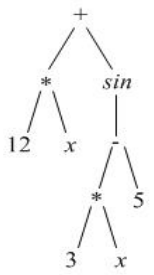
CS312 : « Modéliser les Expressions Mathématiques»
Thème 3 : Conception objet, Classe abstraites, Interfaces

Dans cet exercice on vous demande de définir un ensemble de classes pour représenter des fonctions d'une variable formées avec des constantes, des occurrences de la variable x, les quatre opérations arithmétiques +, –, ×, / et des appels de quelques fonctions convenues comme sin, cos, exp, log, etc… Par exemple : 12x + sin(3x – 5). Dans un programme, une expression comme celle-là peut être efficacement représentée par une structure arborescente, organisée comme le montre la figure ci-contre, faite de feuilles (les constantes et les variables), de nœuds à deux « descendants » (les opérateurs binaires) et de nœuds à un descendant (les fonctions d'une variable, ou *opérateurs unaires*). On souhaite définir les classes représentant les nœuds d'un tel arbre.



On modélise toutes les sortes de nœuds de notre structure arborescente par une interface *Expression*, représentant ce qu'ont en commun toutes les expressions arithmétiques. Elle se compose d'une seule méthode *valeur* qui renvoie la valeur de l'expression pour un x donné:

```
public interface Expression {
   double valeur(double x);
}
```

Exercice 1 : Conception

1) Dessinez le diagramme d'héritage des classes *Constante*, *Variable* (représentant *x*), *OperationBinaire*, *OperationUnaire*, *Addition*, *Soustraction*, *Multiplication*, *Division*, *Sin*, *Cos*, *Log* et *Exp*.
2) Précisez pour chaque classe s'il s'agit d'une classe concrète ou abstraite
3) Donnez les attributs et signatures des méthodes de chaque classe

Exercice 2 : Implémentation des classes

a) Implémenter les classes précédentes en Java.
b) Ecrire une redéfinition intéressante de la méthode *toString()* pour chaque classe.

Exercice 3 : Utilisation des classes

a) Ecrire le code correspondant à la création et à l'affichage de l'expression f(x) = 2*sin(x) + 3*cos(x).
b) Ecrire un code affichant f(x) pour x = {0, 0.5, 1, 1.5, 2, 2.5, 3, 3.5}.

Exercice 4 : Un évaluateur d'expressions monovariables en ligne de commande

En utilisant les classes précédentes, on souhaite implémenter un programme d'évaluation d'expressions monovariables en ligne de commande pouvant s'utiliser comme ceci :

```
> java Evaluer 2 + 3 * ( sin ( x ) / log ( x ) - 4 )
valeur de x ? 10
      résultat: -11.6320633
valeur de x ?
...
```

On suppose que l'on dispose d'une fonction `Expression Strings2Exp(String[] s)` renvoyant une expression à partir d'un tableau de chaînes de caractères représentant des chiffres, des opérateurs ou des parenthèses. Dans un premier temps, on supposera que toutes les expressions données par les utilisateurs sont syntaxiquement correctes.

a) Ecrire la classe exécutable *Evaluer* correspondant au comportement décrit ci-dessus.
b) Ecrire la fonction *Strings2Exp*. On pourra s'aider de la récursivité, et d'une fonction auxiliaire `int prioMin(String[] s)` renvoyant l'indexe d'une chaîne de caractère correspondant à un opérateur de priorité minimale dans le tableau *s*.
   Rappel : l'ordre croissant de priorité des opérateurs est : +,-,×, / puis les opérateurs unaires. Les opérateurs à l'intérieur des parenthèses ont toujours une priorité supérieure. Notons que *(expr) = expr*.
c) Ecrire la fonction *prioMin* pour terminer l'application.
d) Evaluer les différentes erreurs syntaxiques possibles, et gérer ces erreurs.