

Université d'Aix-Marseille III - Licence de Math-Info

I4 : Programmation Objet - Partiel de novembre 2006 Durée : 1h - Aucun document autorisé

Important : dans toutes les classes que vous écrirez, tout attribut devra être déclaré privé.

On veut modéliser du courrier. Pour ceci, on va définir les classes **Feuille**, **Enveloppe** et **Courrier**.

1. Une feuille contient deux pages : son recto (une chaîne de caractères) et son verso (une autre chaîne de caractères). Une seule de ses deux pages est visible à la fois.

*Ecrivez la classe **Feuille** qui modélise une feuille et qui contient les méthodes suivantes :*

- un constructeur qui a 2 paramètres : les deux chaînes de caractères qui constituent le recto et le verso de la feuille. Initialement, seul le recto de la feuille est visible.
- la méthode `void lis()` qui affiche la page visible.
- la méthode `void retourne()` qui simule un retournement de feuille, c'est-à-dire qui ne rend visible que la page qui ne l'était pas.

2. Une enveloppe est caractérisée par les feuilles qu'elle contient (10 au maximum) et le fait d'être ouverte ou fermée.

*Ecrivez la classe **Enveloppe** qui modélise une enveloppe et qui contient les méthodes suivantes :*

- les méthodes `void ouvre()` et `ferme()` qui rendent l'enveloppe ouverte ou fermée. L'enveloppe n'est pas considérée comme cachetée, c'est-à-dire qu'on peut l'ouvrir et la refermer autant de fois qu'on veut.
- la méthode `void accueilleFeuille(Feuille f)` qui ajoute la feuille `f` à celles qui sont déjà présentes.
- la méthode `void lis()` qui lit le texte entier contenu dans l'enveloppe, c'est-à-dire chaque feuille (les deux pages) dans l'ordre dans lequel elles ont été introduites dans l'enveloppe. Cette lecture n'est évidemment possible que si l'enveloppe est ouverte.
- la méthode `int nbFeuilles()` qui retourne le nombre de feuilles actuellement dans l'enveloppe.

3. *Ecrivez la classe **TesteLettre** qui permet l'exécution du programme suivant.* Une enveloppe est créée, ainsi que deux feuilles. La première feuille contient en recto "Blabla" et en verso "Suite du blabla", la deuxième feuille "Encore du blabla" et "Fin du blabla". Ensuite, on place la première feuille puis la deuxième dans l'enveloppe, puis on la ferme, puis on l'ouvre, puis on la lit.

4. Après réflexion, on peut se rendre compte que cette modélisation a un défaut : rien n'empêche qu'on lise une feuille alors qu'elle est dans une enveloppe fermée. Il suffit d'appeler la méthode `lis` sur la feuille. Il faudrait que cet appel de méthode n'affiche rien.

Comment modifier les classes précédentes de manière à ce qu'on ne puisse pas lire le contenu d'une feuille lorsqu'elle est dans une enveloppe fermée ? Vous indiquerez les éventuels attributs à ajouter aux classes ainsi que les nouvelles méthodes ou modifications de méthodes (instructions à y ajouter) pour y parvenir.

5. Un courrier est une enveloppe (remplie de feuilles) particulière. Il faut la cacheter (on ne peut la fermer puis l'ouvrir qu'une seule fois) et il faut l'affranchir pour l'envoyer par la poste.

*Ecrivez la classe **Courrier** qui hérite de la classe **Enveloppe** et ajoute ou redéfinit les méthodes suivantes :*

- la méthode `int tarifAffranchissement()` qui retourne le nombre de centimes correspondant au timbre à mettre : 54 centimes si le contenu du courrier fait au plus 20 grammes (chaque feuille pesant 5 grammes) ou 86 centimes si le contenu du courrier dépasse les 20 grammes.
- la méthode `void ferme()` qui n'autorise la fermeture du courrier qu'une seule fois : la deuxième fois qu'on appelle la méthode `ferme`, rien ne se passe.

Vous ferez en sorte de réutiliser au maximum les méthodes écrites dans les classes précédentes afin d'éviter de réécrire inutilement les instructions déjà présentes dans ces méthodes.