

Conception à base de patrons I

1 - Objectifs

Ce laboratoire permettra aux étudiants de se familiariser avec l'implémentation des patrons de conception « Composite » et « Template Method ». Cette implémentation est effectuée à l'aide du logiciel Visual Studio et le langage C++ sera utilisé tout au long du processus de développement. Un cadriciel est fourni, qui doit être complété afin d'obtenir le résultat final souhaité.

2 - Patron Composite (50 points)

La théière PolyInfusion est constituée d'un ou de plusieurs circuits hydrauliques et mécaniques. Les circuits hydrauliques permettent de stocker, chauffer et acheminer les liquides jusqu'au filtre lors de l'infusion d'un thé. Les circuits mécaniques, quant à eux, permettent de stocker et d'acheminer les solides sous forme de poudre jusqu'au filtre, où les liquides et solides sont mis en contact pour préparer le thé. La version de base de la théière PolyInfusion inclut un circuit hydraulique pour l'eau chaude et un circuit mécanique pour le thé. La version de luxe de la théière ajoute un second circuit hydraulique pour le lait, et un second circuit mécanique pour le sucre.

Implémentation

On vous demande de compléter les fichiers suivants :

- `CircuitLiqComposite.cpp`
- `CircuitSolComposite.cpp`

afin que les tests programmés dans la méthode

`Test_TP4::executeCompositeTest()` s'exécutent avec succès.

Questions à répondre

- 1) Identifiez les points suivants :
 - a) L'intention du patron Composite.
 - b) La structure des classes réelles qui participent au patron ainsi que leurs rôles (faite un diagramme de classes avec Enterprise Architect pour chaque instance du patron composite, ajouter des notes en UML pour indiquer les rôles, et exportez le tout en pdf).
- 2) Dans l'implémentation actuelle du système PolyInfusion, quel(s) objet(s) ou classes(s) est/sont responsable(s) de la création et de la destruction des composantes de chaque arbre.

3 - Patron Template Method (50 points)

Les deux types de théières PolyInfusion offrent des fonctionnalités de diagnostic et de nettoyage automatique qui peuvent être lancées à partir de la classe de base *MachineAbs*. Ces fonctionnalités doivent tenir compte du nombre de circuits présents dans chaque type de machine afin d'effectuer des opérations uniquement sur les systèmes installés dans chaque machine.

Implémentation

On vous demande de compléter le fichier suivant :

- `MachineAbs.cpp`

afin que les tests programmés dans la méthode

`Test_TP4::executeTemplateMethodTest()` s'exécutent avec succès.

Questions à répondre

- 1) Identifiez les points suivants :
 - a) L'intention du patron Template Method.
 - b) La structure des classes réelles qui participent au patron ainsi que leurs rôles (faite un diagramme de classes avec Enterprise Architect, ajouter des notes en UML pour indiquer les rôles, et exportez le tout en pdf).
- 2) Selon vous, dans la définition de la classe *ElmCircuitLiquide*, quel est l'avantage de définir la méthode `nettoyer()` comme méthode pure virtuelle et de définir la méthode `removeTartre()` comme une méthode séparée,

plutôt que de définir l'implémentation de la méthode `nettoyer()` pour qu'elle contienne directement le code fourni dans la méthode `removeTartre()` ?

- 3) Selon vous, la méthode `MachineAbs::infuserThe()` est-elle une instantiation du patron de conception Template Method ? Justifiez votre réponse.

4 – À remettre

- 1) Une archive `LOG2410_TP4_matricule1_matricule2.zip` qui contient les éléments suivants :
 - a) Le fichier `ReponsesAuxQuestions.pdf` avec la réponse aux questions 2.1a), 2.2), 3.1a), 3.2) et 3.3)
 - b) Le fichier `DiagrammeDeClasses_Composite.pdf` pour le diagramme de classes des deux patrons composite de la question 2.1b)
 - c) Le fichier `DiagrammeDeClasses_TemplateMethode.pdf` pour le diagramme de classe de la question 3.1b).
 - d) Les trois fichiers C++ que vous avez modifiés, c'est-à-dire, `CircuitLiqComposite.cpp`, `CircuitSolComposite.cpp` et `MachineAbs.cpp`. Vous ne pouvez pas modifier les autres fichiers `.h` et `.cpp`. Le correcteur va insérer vos trois fichiers dans le code, et ça doit compiler et s'exécuter.