Etapes

1. Récup input user.
2. Check user input
   1. Liste des éléments autorisés (interdire les parenthèses dans un 1er temps).
   2. Double caractère non autorisé (ex : \*\*, ++, …)
3. Lexer = une fonction qui prend l’user input et qui renvoie un containeur ( 🡺 création containeur : template) des briques de la string.
4. Parser = une fonction qui utilisent des **strings** pour déterminer les prios.
5. Evaluator = classe des méthodes add, minus, div, mult.
6. Mettre des emojis dans l’output.

Main

Cout 🡺 « demande à l’utilisateur de rentrer son calcul »

Pour quitter écrire « q ».

While(condition){

Cin >> string1

Check\_input(string1)

} ;

Cout 🡺 « message de sortie.Merci »

**Détails des étapes**

Checker

* *Check\_input()*

Boucle for sur un string. (2.a) et vérifier les blocs (doublement des opérateur 2.b). Liste de check dans le private avec accesseur et éventuellement mutateur.

Lexer

* Fonction avec une boucle for qui :
  + Lit la chaine jusqu’au prochain symbole de calcul.
  + A ce moment la méthode de la classe string qui renvoie un morceau (=le nombre) que l’on stocke dans notre containeur.
  + Stop avec length de la string.

Parser

* Class bloc 🡨 class opérateur, class nombre.

Class bloc{

Attribute : int nature\_ ;bloc\* ptr\_(vers le haut), T valeur\_(classe T), ptr\_bas1\_ ; ptr\_bas2\_

Methodes :

Set\_nature

Get\_nature

Set\_valeur

Get\_valeur

Set\_ptr

}

***Fichiers***

-checker

-lexer

-parser

-Container

-calculateur