UNIVERSITE DU 8 MAI 1945
Faculté des Mathématiques et de l'Informatique
Et des Sciences de la Matière
DEPARTEMENT D'INFORMATIQUE

28/01/2015

D.Bourouaieh 2015 – 2eme LI

Examen POO S1

Exercice 1. 8pts

Q1. Écrivez la classe Piscine selon les spécifications suivantes:
Chaque piscine se caractérise par les attributs suivants:
nomPiscine: chaine de caractère indiquant le nom de la piscine.
nombreMax: nombre entier positif maximum de personnes pouvant accéder en même temps à la piscine.
tarifNormal: tarif d'entrée à la piscine pour les personnes ne bénéficiant d'aucune réduction.
tarifReduit: tarif d'entrée à la piscine pour certaines catégories de clients.
netNormal: nombre d'entrées vendues au tarif normal
netReduit: nombre d'entrée vendues au tarif réduit.
À la création d'un objet Piscine, les attributs netNormal et netReduit sont à zéro. Il est possible de créer une nouvelle piscine en spécifiant le nom, le nombreMax, le tarifNormal et le tarifReduit.
Il est aussi possible de créer un objet Piscine avec seulement le nom et le nombreMax, dans ce cas un tarifStandard est appliqué comme tarif normal et le tarif réduit sera la moitié (1/2) du tarifStandard. Ce tarifStandard est le même pour toutes les piscines et il est possible de le modifier avec une méthode. Il doit rester le même pour toutes les Piscines. Le tarifStandar a une valeur par défaut de 250.
Écrivez ce deuxième constructeur avec une seule instruction.

Pour utiliser la classe Piscine on doit disposer des méthodes publiques suivantes:
public int nombrePlaceDisponible(…): retourne le nombre de places encore disponibles.

public boolean vendrePlace(…): par ce même nom il y'a trois (03) méthodes qui permettent, repspectivement,

1. de vendre un nombre de place (paramètre nombrePlace) soit au tarif normal soit au tarif réduit selon la valeur d'un paramètre booléen tarifReduit ,

2.de vendre une place (01) selon un paramètre booléen tarifReduit (s'il est à true alors appliquer le tarif réduit sinon le tarif normal),

3. et la troisième de vendre une (01) place au tarif normal. Écrivez la 2eme et 3eme méthode avec une seule instruction chacune.

La vente d'une place consiste à incrémenter l'un des deux attributs netNormal ou netReduit. Les trois méthodes renvoient un booléen: true si la vente peut se faire, false sinon (pas assez de place, aucune n'est vendue).

public void RAZ(): permet de commencer une autre session de piscine, toutes les places sont à vendre (piscine vide). C'est une remise à zéro des nombre de places vendues.

public int chiffreAffaire(): affiche la somme d'argent résultat de toutes les ventes effectuées pendant la session en cours.

public double tauxRemplissage(): renvoi le taux de remplissage de la piscine. Par exemple 0.56 indique que la piscine est remplie à 56%.

Il est possible aussi d'avoir ces deux méthodes:

private boolean disponible(**int** n) // true si n places sont disponibles
private int nombreVendu() // renvoi le nombre de places vendues depuis le dernier RAZ()

Q2. Testez votre classe avec les actions suivantes:
- créez une piscine "biga" de 100 places
- créez une piscine "doha" de 200 places avec un tarif normal de 350 et un tarif réduit de 250.
- vendre 30 places au tarif normal dans "biga"
- vendre 01 place au tarif réduit dans "doha"
- afficher le chiffre d'affaire de "biga" et de "doha"
- mettez le tarifStandard à 250.

Exercice 2. 8pts

Tous les Liquides ont une densité et une température, attributs **privés**. Il est possible de réchauffer de n degrés les liquides, ainsi leur température augmente de n degrés et leur densité augmente de 1 pour chaque degré en plus. Il est aussi possible de refroidir les liquides de n degré (température et densité décroissent de n). La classe Liquide dispose d'une méthode **boolean** isTempAbove(**int** n): cette méthode renvoi true si l'attribut privé température est supérieur (>) à n, false sinon.
Une autre méthode de cette classe permet de mélanger le liquide sur lequel est invoquée cette méthode avec un autre Liquide en paramètre. Le résultat du mélange est un nouveau produit de densité du premier et la température du deuxième.
n'écrivez pas de méthodes set...() ni get...() pour accéder aux attributs privés.
Q1. Écrire la classe Liquide avec un constructeur qui initialise tous les attributs.

Dans les liquides manipulés il y'a deux types spéciaux (en plus des autres):
Les liquides **vertBleu**, ont un attribut couleur qui est **vert** si leurs température est inférieur à une température tChange et **rouge** sinon. Cette classe dispose d'une méthode getCouleur(...) qui renvoi la couleur en cours.
Les liquides **dangereux**, chauffent doublement, c-à-d, en leur demandant de chauffer de n, ils chauffent de 2 x n.
Q2. Écrire les deux classes LiquidesVertBleu et LiquideDangereux. (Avec Constructeurs)
Q3. Ecrire une classe Test qui dispose d'un tableau de 4 Liquides (le tableau doit contenir tous les types de liquide possibles). Chauffer tous les éléments du tableau de 10, en plus, si l'élément est un Liquide vertBleu affichez sa couleur.

Exercice 3. 4pts

Q1. Soit le code:

```
class A{ int n; public void m(){n=1;}}

class B extends A{ public void m(){n=2;}}

class C extends A{public void m(int s){n=s;}}
```

- corrigez ce code: A e=new B(); B b=e; A s=new C(); s.m(3);
- Quelles est la valeur de n dans ces cas:

1. A e=new B(); e.m();

2. A e= new C(); e.m();