

FICHE 6 : VARIABLES ET MÉTHODES D'INSTANCE ET DE CLASSE

objectifs

- Distinguer les méthodes et variables d'instance et de classe
- Comprendre l'importance du contexte d'invocation (static ou non)
- Manipuler en UML et en Java les variables et méthodes d'instance et de classe.

Vocabulaire

static	final	variable de classe
variable d'instance	constante	

1. Article

Reprenez votre classe `Article`.

On vous demande d'adapter le code afin de faire ce qui suit :

- Il faut mettre le taux de TVA utilisé par défaut dans le constructeur en constante.
- Il faut pouvoir déterminer le nombre d'articles créés et il faut ajouter une méthode de classe qui renvoie le nombre d'articles créés.

Testez votre code !

2. Appel Téléphonique

On vous demande de développer une classe qui permettra de calculer le prix d'un appel téléphonique fait à partir d'une ligne de téléphone fixe. Il faut garder les informations suivantes concernant un appel :

- la date et l'heure de l'appel (utilisez la classe `LocalDateTime` !);
- le numéro appelé ;
- la durée de l'appel (utilisez la classe `Duration` !);
- le tarif à la minute.

Il faudra deux constructeurs : le premier prendra en paramètres toutes les valeurs nécessaires pour initialiser tous les attributs, le deuxième prendra les mêmes paramètres à l'exception du tarif qui sera initialisé à une valeur par défaut (actuellement cette valeur est de 0.15 euro par minute). Cette dernière est la même pour tous les appels téléphoniques et doit aussi être gardée. De plus, il faut pouvoir récupérer et aussi modifier la valeur par défaut du tarif.

Il faut aussi :

- un getter pour chaque attribut ;
- pouvoir calculer le coût d'un appel téléphonique (la minute entamée compte) ;
- pouvoir récupérer les informations d'un appel téléphonique sous forme de texte.

a) Donnez la représentation en UML de la classe `AppelTelephonique`.

b) Une fois votre diagramme validé, implémentez cette classe en java. N'oubliez pas de tester les paramètres.

Remarque : le début du `toString` contenant le formatage de la date est fourni sur moodle.

c) Écrivez une classe permettant de tester la classe `AppelTelephonique`.

3. Supposons une classe `Truc` définie correctement :

Truc
- i : int
<u>+ s : String</u>
...
+ iMethode() : void
+ getI() : int
<u>+ sMethode() : void</u>
...

Si `t` est une variable de type `Truc`, parmi les lignes de code suivantes, lesquelles sont correctes ? Parmi celles qui sont correctes pour le compilateur, distinguez celles dont l'écriture est déconseillée !

<code>System.out.println(t.i);</code>	
<code>System.out.println(t.s);</code>	
<code>t.iMethode();</code>	
<code>t.sMethode();</code>	
<code>System.out.println(Truc.i);</code>	
<code>System.out.println(Truc.getI());</code>	
<code>System.out.println(Truc.s);</code>	
<code>Truc.iMethode();</code>	
<code>Truc.sMethode();</code>	

4. Machin

Qu'est-ce qui cloche avec la classe `Machin` qui suit ?

```
public class Machin {
    private Cercle c = new Cercle();

    public static void main (String[] args){
        methodeA();
    }

    public void methodeA(){
        methodeB();
    }
    public static void methodeB(){
        System.out.println("rayon du cercle : "+ c.getRayon()) ;
    }
}
```