

Предефиниране на оператори: упражнение (Rev: 1.2)

Любомир Чорбаджиев

11 април 2007 г.

Във файла `09-lab.zip` е даден е кодът на класа `class Rational` и примерна `main()` функция, която го използва.

Задачи

Задача 1: Да се разгледа файла `Rational.cpp`, дефиницията на класа `Rational` и `main()` функцията. Да се компилира програмата и да се изпълни.

□

Задача 2: В класа `Rational` да се добави предефиниране на операторите:

```
1 Rational& operator+=(const Rational& r);
2 Rational& operator-=(const Rational& r);
3 Rational& operator*=(const Rational& r);
4 Rational& operator/=(const Rational& r);
```

В главна функция на програмата `main()` да се добави код, който да тества правилната работа на предефинираните оператори.

□

Задача 3: Да се предефинират операторите:

```
1 Rational operator+(const Rational& r1, const Rational& r2);
2 Rational operator-(const Rational& r1, const Rational& r2);
3 Rational operator*(const Rational& r1, const Rational& r2);
4 Rational operator/(const Rational& r1, const Rational& r2);
```

В главна функция на програмата `main()` да се добави код, който да тества правилната работа на предефинираните оператори.

□

Задача 4: Да се предефинира оператора за изход:

```
1 ostream& operator<<(ostream& out, const Rational& r);
```

□

Приложение

```
1 #include <iostream>
2 using namespace std;
3
4 class RationalError{};
5
6 class Rational {
7     long num_, den_;
8
9     long gcd(long r, long s) {
10         while(s!=0) {
11             long temp=r;
12             r=s;
13             s=temp % s;
14         }
15         return r;
16     }
17
18     void reduce(void) {
19         if(num_==0){
20             den_=1;
21         } else {
22             long tempnum=(num_<0)?-num_:num_;
23             long g=gcd(tempnum,den_);
24             if (g>1){
25                 num_/=g;
26                 den_/=g;
```

```

27     }
28 }
29 }
30
31 void standardize(void) {
32     if(den_<0) {
33         den_=-den_;
34         num_=-num_;
35     }
36     reduce();
37 }
38 public:
39 Rational(int num=0, int den=1){
40     num_=num;
41     den_=den;
42
43     if(den_==0)
44         throw RationalError();
45     standardize();
46 }
47
48 long get_numerator() {return num_;}
49 long get_denominator() {return den_;}
50
51 void dump() {
52     cout << "(" << num_ << "/" << den_ << ")";
53 }
54
55 void add(Rational r) {
56     num_=num_*r.den_+den_*r.num_;
57     den_=den_*r.den_;
58     standardize();
59 }
60 void sub(Rational r) {
61     num_=num_*r.den_-den_*r.num_;
62     den_=den_*r.den_;
63     standardize();
64 }
65 void multiplication(Rational r) {
66     num_*=r.num_;

```

```

67     den_*=r.den_;
68     standardize();
69 }
70 void division(Rational r) {
71     num_*=r.den_;
72     den_*=r.num_;
73     standardize();
74 }
75 };
76
77 int main(int argc, char* argv[]) {
78     Rational r(1,2),p(2,3), q(4,2), s(-3,-9);
79
80     r.dump();p.dump();q.dump();s.dump();
81     cout << endl;
82
83     r.add(p);
84     r.dump();
85     cout << endl;
86
87     p.add(s);
88     p.dump();
89     cout << endl;
90
91     r.multiplication(q);
92     r.dump();
93     cout << endl;
94
95     return 0;
96 }

```