



Nom :  
Prénom :  
N° d'étudiant :

## Compléments en Programmation Orientée Objet

### Contrôle continu de cours n° 1 (20 minutes, sans document)

#### Exercice 1 :

Citez 2 avantages d'utiliser une fabrique statique plutôt que d'appeler directement un constructeur.

#### Exercice 2 :

Voici une classe :

```
1 // Implémentaton d'un compteur incrémentable, dont la valeur initiale est zero.
2 public class Counter {
3     private int count = 0;
4     // incrémente le compteur et retourne sa nouvelle valeur
5     public int incr() { return ++count; }
6     // incrémente deux fois le compteur et retourne sa nouvelle valeur
7     public final int incr2() { incr(); return incr(); }
8 }
```

Écrivez un programme construisant une instance de `Counter` dont les appels à `incr2` ne respectent pas le comportement annoncé (indice : étendez la classe `Counter`).

Comment s'appelle le mal dont est atteint la classe `Counter` ?

Proposez une façon de corriger cette classe (portez vos corrections directement sur le listing de la classe `Counter`).

### Exercice 3 : Traitement de texte

**Le scénario imaginaire.** Vous devez programmer un composant d'édition de texte structuré et formaté capable d'exporter le document édité vers un format de fichier (HTML, LaTeX, Markdown, ...) inconnu à l'avance.

Un tel composant pourra être intégré dans une application, telle une suite bureautique, une application web sociale ou bien un site de blogging.

Ainsi, vous voudriez que le module d'export de fichiers au format désiré par le client/intégrateur puisse être fourni après coup par celui-ci. Ce module devra être capable de créer, à chaque appui sur le bouton "enregistrer" dans votre éditeur, un nouvel assembleur de fichier, lequel fournira un certain nombre de fonctionnalités : ajouter un titre, ajouter un paragraphe, ..., et enfin écrire le fichier (au format spécifique géré par le module fourni).

#### Les questions.

Quel patron de conception permettrait de faire cela (fourniture a posteriori du module d'export) ?

Listez les classes et interfaces intervenant dans ce patron. Reliez les aux différents concepts évoqués dans le scénario, explicitez leurs relations les unes aux autres et indiquez si elles doivent être programmées par vous, l'auteur du composant d'édition, ou bien par le client/intégrateur, auteur de l'application. Vous pouvez répondre à l'aide d'un diagramme de classes UML.