01234567890123456789012345

Université d'Aix-Marseille III - Licence de Math-Info

IN301 : Programmation Objet - TP n^o1

1 Prise en main de l'environnement

Editer, compiler (commande `javac` avec la variable `classpath` contenant les bons chemins) et exécuter (commande `java`) le programme suivant :

```
public class PremierEssai
{
    public static void main(String args[])
    {
        System.out.println("Ca marche");
    }
}
```

2 Premier programme

Ecrire un programme (une suite d'instructions dans une méthode `main`) qui lit un entier n au clavier, crée un tableau de n entiers, remplit ce tableau avec des entiers lus au clavier puis affiche le plus petit entier du tableau (en parcourant ce tableau). En plus de la méthode `main`, vous ajouterez (sans essayer pour l'instant de la comprendre) la méthode suivante, qui permet de lire un entier au clavier :

```
static int litInt()
{
    String s = "";
    BufferedReader b =new BufferedReader(new InputStreamReader (System.in));
    try
    {
        s=b.readLine();
    }
    catch(java.io.IOException e)
    {
        System.out.println ("Erreur de lecture");
        System.exit(0);
    }
    return Integer.parseInt(s);
}
```

3 Classes et instances

Ecrire la classe `Verre` ayant les deux attributs entiers `contenance` et `quantite`, un constructeur prenant en paramètre la contenance (en cl) du verre créé, la méthode `void emplir(int q)` qui ajoute q cl de liquide à la quantité contenue dans le verre et la méthode `boire(int q)` qui fait l'inverse.

Ecrire la classe `Bouteille` ayant un attribut entier `quantite` et un attribut booléen `est_ouverte`, un constructeur prenant en paramètre la quantité initiale (en cl) de la bouteille créée, les méthodes `void ouvrir()` et `void fermer()` et la méthode `void verser_dans(Verre v, int q)` qui verse la quantité q de liquide de la bouteille dans le verre v.

Ajouter une méthode `main` à la classe `Bouteille` pour exécuter la suite d'opérations suivante : création d'une bouteille d'orange et d'une bouteille de vodka d'un litre chacune, création d'un verre de contenance

20cl, versement de 8cl de vodka puis de 12cl d'orange dans le verre, puis consommation avec modération de tout le verre.

4 Composition de classes

Nous allons modéliser des émetteurs radios et des auditeurs, situés dans un espace à deux dimensions. Un récepteur peut se modéliser grâce à ses coordonnées : son abscisse et son ordonnée. Un émetteur peut se modéliser grâce à ses coordonnées, sa portée (le rayon du cercle dans lequel un auditeur peut le capter) et sa fréquence d'émission.

4.1 *Ecrire la classe* **Emetteur** *qui contient les méthodes suivantes* :

- un constructeur qui prend en paramètre ses coordonnées, sa portée et sa fréquence.
- `double abscisse()` qui retourne son abscisse.
- `double ordonnee()` qui retourne son ordonnée.
- `double portee()` qui retourne sa portée.
- `double frequence()` qui retourne sa fréquence.
- `void augmenteFrequence()` qui augmente sa fréquence d'une unité.

4.2 *Ecrire la classe* **Auditeur** *qui contient les méthodes suivantes* :

- un constructeur qui prend en paramètre ses coordonnées.
- `double distance(Emetteur e)` qui retourne la distance entre l'auditeur et l'émetteur. Rappel : la distance entre deux points de coordonnées (x1,y1) et (x2,y2) est $\sqrt{(x1-x2)^2+(y1-y2)^2}$. En Java, la racine carrée de x s'obtient par l'appel `Math.sqrt(x)`.
- `boolean recoit(Emetteur e)` qui indique si l'auditeur peut recevoir l'émission provenant de `e`, c'est-à-dire, si sa distance à `e` est inférieure à la portée de `e`.
- `void controleEmetteurs(Emetteur e1, Emetteur e2)` qui, si l'auditeur peut recevoir des émissions de `e1` et `e2` et qu'ils émettent à la même fréquence, demande à `e1` d'augmenter sa fréquence d'une unité (sinon les émissions se perturbent).

Attention : dans aucune de ces méthodes, on ne fera référence aux attributs de la classe `Emetteur`.

4.3 *Compléter la classe* **Emetteur** *afin qu'elle constitue un programme qui définit un émetteur de coordonnées (0,0), de portée 5 et de fréquence 100, un autre émetteur de coordonnées (1,1), de portée 5 et de fréquence 100, un auditeur de coordonnées (2,3), qui ensuite effectue un contrôle des deux émetteurs. Enfin, on affiche les (nouvelles ?) fréquences des deux émetteurs.*