8-5 Fraction

3220105096　王思睿

2024 年 3 月 29 日

添加了对分数进行自动约分的功能，并判断结果是否为 0；

实现对输入格式不正确的报错。

```
输入不正确的格式，如输入2222报错
    Format error,should be 'numerator/denominator'

输入的数字超过范围，如3333333333322222222/3报错
    Input value out of range:
       3333333333322222222/3

输入错误的分数，比如www/2， 报错示例
    Invalid input: www/2

输入分母是0，报错
    Denominator shouldn't be 0!
```

Listing 1: 报错示例

这个 project 有如下基本要求：

Write a class that represents a fraction number like 2/3.

The functions below have to be implemented for this class:

1. **Default ctor**
2. **Ctor takes two integers as parameters**
3. **Copy ctor**

4. **Arithmetical operators: + - * /**
5. **Relational operators: < <= == != >= >**
6. **Typecast to double**
7. **To string**
8. **Inserter and extractor for streams**
9. **Conversion from a finite decimal string like: 1.414**

```
// 构造函数：Fraction
// 描述：默认构造函数，分子和分母都为1
Fraction();
```

Listing 2: 1.Default ctor

```
Fraction::Fraction(int a, int b)
{
    // 初始化分子和分母。
    this->denominator = b;
    this->numerator = a;
}
```

Listing 3: 2.Ctor takes two integers as parameters

```
// 拷贝构造函数，用于创建一个已存在分数的副本。
Fraction::Fraction(const Fraction& other)
{
    // 拷贝分子和分母。
    this->denominator = other.denominator;
    this->numerator = other.numerator;
}
```

Listing 4: 3.Copy ctor

```
// 运算符重载：+
// 描述：实现分数的加法运算
```

```
    Fraction operator + (Fraction);

    // 运算符重载：-
    // 描述：实现分数的减法运算
    Fraction operator - (Fraction);

    // 运算符重载：*
    // 描述：实现分数的乘法运算
    Fraction operator * (Fraction);

    // 运算符重载：/
    // 描述：实现分数的除法运算
    Fraction operator / (Fraction);
```

Listing 5: 4.Arithmetical operators: + - * /

```
    // 运算符重载：<
    // 描述：实现分数的小于比较运算
    bool operator < (const Fraction&);

    // 运算符重载：<=
    // 描述：实现分数的小于等于比较运算
    bool operator <= (const Fraction&);

    // 运算符重载：!=
    // 描述：实现分数的不等于比较运算
    bool operator != (const Fraction&);

    // 运算符重载：==
    // 描述：实现分数的等于比较运算
    bool operator == (const Fraction&);

    // 运算符重载：>
    // 描述：实现分数的大于比较运算
```

```
bool operator > (const Fraction&);

// 运算符重载: >=
// 描述: 实现分数的大于等于比较运算
bool operator >= (const Fraction&);
```

Listing 6: 5.Relational operators: <, <=, ==, !=, >=, >

```
// 将分数转换为双精度浮点数并输出。
void fractionoper::frac_to_dou()
{
    Fraction a;
    double a_double;
    std::string frac1;
    std::cout << "Enter your fraction:" << std::
        endl;

    std::cin >> frac1;
    try
    {
        a = Fraction(frac1); // 尝试将字符串转换为
            分数对象。
    }
    catch(const std::exception& e)
    {
        system("clear"); // 清屏。
        std::cerr << e.what() << '\n'; // 打印错误
            信息。
        return;
    }
    // 进行转换并输出结果。
    a_double = static_cast<double>(a.getnum()) /
        static_cast<double>(a.getden());
    std::cout << a_double << std::endl;
}
```

Listing 7: 6.Typecast to double

```
// 将分数转换为字符串表示，如"1/2"。
std::string Fraction::printstr()
{
    std::stringstream ss;
    ss << this->numerator << "/" << this->
        denominator;
    return ss.str();
}
```

Listing 8: 7.To string

```
friend std::istream& operator>>(std::istream& in,
    Fraction& frac);
friend std::ostream& operator<<(std::ostream& out,
     const Fraction& frac);

void fractionoper::oper_stream()
{
    Fraction a;
    std::cout<<"Input a fraction\n";
    std::cin>>a;
    std::cout<<"The fraction is "<<a<<std::endl;
}
```

Listing 9: 8.Inserter and extractor for streams

```
void fractionoper::dou_to_frac()
{
    std::string frac1;
    std::cout << "Enter your double number" << std::endl;
    getchar(); // 清理输入缓冲区。
    std::cin >> frac1;
    double num = std::stod(frac1); // 将字符串转换为double。
    int places = frac1.find('.'); // 查找小数点的位置。
    places = frac1.size() - places; // 计算小数位数。
    int numerator = static_cast<int>(num * pow(10, places)); // 转换为分子。
    int denominator = static_cast<int>(pow(10, places)); // 计算分母。
    int max_gcd = gcd(numerator, denominator); // 计算最大公约数以简化分数。
    // 输出转换结果。
    std::cout << "Conversion from a finite decimal string: " << frac1 << " to a fraction is " << numerator / max_gcd << "/" << denominator / max_gcd << std::endl;
}
```

Listing 10: 9.Conversion from a finite decimal string like: 1.414

效果展示：

```
Which operation would you like to take?
1: Add
2: Sub
3: Mul
4: Div
5: Compare 2 fractions
6: Turn a fraction to double
```

```
7: Turn a double to fraction
8: Turn a fraction to string
9: Inserter and extractor for streams
10: Exit

1
Enter your first fraction:
3/3
Enter your second fraction:
-5/5
The result of 3/3 + -5/5 is 0
Which operation would you like to take?
1: Add
2: Sub
3: Mul
4: Div
5: Compare 2 fractions
6: Turn a fraction to double
7: Turn a double to fraction
8: Turn a fraction to string
9: Inserter and extractor for streams
10: Exit

1
Enter your first fraction:
2/3
Enter your second fraction:
4/6
The result of 2/3 + 4/6 is 4/3
```

```
Which operation would you like to take?
1: Add
2: Sub
3: Mul
4: Div
5: Compare 2 fractions
6: Turn a fraction to double
7: Turn a double to fraction
8: Turn a fraction to string
9: Inserter and extractor for streams
10: Exit

1
Enter your first fraction:
3/5
Enter your second fraction:
-7/10
The result of 3/5 + -7/10 is -1/10
```

Listing 11: 测试 add

```
Which operation would you like to take?
1: Add
2: Sub
3: Mul
4: Div
5: Compare 2 fractions
6: Turn a fraction to double
7: Turn a double to fraction
8: Turn a fraction to string
```

```
9: Inserter and extractor for streams
10: Exit

2
Enter your first fraction:
3/4
Enter your second fraction:
7/8
The result of 3/4 - 7/8 is -1/8
Which operation would you like to take?
1: Add
2: Sub
3: Mul
4: Div
5: Compare 2 fractions
6: Turn a fraction to double
7: Turn a double to fraction
8: Turn a fraction to string
9: Inserter and extractor for streams
10: Exit

2
Enter your first fraction:
5/10
Enter your second fraction:
1/2
The result of 5/10 - 1/2 is 0
```

Listing 12: 测试 sub

```
Which operation would you like to take?
1: Add
2: Sub
3: Mul
4: Div
5: Compare 2 fractions
6: Turn a fraction to double
7: Turn a double to fraction
8: Turn a fraction to string
9: Inserter and extractor for streams
10: Exit

3
Enter your first fraction:
0/3
Enter your second fraction:
2/2
The result of 0/3 * 2/2 is 0
Which operation would you like to take?
1: Add
2: Sub
3: Mul
4: Div
5: Compare 2 fractions
6: Turn a fraction to double
7: Turn a double to fraction
8: Turn a fraction to string
9: Inserter and extractor for streams
10: Exit
```

```
3
Enter your first fraction:
5/4
Enter your second fraction:
-6/8
The result of 5/4 * -6/8 is -15/16
```

Listing 13: 测试 mul

```
Which operation would you like to take?
1: Add
2: Sub
3: Mul
4: Div
5: Compare 2 fractions
6: Turn a fraction to double
7: Turn a double to fraction
8: Turn a fraction to string
9: Inserter and extractor for streams
10: Exit

4
Enter your first fraction:
4/5
Enter your second fraction:
-3/4
The result of (4/5) / (-3/4) is 16/-15
Which operation would you like to take?
1: Add
```

```
2: Sub
3: Mul
4: Div
5: Compare 2 fractions
6: Turn a fraction to double
7: Turn a double to fraction
8: Turn a fraction to string
9: Inserter and extractor for streams
10: Exit

4
Enter your first fraction:
3/4
Enter your second fraction:
0/4
The result of (3/4) / (0/4) is 0
```

Listing 14: 测试 div

```
Which operation would you like to take?
1: Add
2: Sub
3: Mul
4: Div
5: Compare 2 fractions
6: Turn a fraction to double
7: Turn a double to fraction
8: Turn a fraction to string
9: Inserter and extractor for streams
10: Exit
```

```
5
Enter your first fraction:
3/3
Enter your second fraction:
7/7
3/3 == 7/7
3/3 <= 7/7
3/3 >= 7/7
Which operation would you like to take?
1: Add
2: Sub
3: Mul
4: Div
5: Compare 2 fractions
6: Turn a fraction to double
7: Turn a double to fraction
8: Turn a fraction to string
9: Inserter and extractor for streams
10: Exit

5
Enter your first fraction:
6/7
Enter your second fraction:
-3/4
6/7 > -3/4
6/7 >= -3/4
6/7 != -3/4
```

Listing 15: 测试 Compare 2 fractions

```
Which operation would you like to take?
1: Add
2: Sub
3: Mul
4: Div
5: Compare 2 fractions
6: Turn a fraction to double
7: Turn a double to fraction
8: Turn a fraction to string
9: Inserter and extractor for streams
10: Exit

6
Enter your fraction:
4/6
0.666667

Which operation would you like to take?
1: Add
2: Sub
3: Mul
4: Div
5: Compare 2 fractions
6: Turn a fraction to double
7: Turn a double to fraction
8: Turn a fraction to string
```

```
9: Inserter and extractor for streams
10: Exit

6
Enter your fraction:
-4/7
-0.571429
```

Listing 16: 测试 Turn a fraction to double

```
Which operation would you like to take?
1: Add
2: Sub
3: Mul
4: Div
5: Compare 2 fractions
6: Turn a fraction to double
7: Turn a double to fraction
8: Turn a fraction to string
9: Inserter and extractor for streams
10: Exit

7
Enter your double number
1.414
Conversion from a finite decimal string:
   1.414 to a fraction is 707/500

Which operation would you like to take?
1: Add
```

```
2: Sub
3: Mul
4: Div
5: Compare 2 fractions
6: Turn a fraction to double
7: Turn a double to fraction
8: Turn a fraction to string
9: Inserter and extractor for streams
10: Exit

7
Enter your double number
-1.414
Conversion from a finite decimal string:
   -1.414 to a fraction is -707/500
```

Listing 17: 测试 Turn a double to fraction

```
Which operation would you like to take?
1: Add
2: Sub
3: Mul
4: Div
5: Compare 2 fractions
6: Turn a fraction to double
7: Turn a double to fraction
8: Turn a fraction to string
9: Inserter and extractor for streams
10: Exit

```

```
8
Input your numerator and denominator:
23 24
The fraction to string is: 23/24
```

Listing 18: 测试 Turn a fraction to string

```
Which operation would you like to take?
1: Add
2: Sub
3: Mul
4: Div
5: Compare 2 fractions
6: Turn a fraction to double
7: Turn a double to fraction
8: Turn a fraction to string
9: Inserter and extractor for streams
10: Exit

9
Input a fraction
34/42
The fraction is 34/42
```

Listing 19: 测试 Inserter and extractor for streams

```
输入10表示exit
程序会退出。
```

Listing 20: 测试 Exit