

MÔN HỌC LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG

BÀI THỰC HÀNH

CHƯƠNG 2: QUÁ TẢI TOÁN TỬ

Cài đặt lớp số hữu tỉ (Rational) gồm:

Các thành viên dữ liệu (**private**):

Tử số (numerator) là số nguyên

Mẫu số (denominator) là số nguyên

Các hàm thành viên (**public**):

Constructor mặc định (gán tử số = 0 và mẫu số = 1)

Constructor sao chép

Destructor

Hàm thành viên (**private**):

reduce() để đưa số hữu tỉ về dạng tối giản

Hàm toàn cục:

Tìm ước chung lớn nhất (Greatest Common Divisor) để hàm reduce() gọi khi tối giản phân số. Hàm này có prototype như sau:

int gCD(int, int);

Các toán tử quá tải:

Toán tử gán

Toán tử +, -, *, / hai số hữu tỉ

Toán tử so sánh:

Toán tử ==, >, <, !=

Toán tử cộng một số nguyên với một số hữu tỉ cho kết quả là một số hữu tỉ (đã được rút gọn)

Toán tử cộng một số hữu tỉ với một số nguyên cho kết quả là một số hữu tỉ (đã được rút gọn)

Toán tử chuyển đổi kiểu số hữu tỉ sang số thực (**double**)

Lập trình hướng đối tượng

Toán tử xuất/nhập số hữu tỉ (<</>>)

Hàm thành viên (**public**) kiểm tra một số hữu tỉ có là số âm hay không ?

Ghi chú:

- Tập tin Rational.h chứa giao tiếp (interface) public cho lớp Rational.
- Tập tin Rational.cpp chứa cài đặt (implementation) của các hàm thành viên, hàm toán tử, hàm bạn, ...
- Tập tin RationalTest.cpp là tập tin của client/uer dùng để test lớp Rational.

Tập tin này có nội dung như sau:

```
#include "rationalnumber.h"
```

```
#include <iostream>
```

```
using namespace std;
```

```
/*
```

```
    Driver Written by Cpp Checker
```

```
    =====
```

```
*/
```

```
void main()
```

```
{
```

```
    // Build the objects.
```

```
    Rational a(2,5);           // Regular Ctor
```

```
    Rational b(8,5);           // Regular Ctor
```

```
    Rational c(2,1);           // Regular Ctor
```

```
    Rational d(c);
```

```
    Rational e;                 // default
```

```
    Rational i , r , r1,r2,r3,r4,r5;
```

```
    cout << endl << "the first rational (A) should be 2/5 : " << a << endl;
```

```
    cout << endl << "the second rational (B) should be 8/5 : " << b << endl;
```

Lập trình hướng đối tượng

```
cout << endl << "the third rational (C) should be 2 : " << c << endl;
cout << endl << "the fourth rational (D) should be like (c) = 2 : " << d <<
endl;
cout << endl << "the fifth rational (E) (checks default Ctor) should be 0 : " <<
e << endl;
```

```
cout << "checking the Cin function on Rational (I): " << endl;
cin >> i;
cout << endl << i << endl;
```

```
cout << endl;
```

```
cout << " Arithmathics operations " << endl;
cout << " ===== " << endl;
cout << "Rational a + b should be 50/25 : " << a + b << endl;
cout << "Rational c - b should be 2/5 : " << c - b << endl;
cout << "Rational b - c should be -2/5 : " << b - c << endl;
cout << "Rational c * b should be 16/5: " << c * b << endl;
cout << "Rational b / a should be 40/10: " << b / a << endl;
cout << "Check returned value for last four Aritmethic operations" << endl;
r1 = a + b;
r2 = c - b;
r3 = b - c;
r4 = c * b;
r5 = b / a;
cout << r1 << " " << r2 << " " << r3 << " " << r4 << " " << r5 << endl;
cout << " ===== " << endl;
```

```
cout << " Logical operations " << endl;
cout << " ===== " << endl;
cout << " a > b should be false: " << (a>b) << endl;
```

Lập trình hướng đối tượng

```
cout << " a < b should be True: " << (a<b) << endl;
cout << " a == b should be false: " << (a==b) << endl;
cout << " a != b should be True: " << (a!=b) << endl;
cout << " c == d should be True: " << (c==d) << endl;
cout << " ===== " << endl;

cout << " Cast & Arithmathics operations " << endl;
cout << " ===== " << endl;
cout << " a + 2 should be 12/5: " << a + 2 << endl;
cout << " 2 + a should be 12/5: " << 2 + a << endl;
cout << "cast a to double should be 0.4: " << double(a) << endl;
cout << " ===== " << endl;

cout << "\n\nDivision by Zero" << endl;
cout << " ===== " << endl;
cout << "Rational g(3,0): ";
Rational g(3,0);
cout << g << endl;
Rational h(0,1);
cout << "Divide by zero - a/h should not give abnormal exit:" << a/h << endl;
cout << " ===== " << endl;
cout << "Good Luck" << endl;

}
```