Sobrecarga de operadores: Números Racionais

Programação Orientada a Objetos

Delano Medeiros Beder

Departmento de Computação
Universidade Federal de São Carlos

13 de janeiro de 2025

Especificação

1. Definir uma classe Racional para representar números racionais
2. Durante a inicialização (construtor da classe), execute a simplificação de números racionais.

    - O construtor da classe deve ser único (ou seja, existe apenas um construtor na classe) e deve inicializar todos os atributos do objeto.

    - Além disso, os atributos do objeto não podem mais ser alterados depois da inicialização (construção) do objeto.

3. Sobrecarregar operadores para
    - somar $(+)$ números racionais,
    - subtrair $(-)$ números racionais,
    - multiplicar $(*)$ números racionais,
    - dividir $(/)$ números racionais,
    - comparar $(<, \leq, ==, !=, \geq, >)$ números racionais e
    - imprimir $(<<)$ números racionais

Exemplo: Main.cpp

```cpp
#include <iostream>
#include "Racional.h"

using namespace std;

int main() {

    Racional r1(4, 6);
    Racional r2(6, 10);
    Racional r3(2,3);
    Racional r4(3,3);

    cout << boolalpha << endl;
    cout << "r1 => " << r1 << endl;
    cout << "r2 => " << r2 << endl;
    cout << "r3 => " << r3 << endl;
    cout << "r4 => " << r4 << endl;
    cout << "(r1 + r2) => " << r1 + r2 << endl;
    cout << "(r1 - r2) => " << r1 - r2 << endl;
    cout << "(r1 * r2) => " << r1 * r2 << endl;
    cout << "(r1 / r2) => " << r1 / r2 << endl;
    cout << "(r1 < r2) => " << (r1 < r2) << endl;
    cout << "(r1 < r3) => " << (r1 < r3) << endl;
    cout << "(r1 <= r2) => " << (r1 <= r2) << endl;
    cout << "(r1 <= r3) => " << (r1 <= r3) << endl;
    cout << "(r1 == r2) => " << (r1 == r2) << endl;
    cout << "(r1 == r3) => " << (r1 == r3) << endl;
    cout << "(r1 != r2) => " << (r1 != r2) << endl;
    cout << "(r1 != r3) => " << (r1 != r3) << endl;
    cout << "(r1 >= r2) => " << (r1 >= r2) << endl;
    cout << "(r1 >= r3) => " << (r1 >= r3) << endl;
    cout << "(r1 > r2) => " << (r1 > r2) << endl;
    cout << "(r1 > r3) => " << (r1 > r3) << endl;
    cout << endl;

    return 0;
}
```

```
r1 => 2/3
r2 => 3/5
r3 => 2/3
r4 => 1
(r1 + r2) => 19/15
(r1 - r2) => 1/15
(r1 * r2) => 2/5
(r1 / r2) => 10/9
(r1 < r2) => false
(r1 < r3) => false
(r1 <= r2) => false
(r1 <= r3) => true
(r1 == r2) => false
(r1 == r3) => true
(r1 != r2) => true
(r1 != r3) => false
(r1 >= r2) => true
(r1 >= r3) => true
(r1 > r2) => true
(r1 > r3) => false
```

Números Racionais

- Os números racionais são os números que podem ser escritos na forma de fração.
- Um número racional q pode ser escrito na forma: $q = \frac{a}{b}$, onde $a \in \mathbb{Z}$, $b \in \mathbb{Z}^*$ e $q \in \mathbb{Q}$.
- Ou seja, um número racional é sempre composto por um numerador (inteiro) e um denominador (inteiro $\neq 0$).
- A representação não é única; por exemplo:

$$\frac{2}{3}, \quad \frac{4}{6} \text{ e } \quad \frac{-8}{-12}$$

representam o mesmo número racional.

Simplificação de números racionais

- Podemos simplificar um número racional encontrando um *divisor* comum ao numerador e denominador:

$$\frac{4}{6} = \frac{2 \times 2}{2 \times 3} \text{ equivalente a } \frac{2}{3}$$

Para simplificar completamente, basta dividir ambos numerador e denominador pelo seu **mdc (máximo divisor comum)**.

- Seja $c = mdc(a, b)$ então

$$\frac{a}{b} \text{ equivalente a } \frac{a \div c}{b \div c}$$

Esta transformação funciona mesmo quando o número racional já está simplificado (nesse caso c $= 1$).

Algoritmo de Euclides

- Para calcular o mdc (máximo divisor comum) podemos usar o algoritmo de Euclides. https://pt.wikipedia.org/wiki/Algoritmo_de_Euclides

```
int mdc(int a, int b) {
    int r;
    while (b != 0) {
        r = a % b;
        a = b;
        b = r;
    }
    return a;
}
```

- Observação: Implemente o construtor da classe Racional de tal forma que garanta que os números racionais são sempre simplificados. Desta forma, cada vez que criarmos um novo número racional temos a garantia que está simplificado.

Números racionais negativos

- Como representar números racionais negativos

$$\frac{-1}{2} \quad \text{vs.} \quad \frac{1}{-2}$$

- Qualquer das duas opções é correta
- Porém vamos fixar que o *denominador* deve ser sempre positivo (ou seja: o sinal está no *numerador*)
- É simples modificar o construtor da classe para garantir esta propriedade

Somar e subtrair números racionais

- Para somar números racionais temos que encontrar um denominador comum

$$\frac{a}{b} + \frac{c}{d} = \frac{a \times d}{b \times d} + \frac{b \times c}{b \times d} = \frac{a \times d + b \times c}{b \times d} \text{ (denominador comum)}$$

$$\text{Exemplo : } \frac{2}{3} + \frac{3}{5} = \frac{2 \times 5 + 3 \times 3}{3 \times 5} = \frac{19}{15}$$

- A subtração é similar à soma

$$\frac{a}{b} - \frac{c}{d} = \frac{a \times d}{b \times d} - \frac{b \times c}{b \times d} = \frac{a \times d - b \times c}{b \times d} \text{ (denominador comum)}$$

$$\text{Exemplo : } \frac{2}{3} - \frac{3}{5} = \frac{2 \times 5 - 3 \times 3}{3 \times 5} = \frac{1}{15}$$

Multiplicar e dividir números racionais

- Para multiplicar números racionais temos que multiplicar os numeradores e denominadores

$$\frac{a}{b} \times \frac{c}{d} = \frac{a \times c}{b \times d}$$

$$\text{Exemplo : } \frac{2}{3} \times \frac{3}{5} = \frac{2 \times 3}{3 \times 5} = \frac{6}{15} = \frac{2}{5} \text{ (simplificado)}$$

Não temos de simplificar: este trabalho é feito pelo construtor da classe Racional.

- A divisão é similar à multiplicação (multiplicar o primeiro número pelo inverso do segundo).

$$\frac{a}{b} \div \frac{c}{d} = \frac{a}{b} \times \frac{d}{c} = \frac{a \times d}{b \times c}$$

$$\text{Exemplo : } \frac{2}{3} \div \frac{3}{5} = \frac{2}{3} \times \frac{5}{3} = \frac{2 \times 5}{3 \times 3} = \frac{10}{9}$$

Comparação de números racionais (regra em cruz)

$$\frac{a}{b} < \frac{c}{d} \Leftrightarrow (a \times d) < (b \times c) \qquad \frac{a}{b} == \frac{c}{d} \Leftrightarrow (a \times d) == (b \times c)$$

Apenas utilize a regra em cruz na sobrecarga dos operadores $<$ e $==$.

Para os demais operadores, considere as seguintes equivalências abaixo (x e y são dois números racionais)

$$x \leq y \Leftrightarrow (x < y \;||\; x == y)$$

$$x\; != y \Leftrightarrow\; !\,(x == y)$$

$$x > y \Leftrightarrow\; !\,(x \leq y)$$

$$x \geq y \Leftrightarrow (x > y \;||\; x == y)$$

Operador <<

- Implemente (Sobrecarregue) o operador << na classe Racional
  - A impressão deve ser no formato a/b, caso $b \neq 1$
  - A impressão deve ser no formato a, caso $b = 1$

```
int main() {

    Racional r1(4, 6);
    Racional r2(6, 3);

    // A linha abaixo imprime "r1 => 2/3"

    cout << "r1 => " << r1 << endl;

    // A linha abaixo imprime "r2 => 2"

    cout << "r2 => " << r2 << endl;

    return 0;
}
```

Observações importantes

- Este exercício-programa deve ser elaborado individualmente.
- Vocês devem utilizar **apenas** os conceitos apresentados em aula.
  - Os atributos das classes devem ser **privados**.
  - Com exceção da implementação da sobrecarga do operador (<<) que recebe 2 (dois) parâmetros, a implementação da sobrecarga dos demais operadores deve receber apenas 1 (um) parâmetro.
- Compacte o código-fonte (arquivos .cpp e .h) em um arquivo <RA>.zip

  Exemplo: 658709.zip (Cuidado para não enviar o arquivo errado!)
- O prazo de entrega é o dia 02 de fevereiro de 2025 às 23h55.
- A entrega será feita unicamente pelo ambiente moodle (`https://ava2.ead.ufscar.br`). Não serão aceitos trabalhos enviados por email.