❑ **Activité 3** : *POO et calcul sur les fractions*

Le but de l'activité est de mettre en oeuvre en Python un module permettant d'effectuer des calculs (additions, soustractions, multiplications, divisions) et des manipulations (simplification) en utilisant le paradigme de programmation objet.

1. L'objet fraction

   On modélise une fraction par ses attributs numérateur et dénominateur, le dénominateur doit être non nul, sinon c'est une division par zéro.
   a) Créer la classe `Fraction` et son constructeur.
   b) Modifier le constructeur de façon à pouvoir créer une fraction à partir d'un entier, par exemple l'entier 4 correspond à la fraction $\frac{4}{1}$.
   c) Toutes les fractions ayant zéro comme numérateur sont nulles, on les représentera toutes par la fraction $\frac{0}{1}$, modifier le constructeur en ce sens.
   d) Instancier la classe fraction pour créer la représentation informatique des fractions suivantes :
   $a = \frac{3}{4}$ $b = \frac{2}{5}$ $c = -\frac{7}{12}$ $d = 3$

2. Affichage de l'objet fraction

   ⚠ Deux méthodes spéciales en Python permettent d'obtenir une représentation sous forme de chaine de caractères d'un objet, il s'agit de `__repr__` et de `__str__`, bien que des nuances existent entre ces deux méthodes, à notre niveau nous ne ferons pas de différences et on utilisera indifféremment l'une ou l'autre pour obtenir une version « chaine de caractères »de notre objet.
   a) Ecrire une méthode spéciale permettant de renvoyer la chaine de caractère `'a/b'` à partir de l'objet représentant la fraction $\frac{a}{b}$
   b) Affiner cette méthode de façon à ce qu'une fraction ayant pour dénominateur 1 affiche simplement son numérateur.
   c) Tester votre méthode sur les fractions a, b, c et d définies ci-dessus.

3. Addition de deux fractions

   La première idée est de créer une méthode `addition` qui prend en argument deux fractions et renvoie leur somme, ainsi si nous avons deux objets fractions `a` et `b`, nous calculons leur somme avec `a.addition(b)`. Cette solution n'est pas satisfaisante, en effet si `a` et `b` sont deux objets fractions, nous aimerions dans notre programme calculer leur somme en écrivant de façon plus naturelle : `a+b`. La méthode spéciale `__add__` permet en Python d'obtenir ce résultat.
   a) Ecrire la méthode `__add__`

      ⛑ On a besoin ici de deux objets fractions, en plus du traditionnel `self`, on notera `other` l'autre objet. On rappelle d'autre part que $\frac{a}{b} + \frac{c}{d} = \frac{ad+bc}{bd}$
   b) Tester votre méthode en additionnant les fractions a et b ci-dessus

      ⛑ `print(a+b)` devrait afficher `23/20`, en effet : $\frac{3}{4} + \frac{2}{5} = \frac{23}{20}$

4. Les autres opérations de base
   a) Ecrire la méthode `__sub__` (la soustraction) et la tester.
   b) Ecrire la méthode `__mul__` (la multiplication) et la tester.
   c) Ecrire la méthode `__truediv__` (la division) et la tester.
   d) On donne ci-dessous quelques calculs fractionnaires (donnés dans le passé au brevet des collèges) utiliser votre module pour effectuer ces calculs :
   $A = \frac{3}{7} + \frac{4}{21} - \frac{5}{2}$ $B = \frac{12}{5} - \frac{3}{5} \times \frac{7}{9}$ $C = 5 + \left(1 + \frac{1}{8}\right) \div \frac{3}{4}$

5. Simplifier une fraction
   a) Ecrire une fonction `pgcd` qui prend en arguments deux entiers et retourne leur plus grand commun diviseur.

      ⛑ On rappelle que le PGCD de deux entiers a et b se calcule avec l'algorithme d'Euclide :
      1. Si $b = 0$ alors l'algorithme se termine et le PGCD est a
      2. Sinon faire la division euclidienne de a, par b, on note r le reste (et on sait que $r < b$). Revenir à l'étape 1 en prenant $a = b$ et $b = r$.
   b) Ecrire une méthode `simplifie` qui simplifie la fraction passée en paramètre pour la rendre irréductible.