**PROJET EXPRESSION (1ERE PARTIE)**

**SUJET :**

Ecrire un ensemble de classes C++ permettant de traiter des expressions mathématiques simples de la forme :

(5/2 + 4) \* (-3 - 20)

Dans cette première partie, les expressions que l'on souhaite traiter sont composées uniquement :

- d'opérateurs arithmétiques : addition, soustraction, multiplication, division

- de constantes flottantes.

Cet ensemble de classes doit permettre de :

- créer en mémoire ("en dur dans le code") une expression quelconque,

- de calculer la valeur de cette expression,

- d’afficher sur un flot de sortie (classe ostream) une représentation de l'expression (sous la forme d'une chaîne de caractères) en notation classique ou polonaise inversée.

**AIDE :**

Ce projet met en œuvre un arbre syntaxique qui représente l'expression. Exemple :

(5/2 + 4) \* (-3 - 20)

\*

/

+

4

20

2

5

-3

Notation classique : (5/2 + 4) \* (-3 - 20)

Notation polonaise inversée : 5 2 / 4 + -3 20 - \*

Valeur de l’expression : -149.5

**les etapes d'une c.o.o.**

|  |  |
| --- | --- |
|  | Déterminer les classes   * noms + bibliothèques de classes réutilisables |
|  | Spécifier les méthodes et attributs |
|  | Spécifier les dépendances   * relations d'héritage, composition, utilisation |
|  | Spécifier les interfaces   * privé, protégé, public * paramètres, valeur de retour |
|  | Tester les principaux cas d'utilisation   * (exemple ici : création d'une expression, impression, évaluation) |
|  | Restructurer les classes   * retour aux phases 1 à 5 ==> cycle |
|  | Utilisation de librairies   * listes, entrées-sorties, ... |

**comment determiner les classes ?**

A partir du texte des spécifications de besoin :

. Isoler et souligner les noms

fréquents ==> noms des classes

peu fréquents ==> noms d'attributs

. Isoler et souligner les verbes

==> méthodes des classes

**ATTENTION** :

Un nom peut représenter une action, donc une méthode