

Module 1 – Informatique – Algorithmique et Programmation Objet

Travaux Dirigés (10), Licence 1ère Année

Classes et Objets

Exercice 1 Bistrot

1.1. Créez une classe **Verre** ayant :

- deux attributs **contenance** (contenance du verre) et **quantite** (quantité actuelle du verre) qui sont des entiers,
- un constructeur sans paramètre qui initialise l'attribut **contenance** avec la valeur retournée par la méthode **tailleVerre()** décrite ci-dessous,
- une méthode **int tailleVerre()** qui retourne la contenance qu'aura le verre, valeur choisie aléatoirement entre **25** cl ou **50** cl.
- une méthode **boolean remplir(int q)** qui ajoute **q** cl de liquide à la quantité actuelle contenue dans le verre (attribut **quantite**). On vérifiera que le verre ne déborde pas (la méthode retourne **true** dans ce cas), sinon la quantité n'est pas ajoutée et la méthode retourne **false**,
- une méthode **int boire(int q)** qui enlève **q** cl de liquide à la quantité actuelle contenue dans le verre (attribut **quantite**). Si la quantité actuellement dans le verre est inférieure à **q** alors le verre est vidé. La méthode retourne une valeur entière correspondant à la quantité réellement bue.

Créez une classe **Bistrot** avec une méthode **main()**. Dans cette dernière, créez deux objets **Verre**. Affichez ensuite la contenance et la quantité de chacun des deux verres créés.

Exemple

```
4- public static void main(String[] args) {
5     Verre v1 = new Verre();
6     System.out.println("Contenance verre 1 : "+v1.contenance+" cl et quantité actuelle : "+v1.quantite+"
7     Verre v2 = new Verre();
8     System.out.println("Contenance verre 2 : "+v2.contenance+" cl et quantité actuelle : "+v2.quantite+"
9 }
```

Problems @ Javadoc Declaration Console

<terminated> Bistrot [Java Application] C:\Program Files\Java\jdk1.7.0_60\bin\javaw.exe (25 oct. 2018 23:42:31)

```
Contenance verre 1 : 50 cl et quantité actuelle : 0 cl
Contenance verre 2 : 25 cl et quantité actuelle : 0 cl
```

1.2. Modifiez les attributs **contenance** et **quantite** de la classe **Verre** de façon qu'ils soient déclarés comme privés (mot clé **private** avant leur déclaration). Quel problème rencontrez-vous ?

Ajouter deux accesseurs (méthodes qui retournent la valeur d'un attribut) à la classe **Verre** :

- **int getContenance()** qui retourne la valeur actuelle de l'attribut **contenance** et
- **int getQuantite()** qui retourne la valeur actuelle de l'attribut **quantite**.

Modifier votre **main()** de façon à obtenir le même résultat que précédemment.

Exemple

```

4 public static void main(String[] args) { // Verre v1 = new Verre();
5     Verre v1 = new Verre();
6     System.out.println("Contenance verre 1 : " + v1.getContenance() + " cl et quantité actuelle : "
7         + v1.getQuantite() + " cl");
8     Verre v2 = new Verre();
9     v2.remplir(25);
10    System.out.println("Contenance verre 2 : " + v2.getContenance() + " cl et quantité actuelle : "
11        + v2.getQuantite() + " cl");

```

Problems @ Javadoc Declaration Console

<terminated> Bistrot [Java Application] C:\Program Files\Java\jdk1.7.0_60\bin\javaw.exe (26 oct. 2018 09:11:51)

Contenance verre 1 : 25 cl et quantité actuelle : 0 cl
 Contenance verre 2 : 50 cl et quantité actuelle : 25 cl

1.3. Créez une classe **Bouteille** ayant :

- un attribut privé **quantite** (entier représentant la quantité de liquide contenue dans la bouteille),
- un attribut privé **estOuvverte** (booléen représentant l'état ouvert ou fermé de la bouteille),
- un constructeur sans paramètre qui initialise l'attribut **quantite** avec la valeur retournée par la méthode **tailleBouteille()** décrite ci-dessous, L'attribut **estOuvverte** sera initialisé à **false** (même s'il l'est déjà par défaut),
- une méthode **static int tailleBouteille()** qui retourne la quantité qu'aura la bouteille, valeur choisie aléatoirement entre **75** cls ou **150** cls.
- un accesseur **int getQuantite()** qui retourne la valeur de l'attribut **quantite**,
- un accesseur **boolean getEtat()** qui retourne la valeur de l'attribut **estOuvverte**,
- la méthode **boolean estVide()** qui retourne **true** si la bouteille est vide (**quantite = 0**) et **false** sinon (**quantite > 0**),
- les méthodes **void ouvrirBouteille()** et **void fermerBouteille()**, qui respectivement affectent la valeur **true** et **false** à l'attribut **estOuvverte**,

Dans la méthode **main()**, créez deux objets **Bouteille** et affichez ensuite la contenance et l'état (valeur de l'attribut **estOuvverte**) des deux bouteilles créées.

Exemple

```

4 public static void main(String[] args) {
5     Bouteille b = new Bouteille();
6     System.out.println("Contenance bouteille : " + b.getQuantite() + " cl");
7     System.out.println("\tBouteille vide = " + b.estVide());
8     System.out.println("\tBouteille ouverte = " + b.getEtat());
9 }

```

Problems @ Javadoc Declaration Console

<terminated> Bistrot [Java Application] C:\Program Files\Java\jdk1.7.0_60\bin\javaw.exe (25 oct. 2018 23:37:00)

Contenance bouteille : 75 cl
 Bouteille vide = false
 Bouteille ouverte = false

1.4. Ajoutez à la classe **Bouteille** :

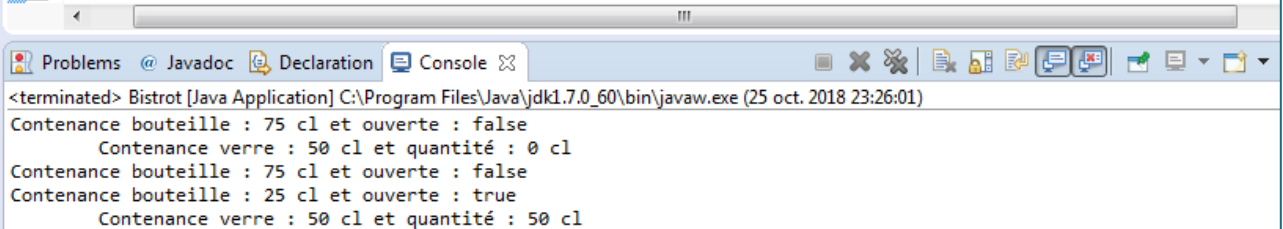
- la méthode **void verserDans(Verre v)** qui remplit à ras bord le verre **v** du liquide de la bouteille. Le liquide versé doit être égal à la valeur de l'attribut **contenance** du verre **v** moins la valeur de l'attribut **quantite** du verre **v**.

Note : le liquide ne sera versé que si la bouteille est ouverte. Tout le contenu de la bouteille sera versé s'il est inférieur à ce que peut recevoir le verre.

Exemple

```

4 public static void main(String[] args) {
5     Verre v = new Verre();
6     Bouteille b = new Bouteille();
7     System.out.println("Contenance bouteille : " + b.getQuantite() + " cl et ouverte : " + b.getEtat());
8     b.verserDans(v); // Demande de remplir le verre
9     System.out.println("\tContenance verre : " + v.getContenance() + " cl et quantité : " + v.getQuantite() +
10    System.out.println("Contenance bouteille : " + b.getQuantite() + " cl et ouverte : " + b.getEtat());
11    b.ouvrirBouteille(); // Demande d'ouverture de la bouteille
12    b.verserDans(v); // Demande de remplir le verre
13    System.out.println("Contenance bouteille : " + b.getQuantite() + " cl et ouverte : " + b.getEtat());
14    b.verserDans(v);
15    System.out.println("\tContenance verre : " + v.getContenance() + " cl et quantité : " + v.getQuantite() +
  
```



```

<terminated> Bistrot [Java Application] C:\Program Files\Java\jdk1.7.0_60\bin\javaw.exe (25 oct. 2018 23:26:01)
Contenance bouteille : 75 cl et ouverte : false
    Contenance verre : 50 cl et quantité : 0 cl
Contenance bouteille : 75 cl et ouverte : false
Contenance bouteille : 25 cl et ouverte : true
    Contenance verre : 50 cl et quantité : 50 cl
  
```

1.5. Ajoutez une classe **Consommateur** ayant :

- Un attribut **nom**, une chaîne de caractères qui représentera le nom du consommateur,
- Un attribut **poids**, un entier qui représentera le poids en kg du consommateur,
- Un attribut **consommation**, un entier qui représentera la consommation totale du consommateur en cl,
- Un attribut **verre**, objet de la classe **Verre**,
- Un attribut **bouteille**, objet de la classe **Bouteille**,
- Un attribut **etat** qui représentera l'état (d'ivresse) du consommateur. L'état du consommateur sera déterminé par sa consommation en cl.
 - si le rapport **consommation/poids < 1**, alors **etat = "Sobre"**,
 - si **1 ≤ consommation/poids ≤ 2.5**, **etat = "Éméché"**,
 - si **consommation/poids > 2.5**, **etat = "Ivre"**.
- Un constructeur acceptant quatre paramètres :
 - **n** (String) et **p** (entier) avec lesquels il initialise respectivement les attributs **nom** et **poids**,
 - **v** (objet **Verre**) et **b** (objet **Bouteille**) avec lesquels il initialise respectivement les attributs **verre** et **bouteille**,
- Une méthode **int gorgee()** qui retourne le quart de la valeur de la contenance du verre du consommateur,
- Une méthode **Bouteille commandeBouteille()** qui crée un objet **Bouteille** et le retourne comme résultat. Quand la bouteille actuelle du consommateur est vide, il en commande une nouvelle qui remplace la précédente,
- Une méthode **void consomme()** qui fait boire le consommateur jusqu'à ce qu'il soit ivre (comportement à ne pas suivre).

Dans la méthode **main()**, créez deux consommateurs, chacun avec un verre et une bouteille et faites-les consommer.

Exemple

```

4 public static void main(String[] args) { // Verre v1 = new Verre();
5     Verre v1 = new Verre();
6     Bouteille b1 = new Bouteille();
7     Consommateur conso1 = new Consommateur("Nicolas", 75, v1, b1);
8     conso1.consomme();
9     Verre v2 = new Verre();
10    Bouteille b2 = new Bouteille();
11    Consommateur conso2 = new Consommateur("Patrick", 95, v2, b2);
12    conso2.consomme();

```

Problems @ Javadoc Declaration Console

<terminated> Bistrot [Java Application] C:\Program Files\Java\jdk1.7.0_60\bin\javaw.exe (26 oct. 2018 09:30:51)

Nicolas est ivre apres avoir consommer 240 cls !
 Patrick est ivre apres avoir consommer 312 cls !