Questions TP4 LOG2410

## Patron Composite

Intention : Traiter les objets individuels et les objets multiples, composés récursivement, de façon uniforme.

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.

## Patron Template Method

Intention : Définir le squelette d’un algorithme dans une opération, et laisser les sous-classes définir certaines étapes.

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.

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. 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, dont MachineAbs. 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.