1 Confitures

On définit une classe `Confiture` qui aura comme attributs **privés**

- un attribut `fruit` de type chaîne de caractères ;
- un attribut `proportion` de type entier qui correspondra au pourcentage de fruit dans la confiture ;
- un attribut `cal` de type entier qui correspondra au nombre de calories par 100 grammes de la confiture.

1. Écrivez la classe `Confiture` avec un constructeur public adapté.
2. Écrivez un deuxième constructeur qui ne prend en argument que le fruit et le nombre de calories ; la proportion sera initialisée à 50.
3. Écrivez une méthode publique **d'objet** `description()` et qui renvoie une chaîne de caractères le décrivant ("Confiture de fraise, 50% de fruit, 120 calories aux 100 grammes").
4. Dans un `main` situé dans une classe `Test`, créez un objet de type `Confiture` et affichez sa description.
5. Dans la classe `Confiture`, écrivez une méthode publique d'objet qui prend en argument une quantité en grammes, et donne le nombre de calories correspondant à cette quantité pour cette confiture.
6. Écrivez une méthode publique de prototype `public boolean egal(Confiture c)` qui teste si la confiture `this` a les mêmes attributs que la confiture `c`.
7. On écrit le bout de code suivant situé dans le main de la classe `Test`. Dites quelles lignes ne compilent pas, que produisent les autres lignes ?

```
Confiture c1 = new Confiture("fraise", 50, 120);
Confiture c2 = new Confiture("fraise", 50, 120);
System.out.println(c1.egal(c2));
System.out.println(c1==c2);

System.out.println(c1.fruit);
```

8. On voudrait que l'attribut `fruit` ne puisse être modifié, même par une méthode de la classe `Confiture` ; comment faire ?
9. Écrivez une méthode qui retourne la valeur de `fruit`. Écrivez-en une qui permet de modifier l'attribut `cal`.

2 Pot de Confiture

On définit une classe `Pot` qui représente des pots de confiture. Pour chaque pot, on saura la confiture qu'il contient et sa contenance en grammes.

1. Écrivez la classe `Pot` avec un constructeur public adapté.

2. Écrivez une méthode publique `description` et qui renvoie une chaîne de caractères le décrivant. On pourra utiliser la méthode `description` de `Confiture`. On notera que lors d'un appel à `description`, c'est le type de l'objet sur lequel la méthode est appelée qui permettra au compilateur de décider s'il utilise celle de `Confiture` ou celle de `Pot`.
3. On veut numéroter les pots de confitures, à partir de 1, dans l'ordre de leur création. Comment faire ?
4. Écrivez une méthode statique qui retourne le dernier numéro attribué. Puis écrivez un `main` (dans une autre classe) qui crée un `Pot`, affiche sa description et enfin affiche le dernier numéro attribué.

3 Température

Le but de cet exercice est d'écrire une classe représentant la température. Les trois unités possibles seront "Kelvin", "Celsius" ou "Fahrenheit".
Les méthodes écrites devront toutes être des `méthodes d'objet.`

1. Définir une classe `Temperature`, décrite par un `double` représentant la température, et un `String` représentant l'unité. Définir un constructeur initialisant un objet `Temperature` à zéro Kelvin.
2. Définir un deuxième constructeur prenant en argument un `double` et un `String` et initialisant la température correspondante.
3. Définir un troisième constructeur prenant en argument une `Temperature` et initialisant une copie de celui-ci.
4. Définir des méthodes permettant d'afficher et de modifier chaque élément d'une `Temperature`.
5. Definir une méthode `conversionKC` convertissant une température donnée en Kelvin en une autre donnée en degrés Celsius, et ne faisant rien si la température initiale n'était pas en Kelvin. On rappelle la formule $T_C = T_K - 273.15$.
6. De même, définir une méthode `conversionCF` convertissant une température donnée en degrés Celsius en une autre donnée en degrés Fahrenheit, et ne faisant rien si la température initiale n'était pas en degrés Celsius. On rappelle la formule $T_F = 9/5 * T_C + 32$.
7. Comment tester l'égalité de deux `Temperature`s (même valeur et même unité) ?
8. Définir une méthode `plusGrande` permettant de comparer deux `Temperature`s.