

Un exercice de 'ValueTypes' pour ceux que cela puisse intéresser

Les ValueTypes sont

- toujours mis sur le stack = plus rapide
- toujours copiés puisqu'il n'y a pas de référence
  - o `int a = 2`
  - o `int b = a`
  - o `a = 1`
  - o `Console.WriteLine(« b = {0} », b) ; // 2`

Une Structure **Fraction** qui implémente des operateurs et des casts

**Operateurs** (tout opérateur binaire nécessite au moins un opérant Fraction).

- Boolean : `<, <=, ==, !=, >, >=`
  - o Le deuxième opérant peut être du type `int`, `decimal`, `double` ou `string`
- Arithmétique : `+, -, *, /, ++, --`
  - o Le deuxième opérant peut être du type `int` ou `string`

## Casts

- explicite
  - o `(int) Fraction`
  - o `(double) Fraction`
  - o `(decimal) Fraction`
  - o `(string) Fraction`
- Implicite
  - o `(Fraction) int` (pour `+`, `-`, `*`, `/`)
  - o `(Fraction) string` (pour tout opérateur)

## Constructeurs

- `new Fraction (int numerator)`
- `new Fraction( int numerator, denominator)`
- `new Fraction( "-3/4" )` ou `new Fraction("22")`

## Methods

- Static
  - o `GCD(int x, int y)` : greatest common divisor, always  $\geq 0$
  - o `LCM(int x, int y)` : lowest common multiple, always  $> 0$
- d'instance
  - o `Equals(object f2)`
  - o `Reciprocal()` : returns  $1/\text{fraction}$
  - o `ToString()`