

## Le patron *Passerelle*

### Exercice 1 –

Une compagnie de *n'importe quoi*<sup>1</sup> a mis en place un système pour produire des formulaires d'abonnement à une lettre d'information dans plusieurs formats de manière uniformisée, c'est à dire que la même interface sert à produire les formulaires dans tous les types de formats. Pour l'instant, l'entreprise utilise deux format : *HTML* et *PDF*. L'interface en question se présente comme ceci :

```
public interface Formulaire {  
    public void showTitleHeader(String formTitle);  
    public void showNameLine();  
    public void showFirstNameLine();  
    public void showEmailLine();  
    public String send();  
}
```

Le but de la méthode *showTitleHeader* est d'afficher le nom de la compagnie suivie du titre du formulaire donné en paramètre. Les autres méthodes *showXxx()* affiche le nom du champs correspondant (par exemple "Nom : " pour *showNameLine()*) suivie d'un champs *input* pour le format HTML, ou d'une ligne horizontale pour le format PDF. L'implémentation actuelle comporte une classe abstraite *Formulaire* qui contient le code, commun à tous les formats, de la méthode *showTitleHeader*. Deux classes concrètes, *FormulaireHTML* et *FormulairePDF* ont été dérivées de la classe *Formulaire* pour implémenter les deux formats.

**Problème :** l'entreprise étant maintenant ses activités à l'international et souhaite maintenant pouvoir produire les même formulaires en langue anglaise. Le stagiaire employé pour implémenter cette évolution, qui n'a pas suivi le Master GIL, propose de dériver deux nouvelles classes *FormulaireEnHTML* et *FormulaireEnPDF*.

**Question 1.1 :** Dessinez le diagramme UML de cette solution et faites en une critique, en particulier, imaginez ce qu'il se passe lors de l'ajout d'une nouvelle langue (ex : espagnol, hongrois, ...), et/ou d'un nouveau format (ex : XML, odt).

**Question 1.2 :** Proposez une solution faisant intervenir une abstraction qui permettra de séparer les aspects linguistique et format. Donnez le diagramme UML de cette solution, puis réalisez son implémentation.

### Exercice 2 –

Dans une application, on souhaite disposer d'un système de journalisation des événements (*logger*). Au début on souhaite simplement disposer d'une méthode `log(String)` qui envoie les messages sur la sortie standard du programme. Par la suite, on voudrait pouvoir envoyer ces messages vers un fichier ou une base de données. Enfin, on souhaiterait que le programme puisse utiliser différent types de logger : un pour les avertissement, un pour les erreurs, faisant automatique précéder leur message par "**warning :**" et "**error :**" respectivement.

**Question 2.1 :** Proposez une architecture et donnez son diagramme UML.

**Question 2.2 :** Fournissez une implémentation de votre solution.

---

1. Le *n'importe quoi* est un produit dont personne a besoin mais que les services marketing excellent à vendre.