Gabriel Bourgault - 1794069
Giuseppe La Barbera - 1799919
Remise du 15 Novembre 2016

Réponses aux questions du TP4 - LOG2410

2) Patron Composite

2.1 - a) **Intention :** Traiter les objets individuels (*leafs*) et les objets multiples, composés récursivement, de façon uniforme. Autrement dit, un objet dans une classe donnée peut contenir un objet qui provient de cette même classe. Un exemple digne du patron composite pourrait être le répertoire de fichier (conçu en arbres) du système d'exploitation Windows.

2.1 - b) Voir le document PDF ci-joint : DiagrammeDeClasses_Composite.pdf

2.2) **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.**
Les classes composite (CircuitLiqComposite et CircuitSolComposite) sont responsables de l'ajout et du retrait des éléments.
Les classes composites sont créés, puis les fils (leafs) y sont ajoutés. La destruction, elle, se fait par l'appel du destructeur. Par défaut, les destructeurs des feuilles (leafs) sont appelés et, ensuite, ceux des sous-composite. Le tout se fait de façon récursive jusqu'à ce que l'arbre soit parcouru au complet (du bas vers le haut).

3) Patron Template Method

3.1 - a) **Intention :** Définir le squelette d'un algorithme dans une opération (une classe abstraite), et laisser les sous-classes définir certaines étapes.

3.1 - b) Voir le document PDF ci-joint : DiagrammeDeClasses_TemplateMethode.pdf

3.2) **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()?**

Pour la méthode nettoyer(), on force toutes les classes dérivées à implémenter cette fonction. Cela est essentiel, car on n'a pas de comportement générique à appliquer pour cette méthode, puisqu'elle dépend des attributs de l'objet. Bref, on augmente l'abstraction en l'utilisant comme méthode virtuelle pure.

Pour ce qui est de removeTartre() par contre, il est possible de l'implémenter directement dans la classe abstraite, car il s'agit d'un comportement universel peu importe s'il s'agit d'une feuille ou d'un composite.

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

Non, il ne s'agit pas du patron de conception Template Method, car la fonction infuserThe() de MachineAbs est virtuelle pure et n'est pas implémentée dans cette classe. Elle doit donc être implémentée dans ses classes dérivées, soit MachineBase et MachineLuxe. Ainsi, on ne parle pas ici d'une fonction qui implémente le code commun puis qui appelle les sous-classes pour effectuer les comportements particuliers, tel qu'il est défini dans le patron Template Method.