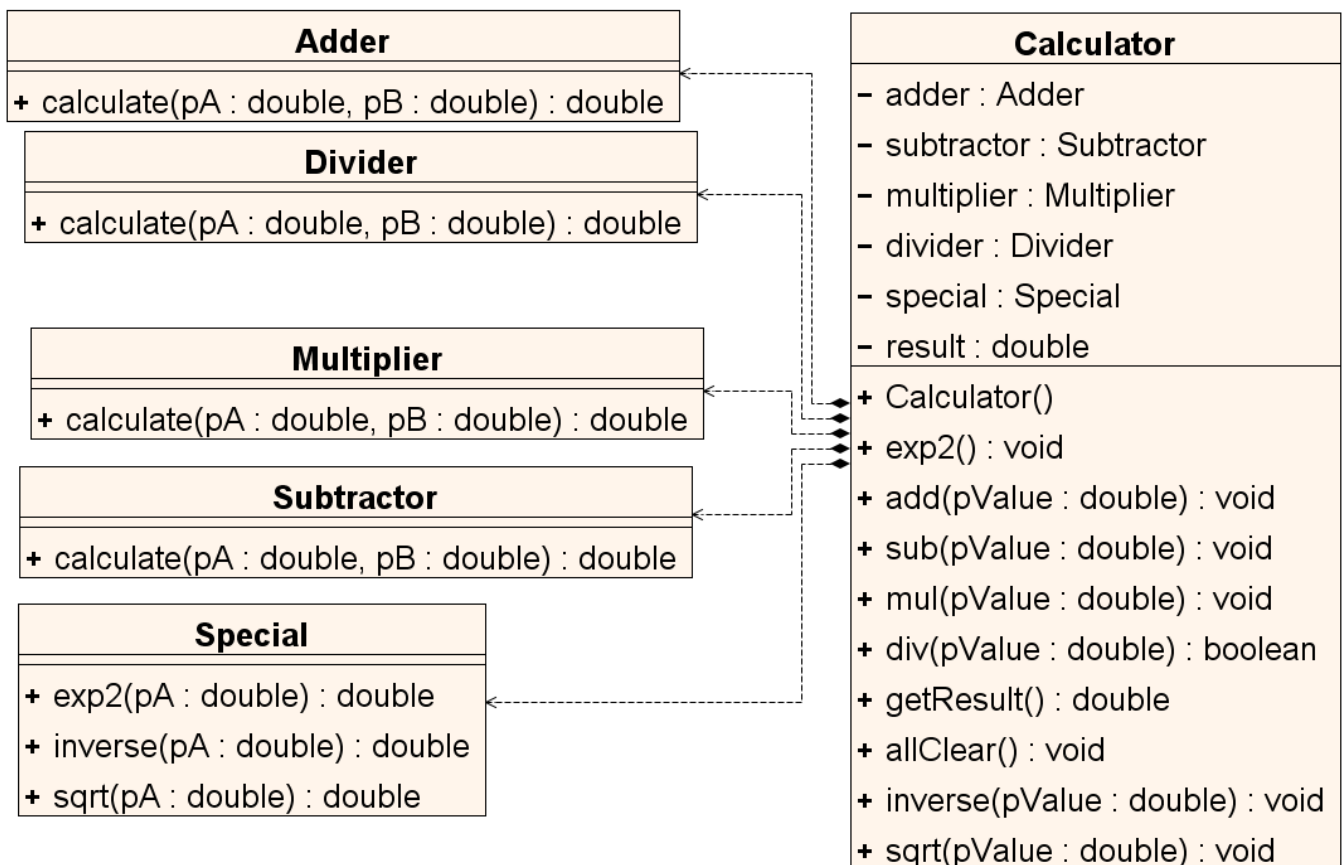


# Exercice : Calculator with multiple objects

Vous devez implémenter le projet **Calculator with multiple objects** qui est capable d'effectuer les quatre calculs de base (+ - / \*), la fonction carrée, la fonction racine carrée et la fonction inverse.

La classe à partir de laquelle vous lancerez les méthodes de calcul est la classe **Calculator**. Les méthodes **add**, **sub**, **div** ... de la classe **Calculator** n'effectuent pas les calculs mais font appels aux classes **Adder**, **Subtractor**, **Divider**, **Multiplier** et **Special** pour effectuer les calculs nécessaires. L'attribut **result** contient le résultat du dernier calcul. Le calcul qui suit utilise **result** et la valeur spécifiée par paramètre.

Programmez le projet comme décrit par la suite tout en respectant le diagramme UML suivant:



## Classe Adder

La méthode `calculate(...)` retourne la somme de `pA` et de `pB`.

## Classe Subtractor

La méthode `calculate(...)` retourne la différence de `pA` et de `pB`.

## Classe Divider

La méthode `calculate(...)` retourne la division de `pA` par `pB`.

## Classe Multiplier

La méthode `calculate(...)` retourne la multiplication de `pA` par `pB`.

## Classe Special

1. La méthode `exp2(...)` retourne le carré de `pA`.
2. La méthode `inverse(...)` retourne l'inverse de `pA`.
3. La méthode `sqrtd(...)` retourne la racine carrée de `pA`.

## Classe Calculator

1. Attributs: voir diagramme UML
2. Le constructeur crée cinq instances de la classe respective (objets) et initialise `result` à la valeur `0.0`.
3. La méthode `add(...)` ajoute la valeur `pValue` à `result` en utilisant la référence `adder`. Pour le calcul, vous devez obligatoirement utiliser l'instance de la classe `Adder`. Vous ne devez, en aucun cas, effectuer le calcul dans la classe `Calculator`.
4. La méthode `sub(...)` déduit la valeur `pValue` de `result` en utilisant la référence `subtractor`. Pour le calcul, vous devez obligatoirement utiliser l'instance de la classe `Subtractor`. Vous ne devez, en aucun cas, effectuer le calcul dans la classe `Calculator`.
5. La méthode `mul(...)` multiplie la valeur `result` par `pValue` en utilisant la référence `multiplier`. Pour le calcul, vous devez obligatoirement utiliser l'instance de la classe `Multiplier`. Vous ne devez, en aucun cas, effectuer le calcul dans la classe `Calculator`.
6. La méthode `div(...)` divise la valeur `result` par `pValue` en utilisant la référence `divider`. Pour le calcul, vous devez obligatoirement utiliser l'instance de la classe `Divider`. Vous ne devez, en aucun cas, effectuer le calcul dans la classe `Calculator`. Cette méthode retourne la valeur booléenne `true` si le calcul a pu être effectué et `false` si `pValue < 1E-99`. Dans ce cas, le message « Erreur, division par 0 » est affiché à la console.
7. La méthode `exp2(...)` programme `result` au carré de `result`.
8. La méthode `inverse(...)` programme `result` à `1/result`.
9. La méthode `sqrtd(...)` enregistre la racine carrée de `result` dans `result`.
10. La méthode `allClear()` réinitialise `result` à la valeur `0.0`.
11. Programmez l'accesseur pour l'attribut `result`.