Qualité de Développement – TP8

L'objectif de ce TP est de réaliser sous Modelio un diagramme de classes de la version 1 de l'évaluateur d'expressions étudié dans le TD2 ainsi que le codage java correspondant, en privilégiant une approche qui facilite l'évolution vers les versions 2 et 3.

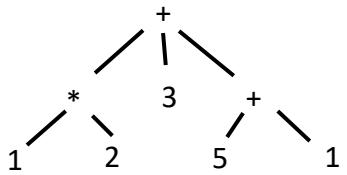
Un développement de la version 1 de l'évaluateur selon une approche mono-bloc est fourni sur Moodle (diagramme de classes + code Java). L'algorithme d'analyse et d'évaluation des expressions étudié en TD est codé dans une seule classe nommée Evaluateur sous la forme d'une seule méthode nommée analyserEtEvaluer().

Travail à faire :

1. Télécharger la version mono-bloc de l'évaluateur à partir de Moodle. Examiner le diagramme de classes ainsi que le code Java puis tester son fonctionnement après avoir copié le code dans un nouveau projet Java.

2. Créer un nouveau projet Modelio et concevoir un nouveau diagramme de classes de manière à obtenir une application ouverte aux modifications (au sens du principe SOLID Open/Closed). La conception doit être guidée par les points suivant, discutés en TD :

- Séparer l'analyse de l'évaluation. D'abord analyser les expressions, puis les évaluer si elles sont correctes.
- Distinguer l'analyse selon le type des expressions : analyse d'une constante, analyse d'une expression naire.
- Distinguer l'évaluation selon le type des expressions : évaluation d'une constante, évaluation d'une expression n-aire.
- Pour pouvoir évaluer une expression correcte déjà analysée, il faut pouvoir représenter l'expression par une structure exploitable par l'évaluation (sans avoir à réanalyser l'expression). Cette structure doit permettre d'identifier : le type d'une expression (constante ou expression n-aire) ; la valeur d'une constante ; l'opérateur et les opérandes d'une expression n-aire.
- Utiliser une structure arborescente pour représenter les expressions correctes, dans laquelle les feuilles correspondent à constantes et les autres nœuds reprsentent des expressions naires, comme le montre la figure ci-après qui représente l'expression "(+ (* 10 2) 3 (+ 5 10))".



3. Générer sous Modelio le squelette de code Java correspondant.
4. Créer un nouveau projet Java sous Netbeans et y intégrer le code généré sous Modelio. Retoucher le code de manière à éliminer les erreurs signalées par le compilateur (ajouter des « bouchons » si nécessaire).
5. Compléter le code de manière à finaliser la version 1 de l'évaluateur et tester son bon fonctionnement.