

# Exercice Fraction calculator

Dans cet exercice, vous allez programmer une calculatrice capable d'effectuer des calculs sur des fractions.

## 1 Classe Fraction

Programmez la classe **Fraction** qui représente une fraction mathématique en suivant les instructions ci-dessous tout en respectant le diagramme UML.

- a) Le constructeur initialise les deux attributs à partir de ses paramètres respectifs.

**Note** : on suppose (sans vérification) que le dénominateur est différent de zéro, sinon les résultats des calculs sont imprévisibles.

- b) Programmez les accesseurs.
- c) La méthode `calcDecimal()` calcule et retourne la valeur décimale de la fraction.
- d) La méthode `public String toString()` retourne un texte de la forme : *numérateur / dénominateur (valeur décimale)*. Si le dénominateur est égal à 1, alors celui-ci n'est pas affiché.

Exemples:     8/16   (0.5)

                 125/100   (1.25)

                 23   (23.0)

- e) La méthode `public int gcd(int pA, int pB)` calcule et retourne le **plus grand commun diviseur** (PGCD) de **pA** et de **pB** (calcul selon la méthode d'Euclide).
- f) La méthode `public void simplify()` simplifie la fraction en utilisant la méthode ci-dessus.
- g) Les 4 méthodes **add**, **subtract**, **multiply**, **divide** (→ voir plus bas) permettent d'effectuer des calculs entre deux objets de la classe **Fraction**. On peut ajouter, soustraire, diviser et multiplier la fraction actuelle par une deuxième fraction passée comme paramètre. Après le calcul, la fraction actuelle est à simplifier à l'aide de la méthode **simplify**.

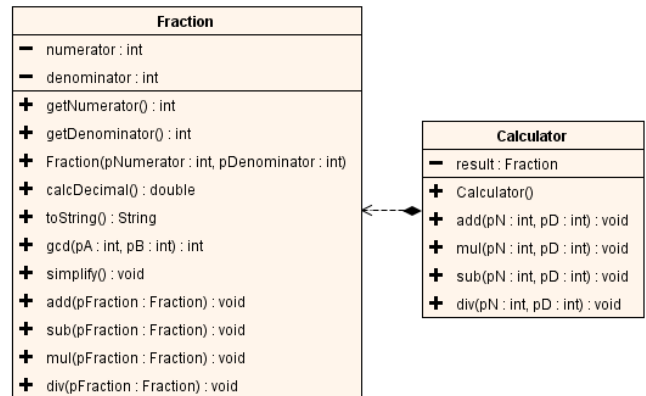
Dans cet exercice, on va **passer une référence vers un objet comme paramètre**, de la même façon qu'on peut passer des nombres ou des textes comme paramètres.

```
public void add(Fraction pFraction)      : ajouter et simplifier

public void subtract(Fraction pFraction) : soustraire et simplifier

public void multiply(Fraction pFraction) : multiplier et simplifier

public void divide(Fraction pFraction)   : diviser et simplifier
```



## 2 Classe Calculator

Comme expliqué dans la partie **Classe Fraction**, pour pouvoir effectuer un des quatre opérations de base, il nous faut une deuxième fraction dont la référence, qui pointe vers celle-ci, est passée comme paramètre. Dans la classe **Calculator**, vous devez programmer quatre méthodes dans lesquelles vous devez créer temporairement cette deuxième fraction.

- a) L'attribut **result** contient une référence vers le dernier résultat calculé.
- b) Le constructeur initialise l'attribut **result** avec la fraction  $\frac{0}{1}$ .
- c) La méthode **public void add(int pN, pD)** effectue deux étapes :
  1. Création d'une instance (création d'un objet) de la classe **Fraction** en passant les deux paramètres **pN** et **pD** à l'aide d'une variable locale.
  2. Appel à la méthode **add()** de l'objet référencié par **result** en passant la référence vers l'objet créé au point 1.
- d) Les méthodes **mul**, **sub** et **div** sont implémentées de façon similaire que la méthode **add** du point c).
- e) Testez votre programme sous Unimozer et utilisez la fonction **inspect** et **monitor** pour voir la fraction référenciée par l'attribut **result**.

