit par une référence (String) et un prix unitaire.

1. Donner l'implémentation d'une classe Article comportant une méthode Lire pour la saisie au clavier et une méthode Afficher.

On s'intéresse aux achats des clients dans le supermarché. Un client peut acheter plusieurs articles avec des quantités différentes. On appelle LigneAchat la description de l'achat d'un article donné, Une LigneAchat est décrite par l'article (référence et prix unitaire) acheté, et la quantité achetée de l'article.

2. Donner l'implémentation de la classe LigneAchat comportant une méthode Lire pour la saisie au clavier, une méthode SommeAchat qui calcule le prix de revient de la quantité achetée et une méthode Afficher.

Maintenant, on s'intéresse au ticket de caisse délivré par le caissier au client après paiement de ses achats, on l'appelle TicketAchat. Un TicketAchat comporte une date (la date du jour, réinitialisée chaque jour), un numéro de ticket séquentiel, une ou plusieurs Ligne Achat et affiche la somme totale des achats.

3. On suppose que chaque client achète exactement 5 articles différents avec les quantités voulues. Donner l'implémentation de la classe TicketAchat dans ce cas, en définissant un constructeur, une méthode InitDate qui initialise la date, une méthode InitNumero qui initialise à 1 le numéro de ticket, une méthode Lire pour la saisie des LigneAchat au clavier, une méthode Total qui calcule la somme totale des achats et une méthode Afficher qui affiche le ticket à délivrer au client avec les détails de ses achats.

4. A partir de la classe TicketAchat, implémenter une classe TicketAmélioré qui permet de donner la recette (somme d'argent globale) récoltée dans la caisse en fin de journée.

5. Ecrire le programme qui crée 500 tickets de caisse (TicketAchat) et affiche la recette récoltée en fin de journée.

6. En supposant que le nombre d'articles achetés diffère d'un client à l'autre et n'est pas fixé ni limité, donner une autre implémentation de la classe TicketAchat (de la question 3). Ne pas réécrire les méthodes InitDate et InitNumero.

SUJET 3

Créez une classe qui s'appelle `Forme2D` qui représente, de façon générique, des formes géométriques à deux dimensions. Cette classe doit contenir : — un attribut dont la visibilité est `private` qui s'appelle `couleur` et qui représente la couleur de la forme en question — un constructeur par défaut (qui ne prend aucun paramètre) qui crée une forme 2D dont la couleur est noire. — un constructeur qui prend en paramètre une couleur et qui crée une forme 2D dont la couleur correspond à celle-ci. — un accesseur qui permet de récupérer la valeur de la couleur de la forme 2D. — une méthode `aire` (qui est sensée retourner l'aire de la forme 2D) qui retourne 0. (N.B. Cette méthode sera redéfinie dans les classes filles de `Forme2D` pour calculer l'aire correctement) — une méthode `perimetre` (qui est sensée retourner le périmètre de la forme 2D) qui retourne 0. (N.B. Cette méthode sera redéfinie dans les classes filles de `Forme2D` pour calculer le périmètre correctement) Maintenant, créez une classe `Rectangle` qui hérite de la classe `Forme2D`. La classe `Rectangle` doit contenir : — deux attributs supplémentaires `longueur` et `largeur`. — un constructeur qui prend en paramètres une couleur, une longueur et une largeur et qui instancie le rectangle correspondant. — les accesseurs adéquats pour les deux attributs `longueur` et `largeur`. — une redéfinition de la méthode `aire` qui retourne l'aire du rectangle. — une redéfinition de la méthode `perimetre` qui retourne le périmètre du rectangle. Créez également une classe `Disque` qui hérite elle aussi de la classe `Forme2D` et qui contient : — un attribut qui s'appelle `rayon`. — un constructeur qui prend une couleur et un rayon en paramètre et qui crée le disque correspondant. — un accesseur pour l'attribut `rayon`. — une méthode `diametre` qui renvoie le diamètre du disque. — les redéfinitions des méthodes `aire` et `perimetre` qui calculent l'aire et le périmètre d'un disque correctement. Au choix dans l'une des deux classes `Rectangle` et `Disque`, essayez d'écrire un modificateur `setCouleur` qui prend en paramètre une couleur et qui l'attribue à la forme concernée (autrement dit, essayez de modifier l'attribut `couleur` que `Rectangle` et `Disque` héritent de leur classe mère `Forme2D`). Est-ce que ça marche ? Pourquoi ? Modifiez la visibilité de l'attribut `couleur` dans la classe `Forme2D` pour la passer en `protected`. Est-ce que ça marche maintenant ? Amusez-vous à créer des rectangles et des disques. Considérez les tantôt comme appartenant à leurs

classes spécifiques, tantôt comme étant des formes 2D et vérifiez quelles sont les méthodes que vous pouvez invoquer ou non

NB: Projet à rendre avant le 23/07/2023, Zippez le projet et renommez le projet sous votre nom et prénoms d'école