Mini projet Design Pattern

Problématique

La réalisation de l'interface utilisateur d'une application est un problème complexe. La principale caractéristique de la conception d'une interface est qu'elle doit être souple afin de répondre aux exigences suivantes qui sont celles d'une interface moderne :

- Les utilisateurs de l'application peuvent demander des changements à cette interface pour la rendre plus efficace et plus efficace.
- L'application peut offrir des nouvelles fonctionnalités ce qui nécessite de mettre à jour son interface utilisateur.
- La même information peut être affichée avec différentes vues et saisie au travers de différentes fenêtres.
- L'affichage doit refléter immédiatement les modifications des données manipulées par l'application.

Ces exigences rendent quasi impossible la conception de l'interface utilisateur en l'implantant directement au sein du noyau fonctionnel de l'application. Il convient d'adopter une solution plus modulaire telle que celle que propose le pattern composite MVC.

Le design pattern composite MVC

Les auteurs de Smalltalk-80 ont proposé une solution à ce problème qui s'appelle MVC pour Model-VIew-Controller qui prône la séparation suivante des composants d'une application :

- Model (Modèle) : Il s'agit du noyau fonctionnel qui gère les données manipulées par l'application.
- View (Vue) : il s'agit des composants destinés à afficher les informations à l'utilisateur. Chaque vue est liée à un modèle. Un modèle peut être lié à plusieurs vues.
- Controller (Contrôleur) : un composant de type contrôleur reçoit des évènements en provenance de l'utilisateur et les traduit en requêtes pour le modèle ou la vue. Chaque vue est associée à un contrôleur.

Le pattern composite MVC montre la présence de 3 patterns principaux : Observer, Composite et Strategy

Etude de cas :

On va supposer qu'on a une petite base de donnée de chaussures de sport qui est disponible que pour la consultation. Un menu déroulant est situé en haut de la vue et offre la possibilité à l'utilisateur de choisir les paires de chaussures qu'il désire afficher. La marque, le modèle et le prix sont alors affichés ainsi que le premier avis donné pour ces paires de chaussures.

Un bouton **Avis Suivant** permet d'afficher le prochain avis donné sur les mêmes paires de chaussures. Lorsque l'utilisateur appuie sur ce bouton alors que le dernier avis est affiché, l'interface affiche à nouveau le premier avis

1. Faites une petite recherche sur les design pattern qu'on n'a pas vu dans le cours (Composite, Strategy) et décrivez leur utilité.
2. Concevez le diagramme de classe de cette problématique avec le design pattern MVC composite (Observer, Strategy, Composite).
3. Ecrivez le code nécessaire à votre solution UML. Pour la conception UML , il suffit de me faire un imprime écran de la modélisation faite sur un logiciel de votre choix

NB : Pour faire la conception UML, choisissez le logiciel avec lequel vous êtes familier. (Pas d'exigence dans le choix de logiciel de conception)

Le travail doit être envoyé par email. La date limite sera 20/05/2019