

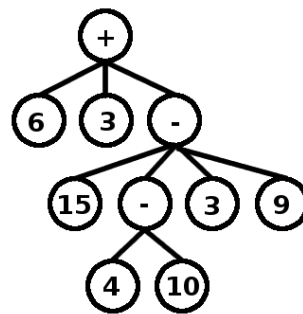
## TD n°9 : Autres patterns

### Exercice 1 (Arithmétique simple)

Une expression arithmétique est encodée sous forme d'arbre. Nous ne considérerons que les expressions composées des opérations  $+$  et  $-$  sur les entiers naturels.

Les feuilles de l'arbre contiennent des valeurs dans  $\mathbb{N}$ . Les noeuds correspondront aux opérations. Ainsi, l'expression suivante sera représenté par l'arbre ci-dessous.

$$6 + 3 + (15 - (4 - 10) - 3 - 9) = 18$$



1. Proposer une modélisation UML de l'arbre en utilisant le pattern Composite.
2. On souhaite pouvoir évaluer une même expression de plusieurs manières différentes ( par exemple, "a-b" pourrait correspondre à  $-(a+b)$  ). Proposer une modélisation UML des évaluations d'une expression à l'aide du pattern Visitor.

### Exercice 2 (Achat et vente d'actions)

On souhaite écrire un programme exécutant des **Ordre** d'achat et de vente d'actions. Tous les ordres du **Client** concernant une **Action** sont envoyés à un **Agent** s'occupant de les exécuter. La classe **Agent** garde une file interne d'ordres à envoyer au système. Par exemple, on peut en effet imaginer que le système ferme durant le week-end, donc l'agent enfilera les ordres qu'il aura reçu le samedi et dimanche pour ne les envoyer que le lundi.

Il y a du pattern Command là-dessous ou je ne m'y connais pas. Zou, diagramme UML.

### Exercice 3 (Le visualiseur de photos)

Vous écrivez un visualiseur d'images qui est capable d'afficher une liste de photos de canards en HD se trouvant dans le même répertoire. Au lancement, le programme écrit en mémoire une référence pour chaque photo afin d'y avoir accès plus rapidement. Cependant il n'est pas utile de charger en mémoire toutes les photos HD au démarrage! Cela risquerait de le ralentir considérablement.

On souhaite donc ne charger en mémoire que la photo que l'utilisateur regarde à chaque moment, et faire une référence à chaque photo en utilisant un Virtual Proxy.

UML, guys.