Mise en pratique du MVC

L'objectif de ce TP est de réaliser une application de discussion instantanée (chat). Évidement, nous n'aborderons par les aspects "communication réseau". Nous nous contenterons simplement d'avoir plusieurs instances lancées depuis le même programme qui permettent d'ajouter de nouveaux messages dans l'unique salon de discussion.

L'interface proposée est la suivante :

- La partie centrale affiche les messages échangés entre les participants.
- La zone inférieure rappelle le pseudo de l'utilisateur, affiche une zone de saisie permettant de rentrer le texte, et finalement un bouton OK l'envoie sur le chat. Le message `<pseudo>: <texte>` s'ajoute alors dans la partie centrale de cette fenêtre et sur toutes les autres vues sur le même chat.
- La partie supérieure contient un bouton `Fermer` qui fait disparaître la fenêtre et donc correspond à la déconnexion de l'utilisateur. Les fenêtres correspondant aux autres utilisateurs restent actives et ils peuvent continuer à dialoguer.

Comme dit juste au dessus, quand un utilisateur envoie un nouveau message, celui-ci est reçu par tout le monde, lui compris.

Pour mettre en place ce système, nous utiliserons le patrons MVC avec modèle actif.

Exercice 1: Le modèle de chat

1. Implémenter un modèle de chat permettant d'enregistrer une liste de messages ordonnée.
2. Expliquer pourquoi ce serait une mauvaise idée d'avoir un accesseur `getMessages()` qui retournerait la valeur de l'attribut contenant la liste des messages.
3. Implémenter la classe représentant le salon de discussion de manière à pouvoir faire un foreach sur un objet de ce type, et ainsi obtenir successivement chacun des messages qui composent ce salon.
4. Utiliser le patron observateur amélioré construit au TP précédent pour rendre ce modèle actif. Pour le tester, écrire un observateur qui affiche sur la sortie standard le dernier message ajouté au chat.

Exercice 2: La vue

Intéressons nous à la vue graphique, en Swing, présentant les messages à l'utilisateur. Il s'agit de la partie centrale de la fenêtre. Ce composant graphique devra être capable sans stocker de référence vers le modèle, de représenter tous les messages du salon lors de sa création, et sera mis à jour dès qu'un nouveau message est envoyé.

1. Définir et implémenter ce composant.
2. Écrire un programme utilisant ce composant, et permettant de visualiser graphiquement les messages. Pour le tester on considèrera que le salon contient initialement quelques messages. On ne testera pas l'ajout d'un nouveau message au salon.

Exercice 3: Le contrôleur

La partie inférieure de la fenêtre correspond au contrôleur. Il permet à l'utilisateur de saisir un message et de le diffuser à tous.

1. Définir une classe correspondant au contrôleur.
2. Compléter le programme principal pour y intégrer le contrôleur, selon la méthode qui vous semble la plus appropriée.

Exercice 4: La classe application

1. Factoriser le programme précédemment écrit dans une classe intégrant à la fois la vue et le contrôleur.
2. Modifier le programme principal afin de lancer trois instances de l'application, et tester son fonctionnement.