Université Libre de Bruxelles
Service CoDE

ULB

Info-H-302 – Analyse et conception par objets
Séance 4&5 : Design Patterns

Exercice 1 – Iterator Pattern

Écrire un itérateur donnant la suite des nombres i^2 pour i=1,2,3…n, où n est un paramètre, et illustrer son usage par un exemple.

Exercice 2 – Template method

Écrire plusieurs classes *Logger* qui enregistrent des messages de type String. Chaque entrée du log comprend un timestamp (JJ/MM/AAAA hh:mm:ss) suivi du message.

Le *Logger* ajoute une double, respectivement simple, ligne de séparation si la date, respectivement l'heure (hh), a changé par rapport à la précédente entrée.

Les classes sont :

- FileLogger
- ConsoleLogger

Illustrer par un exemple (main).

Exercice 3 – Observer Pattern

Implémenter les classes suivantes :

- *WeatherServer* : capte la température et l'ensoleillement ("tempête", "pluie", "nuageux", "nuages épars", "éclaircies", "soleil"). On simulera ce serveur avec des méthodes get et set pour changer l'état 'capté'.
- *SimpleWeatherClient* :
  - s'enregistre auprès du *WeatherServer*
  - imprime sur la console la température et l'ensoleillement chaque fois qu'il y a un changement.
- *SuperWeatherClient* :
  - s'enregistre auprès du server
  - imprime sur la console l'heure, la date, la température et l'ensoleillement chaque fois qu'il y a un changement, accompagné d'un smiley :-) s'il fait beau, :-(dans les autres cas.

Chaque client (chaque *SimpleWeatherClient* et chaque *SuperWeatherClient*) porte un nom. Le message qu'il imprime est toujours précédé de son nom.

Exemple :

```
simpleclient1 : 25°, soleil
monSuperClient34 : 25 apr 2008, 10h30: 25°, soleil, :-)
```

Illustrer par un exemple (main).

Exercice 4 – Composite pattern

Écrire des classes permettant de représenter des expressions et de les évaluer. Une expression est de la forme "expression + opérateur + expression" ou "nombre". Les nombres sont des nombres entiers, les opérateurs sont l'addition, soustraction, multiplication et division.

Ajouter une méthode permettant d'obtenir un itérateur énumérant l'expression de gauche à droite.

Exercice 5 – Command Pattern

Voici le code d'une application console permettant de

- Faire circuler un des promeneurs
- Ajouter de nouveaux promeneurs

1. Modifier ce code pour permettre des "undos" multiples.
2. Modifier ce code pour permettre des "redos" multiples.
3. Modifier ce code pour ajouter une commande imprimant toutes les commandes effectuées jusqu'à present (et non annulées) avec un timestamp.
4. Ajouter une commande pour retirer tous les promeneurs