

TD 4 Design Patterns

commande, chaine de responsabilité

Exercice 1 (commande)

Nous voulons développer des services d'encodage/décodage afin de transmettre des messages de manière sécurisée dans des applications de courrier électronique. Avant d'être envoyé, le message doit être transformé en majuscules puis encodé à l'aide des deux algorithmes suivants :

- i) Le premier algorithme prend en entrée une chaîne de caractère **str**, un entier **i** et un caractère **c** et introduit dans la **str** le caractère **c** tous les **i** caractères en commençant par introduire **c** en première position. Par exemple, pour **str**=« Un premier message.», **i**=3 et **c**= « & », on obtient « &Un &pre&mie&r m&ess&age&.»
- ii) Le second algorithme prend en entrée une chaîne de caractère **str** et en permute toutes les lettres 2 à 2. Par exemple, pour **str**= « Un second message.», on obtient « nUs ceno demssga.e»

Ces deux algorithmes sont disponibles dans la classe **Encodeur** fournie pour le TD.

1) Introduisez le patron de conception « Commande » en considérant que la classe contenant la méthode **main()** joue le rôle de **Client** et que la classe **Encodeur** joue le rôle de **Réalisateur** :

- introduire une commande concrète pour chacun des algorithmes précédemment donnés,
- créer une classe **MailService** jouant le rôle de **Demandeur**,
- fournir un client avec l'algorithme suivant :

```
String s = "Un premier message.";
MailService ms = new MailService();
ms ajouteCommande(new CommandeConversionMajuscule());
ms ajouteCommande(new CommandeEncodeAlgo1(encodeur, '&', 3));
System.out.println(s + " -> " + ms.prepareMsg(s));

String s2 = "Un second message.";
ms = new MailService();
ms ajouteCommande(new CommandeConversionMajuscule());
ms ajouteCommande(new CommandeEncodeAlgo2(encodeur));
System.out.println(s2 + " -> " + ms.prepareMsg(s2));
```

Exercice 2 (chaine de responsabilité)

Nous voulons maintenant pouvoir enchaîner plusieurs commandes tout en donnant l'impression qu'une seule commande est exécutée. Utilisez pour cela, utilisez le patron de conception « Chaine de responsabilité » :

1. *Créer classe abstraite ChaineResponsabilite ;*
2. *Modifier les constructeurs des classes d'encodeur pour pouvoir passer la chaine de caractère à traiter d'une commande à une autre ;*
3. *Réaliser le scénario suivant :*

```
String s = "un exemple de composition";  
MailService ms = new MailService();  
ms.ajouteCommande(new CommandeConversionMajuscule());  
ChaineResponsabilite c1 = new CommandeEncodeAlgo2(encodeur);  
ChaineResponsabilite c2 = new CommandeEncodeAlgo1  
                           (encodeur, c1, '&', 3);  
ChaineResponsabilite c3 = new CommandeEncodeAlgo1  
                           (encodeur, c2, '=', 4);  
ms.ajouteCommande(c3);  
System.out.println(s + " -> " + ms.prepareMsg(s));
```