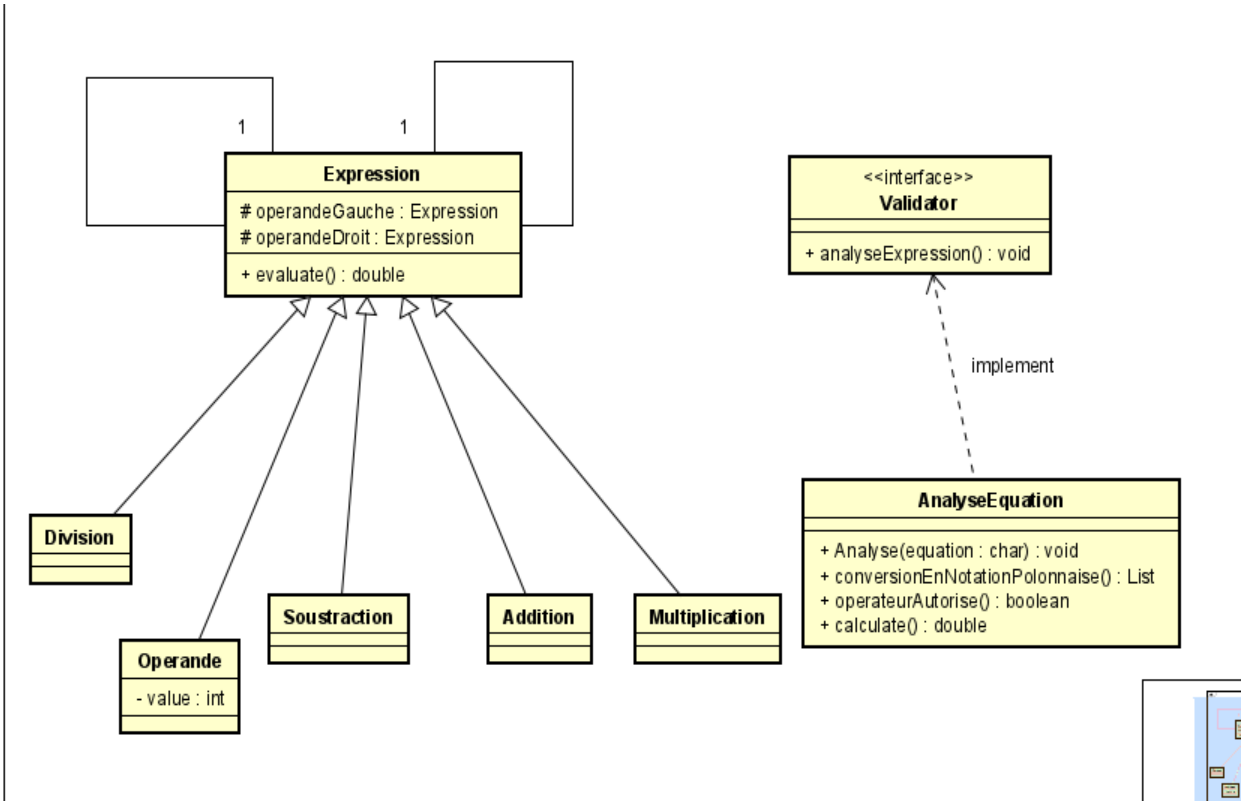
PARTICIPANTS :
SILUE A DAMA DOFARA
GRAMBOUTE COULIBALY

DIAGRAMME DE CLASSE DU PROJET



STRUCTURE GLOBALE DU PROJET

Nous avons scindé le 3 parties comme il a été conseillé de le faire :

- Analyse de l'équation en gérant tous les cas susceptibles de lever une exception. Voici une liste des cas que nous avons pris en compte :
  `++8, +8*/7, (2+3, +8*, )(0+1, )127(, ...+5, 2  2  3, a+3*5`
- Conversion de l'équation sous forme polonaise inversée : prendre en entrée une expression arithmétique et la convertir en expression post-fixée.
  Pour ce faire nous avons lu et réadapté l'algorithme de `shunting yard`
- Calcule de la valeur de l'expression en prenant en paramètre une expression déjà sous forme polonaise.
  Nous utilisons une pile et faisons les calculs au fur et à mesure.

PRINCIPAUX ALGORITHME UTILISES

- Algorithme de shnting yard pour convertir une expression sous forme polonaise
- conversionEnNotationPolonaise de validation d'une équation qui vérifie la structure générale de l'équation, en gérant les parenthèses, les opérateurs et les lettre.
- Des algorithmes intermédiaires pour rendre le code plus clair, comme operateurAutorise, AppliquerOperation

ETAT DE CE QUI MARCHE ET CE QUI MARCHE PAS

- CE QUI MARCHE
  Nous avons pu tester l'ensemble des cas de figure de calcul auxquels nous avons réfléchi et avons vérifié les résultats.
- Nous avons eu beaucoup de mal à transformer l'équation sous forme de chaine en Objet à partir des classes

DIFFICULTES

- COMPRENDRE CE QUI EST DEMANDE.

Comprendre précisément ce qui est demandé dans l'énoncé, notamment les opérateurs à prendre en charge, les priorités d'opérateurs, et les restrictions sur les expressions valides.

- COMMENT CONCEVOIR L'ARCHTECTURE DU PROGRAMME.

Pour Concevoir une architecture intéressante et modulaire a été compliquer pour une des étudiantes du binôme paracerque pas d'expérience préalable. Nous avons réfléchi à la manière de structurer le code pour faciliter la lisibilité, la maintenance et l'évolutivité du programme.

- TRAITEMENT DES EXPRESSIONS MATHEMATIQUES

Analyser et évaluer des expressions mathématiques complexes a été difficile, surtout lorsqu'il faut gérer les parenthèses, les priorités des opérateurs et les opérations dans un exercice.

On a essayé de convertir les équations sous forme polonaise inversée histoire de voir si le code marche bien et si on a les résultats qu'il faut, ensuite on calcules la valeur.

- GESTION DES ERREURS

Savoir détecter et gérer les erreurs telles que les divisions par zéro, les expressions invalides ou mal formées, et les exceptions peut être délicat, surtout pour un débutant.

- TESTS JUNIT

Écrire des cas de test complets et exhaustifs pour couvrir tous les scénarios possibles et vérifier que le programme fonctionne correctement dans toutes les situations peut être un défi, surtout pour les débutants en programmation.
Nous avons testé :
les fonctions de conversions d'expressions simple en expression sous forme polonaise, les fonctions de validation de l'expression et les fonctions de calcul.

- CONVERTIR UNE EXPRESSION SOUS FORME POLONAISE.

Nous avons eu des soucis avec le fait de transformer une expression sous forme polonaise.