

Module 3 – Aide à la Réussite : EP 2 Soutien en Programmation

Travaux Dirigés (8), Licence 1ère Année

Classes et Objets

Dans l'exercice suivant, nous souhaitons gérer une cave (ou plusieurs) contenant des caisses elles-mêmes contenant des bouteilles.

Exercice

- 1.1. Créez une classe **Bouteille** ayant :
 - un attribut **couleur**, objet **String**,
 - un constructeur acceptant un paramètre **String** avec lequel il initialise l'attribut **couleur**.
 - Une méthode **String getCouleur()** qui retourne l'attribut **couleur**.

- 1.2. Créez une classe **Caisse** ayant :
 - Un attribut entier **taille** qui correspond à la taille de la caisse (nombre maximum de bouteilles pouvant être contenues),
 - Un attribut entier **nombre** (nombre de bouteilles actuellement dans la caisse),
 - un attribut **bouteilles** : un tableau d'objets de la classe **Bouteille**,
 - un constructeur sans paramètre qui initialise l'attribut **bouteilles** aléatoirement soit par trois soit par six objets **Bouteille** qu'il crée. L'attribut **nombre** sera initialisé en conséquence (trois ou six). Chacun des objets **Bouteille** sera créé avec un attribut **couleur** choisi aléatoirement parmi "**rouge**", "**blanc**" ou "**rosé**".
 - Une méthode **String getContenu()** qui retourne le contenu de la caisse sous forme d'une chaîne de caractères.

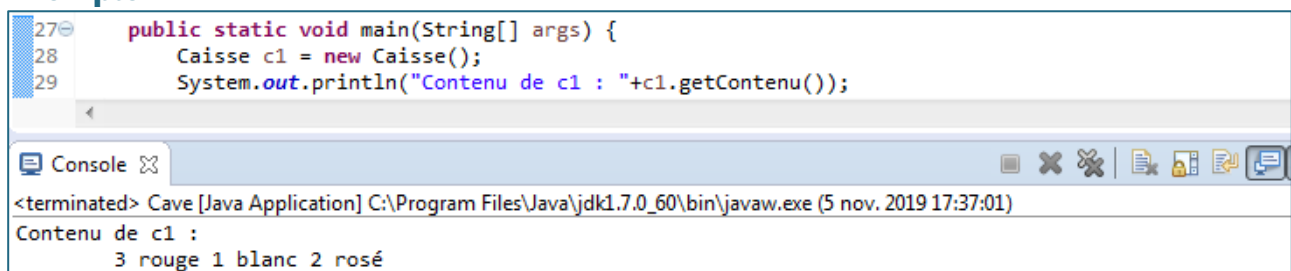
- 1.3. Créer une classe **Cave** comprenant une méthode **main()** et tester ce qui précède.

Exemple

```

27 public static void main(String[] args) {
28     Caisse c1 = new Caisse();
29     System.out.println("Contenu de c1 : "+c1.getContenu());

```



Console

<terminated> Cave [Java Application] C:\Program Files\Java\jdk1.7.0_60\bin\javaw.exe (5 nov. 2019 17:37:01)

Contenu de c1 :
3 rouge 1 blanc 2 rosé

- 1.4. Ajouter à votre classe **Cave** :
 - Un attribut entier **taille** qui correspond à la taille de la cave (nombre maximum de caisses pouvant être contenues),
 - Un attribut entier **nombreCaisses** (nombre de caisses actuellement dans la cave),
 - un attribut **caisses** : un tableau de **taille** objets de la classe **Caisse**,
 - un constructeur avec un paramètre entier qui correspond au nombre de caisses que peut contenir la cave. Ce paramètre servira à initialiser l'attribut **taille** et la dimension de l'attribut **caisses**. L'attribut **nombreCaisses** sera initialisé à 0.

- une méthode **String getContenu()** qui retourne le contenu de la cave sous forme d'une chaîne de caractères (nombre de caisses et nombre et type (couleur) de bouteilles par caisse (réutiliser la méthode **getConetnu()** de la classe **Caisse**).

1.5. Compléter votre méthode **main()** pour tester ce qui précède.

Exemple

```
28 public static void main(String[] args) {
29     Cave cav1 = new Cave(3);
30     System.out.println("Contenu de cav1 : " + cav1.getContenu());
}
```

Console

```
<terminated> Cave [Java Application] C:\Program Files\Java\jdk1.7.0_60\bin\javaw.exe (5 nov. 2019 17:41:20)
Contenu de cav1 : Cave vide !
```

- 1.6. Ajouter à votre classe **Cave** une méthode **ajoutCaisse(Caisse c)** qui accepte en paramètre un objet **c** de la classe **Caisse** et qui l'ajoute à l'objet **Cave**. L'attribut **nombreCaisses** doit être incrémenté si une caisse est ajoutée. La caisse ne doit être ajoutée que si l'objet **Cave** peut la contenir (**nombreCaisses < taille**).

Exemple

```
38 public static void main(String[] args) {
39     Cave cav1 = new Cave(3);
40     System.out.println("Contenu de cav1 : " + cav1.getContenu());
41     Caisse c1 = new Caisse(); cav1.ajoutCaisse(c1);
42     System.out.println("Contenu de cav1 : " + cav1.getContenu());
43     Caisse c2 = new Caisse(); cav1.ajoutCaisse(c2);
44     System.out.println("Contenu de cav1 : " + cav1.getContenu());
}
```

Console

```
<terminated> Cave [Java Application] C:\Program Files\Java\jdk1.7.0_60\bin\javaw.exe (5 nov. 2019 18:08:45)
Contenu de cav1 : Cave vide !
Contenu de cav1 :
    3 rouge 1 blanc 2 rosé
Contenu de cav1 :
    3 rouge 1 blanc 2 rosé
    2 rouge 0 blanc 1 rosé
```

- 1.7. Vous pouvez ajouter :
- Une méthode **void retireBouteille(String c)** à votre classe **Cave** qui retire une bouteille (lui affecter **null**) dont l'attribut **couleur** est égal à **c**, si elle est présente.
 - Une méthode **void retireCaisse()** à votre classe **Cave** qui retire une caisse (lui affecter **null**) lorsqu'elle devient vide.
 - ...