

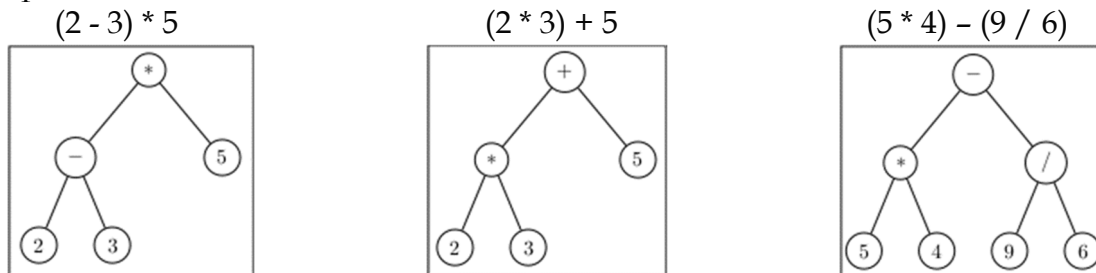
## Devoir - Expressions Arithmétiques

### A rendre le 9 mai avant 23 heures

On considère les expressions arithmétiques composées des quatre opérations mathématiques usuelles ( $-$ ,  $+$ ,  $/$  et  $*$ ). Leurs opérateurs sont binaires, c'est-à-dire qu'ils ont toujours deux opérandes. Les exemples  $+(2+3)*4-1$  et  $*1$  ne correspondent donc pas à des expressions.

En respectant les priorités des opérateurs et des parenthèses, on peut représenter graphiquement une expression arithmétique sous forme d'une structure arborescente (appelée *arbre binaire*)

Par exemple :



On souhaite programmer une petite application qui, à partir d'une expression arithmétique donnée, calcule la valeur du résultat de cette expression. Par exemple pour chacun des trois exemples donnés ci-dessus, le résultat devrait être -5, 11 et 18.5 respectivement.

D'une façon générale, une **Expression** représente les expressions arithmétiques, et pour chacune d'elles on peut utiliser une méthode **double calculerValeur()**. Cette méthode retournera le résultat du calcul de l'expression arithmétique.

### Partie 1

Vous écrirez les classes **Nombre** puis **Operation** qui proposeront aussi cette même méthode **double calculerValeur()**.

La classe **Nombre** correspond à un nombre simple. Lorsqu'on construit un **Nombre**, on fournit une valeur de type double, et qui sera la valeur de cette **Expression**.

Une **Operation** est composée d'un caractère représentant l'opérateur, et de ses deux opérandes, qui sont en fait des **Expressions**. Ces trois informations sont transmises au constructeur pour être conservées dans l'**Operation**.

Ecrivez le code nécessaire de ces deux classes. Pour chacune d'elles, vous implémentez la méthode **calculerValeur()** : dans le cas d'un **Nombre**, elle en retourne la valeur, et dans le cas d'une **Opération**, elle retourne le résultat d'un calcul, qui dépend de l'opération et des chacune des **Expression**, opérande gauche et droite.

Ecrivez une classe **AppExpression** dans laquelle vous construirez des expressions arithmétiques et calculer leurs résultats :

```
Expression monExpr;  
// pour construire une Expression, on crée d'abord les Nombres,  
// puis les Expressions utilisant ces nombres, etc.  
double resultat = monExpr.calculerValeur(); // évalue l'expression  
System.out.println(resultat);
```

Remarques :

- Les seuls caractères possibles pour les opérateurs sont :  $+$ ,  $-$ ,  $*$  et  $/$
- Lorsque le calcul implique une division par zéro, vous pouvez vous contenter de lever une exception correspondante, en utilisant l'instruction

```
throw new ArithmeticException("Division par zéro");
```

## Partie 2

On souhaite maintenant pouvoir afficher une expression arithmétique en format texte à partir des **Expression** construites. Par exemple le texte de la première expression devrait être  $((2-3)*5)$ . Inspirez-vous de la solution à la question précédente pour redéfinir la méthode **toString()** dans **Operation** et **Nombre** afin de permettre l'affichage d'une expression arithmétique. Utilisez dans la classe AppExpression, cela ressemblera à ceci :

```
Expression monExpr; // = new...  
System.out.println(monExpr); // appelle de toString() pour monExpr
```

Remarques :

- Seule l'expression arithmétique devra être retournée par la méthode **toString()**, pas le résultat de calcul, pas de texte supplémentaire...
- Pensez à respecter les priorités des opérateurs et donc à mettre les parenthèses aux bons endroits. Pour vous simplifier la tâche, vous pouvez parenthéser toutes les expressions, y compris celles pour lesquelles habituellement on n'ajoute pas les parenthèses, par exemple :  $(2+3)$  ou  $((2*3)+5)$  ou  $(8*9)$ .