

Contrôle 1 - Instructions simples et conditionnelles, classes

CORRECTION

Question 1. Savoir écrire une séquence d'instructions dans un main.

Remplissez le main du listing 1 afin qu'il effectue les opérations suivantes pour calculer le périmètre d'un rectangle dont la largeur et la longueur sont saisies au clavier :

1. Créer un clavier (objet de la classe **Scanner**)
2. Afficher un message demandant la largeur du rectangle
3. Saisir au clavier cette largeur
4. Afficher un message demandant la longueur du rectangle
5. Saisir au clavier cette longueur
6. Calculer le périmètre
7. Afficher le périmètre
8. Fermer le clavier

Listing 1 – Programmel.java

```
package controle;  
public class Programmel {  
    public static void main(String[] args){  
  
        Scanner clavier = new Scanner(System.in);  
  
        System.out.println("Entrez la largeur");  
  
        double l = clavier.nextDouble();  
  
        System.out.println("Entrez la hauteur");  
  
        double h = clavier.nextDouble();  
  
        double perimetre = 2*l+2*h;  
  
        System.out.println("Le perimetre vaut "+perimetre);  
  
        clavier.close();  
  
    }  
}
```

Question 2. Savoir utiliser une classe déjà définie. Etudiez le listing de la classe **Train** (listing 2).

Listing 2 – Train.java

```
package controle;
import java.util.Scanner;
public class Train {
    private String identifiant;
    private int nbWagons;
    private int nbPlacesParWagon;

    public Train(String identifiant, int nbWagons, int nbPlacesParWagon) {
        this.identifiant = identifiant;
        this.nbWagons = nbWagons;
        this.nbPlacesParWagon = nbPlacesParWagon;
    }

    public String getIdentifiant() {return identifiant;}
    public void setIdentifiant(String identifiant) {this.identifiant = identifiant;}

    public int getNbWagons() {return nbWagons;}
    public void setNbWagons(int nbWagons) {this.nbWagons = nbWagons;}

    public int getNbPlacesParWagon() {return nbPlacesParWagon;}
    public void setNbPlacesParWagon(int nbPlacesParWagon) {this.nbPlacesParWagon = nbPlacesParWagon;}

    public int nbPlacesDisponibles() {return this.nbWagons * this.nbPlacesParWagon;}
}
```

Remplissez le main du listing 3 afin qu'il effectue les opérations suivantes :

1. Créer une instance de la classe **Train** avec l'identifiant 56344, et 12 wagons contenant chacun 80 places.
2. Modifiez le nombre de wagons pour qu'il devienne égal à 10.
3. Afficher le nombre de places disponibles dans le train en utilisant la méthode correspondante.

Listing 3 – Programme2.java

```
package controle;
public class Programme2 {
    public static void main(String[] args){

        Train t = new Train("56344",12,80);

        t.setNbWagons(10);

        System.out.println(t.nbPlacesDisponibles());

    }
}
```

Question 3. Savoir écrire le code d'une nouvelle classe

a- Ecrire l'entête et les attributs d'une classe représentant le concept de wagon avec les informations suivantes. Un wagon a un numéro (de type chaîne de caractères), il a un standing (entier qui vaut 1 ou 2 pour représenter la 1e classe ou la 2e classe) et il peut faire partie d'un train (l'attribut est de type `Train` et il vaudra `null` si le wagon n'est pas dans un train).

Réponse à la question 3.a :

```
public class Wagon {  
  
    private String numero;  
    private int classe = 2;  
    private Train train;  
  
    ....}
```

b- Ecrire pour la classe `Wagon` un constructeur qui prenne comme paramètres le numéro, le standing, et le train d'appartenance et initialise les attributs correspondants.

Réponse à la question 3.b :

```
public Wagon(String numero, int classe, Train train) {  
    this.numero = numero;  
    this.classe = classe;  
    this.train = train;  
}
```

c- Ecrire pour la classe `Wagon` l'accesseur en modification (`set`) à l'attribut représentant le standing. On ne peut accepter comme standing que les entiers 1 ou 2. Si la nouvelle valeur n'est ni 1 ni 2, on affiche un message d'erreur et l'attribut prend la valeur 2.

Réponse à la question 3.c :

```
public void setClasse(int classe) {
    if (classe ==1 || classe ==2)
        this.classe = classe;
    else {System.out.println("problème de valeur, ce doit etre 1 ou 2");
        this.classe = 2;}
}
```

d- Ecrire pour la classe `Wagon` une méthode `toString` retournant une chaîne de caractères incluant le numéro, le standing, et s'il appartient à un train, l'identifiant de ce train.

Réponse à la question 3.d :

```
public String toString() {
    String s = "Wagon numero=" + numero + ", classe=" + classe ;
    if (train != null)
        s += " train : "+train.getIdentifiant() + "";
    return s;
}
```