**C++课程设计报告**

**分数计算器设计**



班 级：网络1602

姓 名：邓 英 豪

学 号：3160610049

## 目录

[目录 2](#_Toc19215)

[一、 题目与要求 3](#_Toc19462)

[1.题目： 3](#_Toc3434)

[2.问题描述： 3](#_Toc15959)

[3.功能要求： 3](#_Toc18038)

[二、解决方案与思想 3](#_Toc11992)

[1.解决方案 3](#_Toc16163)

[2.思想 4](#_Toc3802)

[2.1结构图 4](#_Toc18205)

[2.2实现 5](#_Toc27794)

[2.2.1 5](#_Toc1068)

[ 输入式子给出计算结果 5](#_Toc3424)

[2.2.2 6](#_Toc2535)

[ 随机生成式子并判断输入结果的正确性 6](#_Toc29529)

[三、 源代码及其具体实现 7](#_Toc25503)

[3.1 源代码 7](#_Toc28650)

[3.2 具体实现 21](#_Toc19172)

[四、 总结 23](#_Toc13522)

## 题目与要求

1.题目：分数计算器的设计。

### 2.问题描述：

（1）定义一个整数类。

（2）定义一个分数类，由整数类派生，能对分数进行各种计算和输出输入。

### 3.功能要求：

（1）定义整数类和分数类，其中包括构造函数、析构 函数、显示函数等；

（2）输入输出：对插入、提取运算符进行重载；

（3）计算功能：对分数进行加、减、乘、除运算；

（4）化简功能：将分数化成最简分数；

（5）异常处理：分母不能为零；

（6）菜单功能：每种功能的操作都是在菜单中进行相应选择。

## 二、解决方案与思想

### 1.解决方案

（1）应用系统分析，建立该系统的功能模板框图以及界面的组织与设计；

（2）分析系统的各个实体以及他们之间的联系；

（3）根据问题描述，设计系统的类层次；

（4）完成类层次中各个类的描述；

（5）完成各个类中成员函数的定义；

（6）完成系统的应用模块；

（7）功能调试；

### 2.思想

### 2.1结构图

整数类

随机生成式子判断输入结果的正确性

输入式子给出计算结果

分数类

图1

### 2.2实现

### 2.2.1

### 输入式子给出计算结果

1. 重载“<<”和“>>”运算符，完成对分式的输出与输入以及对分母的检查。（代码实现：如下图）

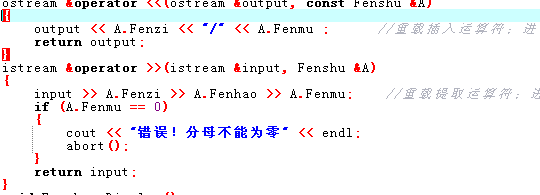


图2

1. 重载“+”，“-”，“\*”，“/”四则运算符号实现对分式的计算。（代码如图：）

IMG_256

图3

3.对分式进行判断看是否需要进行通分计算，并根据菜单选择所要进行的运算。（代码如图：）



图4

4.化简功能。先找出分子和分母的最大公因数，再进行约分处理。（如图：）

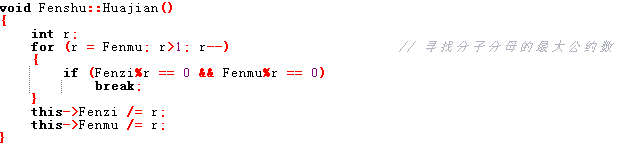


图5

### 2.2.2

### 随机生成式子并判断输入结果的正确性

1.用户输入分式的最大整数范围，系统随机生成两个分式，分式由分子和分母两个整数类构成，而分子和分母是有系统随机生成组合成一个分式，故分式是随机的。（代码如图：）

IMG_256

图5

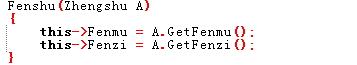


图6

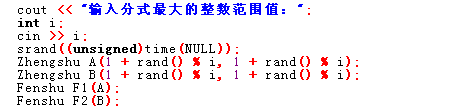


图7

2.随机生成四则运算，分别用1,2,3,4代替加减乘除。

3.对用户输入的计算结果进行判断，错则显示出正确答案。

## 源代码及其具体实现

### 3.1 源代码

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* 分数计算器设计 \*\*

\* 功能要求： \*\*

\*（1）定义整数类和分数类，其中包括构造函数、析构函数、显示函数等；\*\*

\*（2）输入输出：对插入、提取运算符进行重载； \*\*

\*（3）计算功能：对分数进行加、减、乘、除运算； \*\*

\*（4）化简功能：将分数化成最简分数； \*\*

\*（5）异常处理：分母不能为零； \*\*

\*（6）菜单功能：每种功能的操作都是在菜单中进行相应选择； \*\*

\*完成时间： 2017年6月22日 \*\*

\*设计人： 邓英豪 \*/

#include<iostream>

#include<ctime>

#include<cstdlib>

using namespace std;

class Zhengshu

{

private:

int Fenzi, Fenmu;

public:

Zhengshu(){}

Zhengshu(int FZ, int FM) :Fenzi(FZ), Fenmu(FM)

{}

int GetFenzi(){ return Fenzi; }

int GetFenmu(){ return Fenmu; }

virtual ~Zhengshu(){}

};

class Fenshu :public Zhengshu

{

private:

int Fenzi, Fenmu;

char Fenhao;

public:

Fenshu(){}

Fenshu(Zhengshu A)

{

this->Fenmu = A.GetFenmu();

this->Fenzi = A.GetFenzi();

}

friend Fenshu operator +(Fenshu A, Fenshu B);

friend Fenshu operator -(Fenshu A, Fenshu B);

friend Fenshu operator \*(Fenshu A, Fenshu B);

friend Fenshu operator /(Fenshu A, Fenshu B);

friend ostream &operator <<(ostream &, const Fenshu &);

friend istream &operator >>(istream &, Fenshu &);

void Display();

void JiSuan(Fenshu a1, Fenshu a2, char fuhao);

void SShow();

void Huajian();

void Show(Fenshu A);

~Fenshu(){}

};

Fenshu operator +(Fenshu A, Fenshu B)

{

Fenshu C;

C.Fenzi = A.Fenzi + B.Fenzi;

C.Fenmu = A.Fenmu = B.Fenmu;

return C;

}

Fenshu operator -(Fenshu A, Fenshu B)

{

Fenshu C;

C.Fenzi = A.Fenzi - B.Fenzi;

C.Fenmu = A.Fenmu = B.Fenmu;

return C;

}

Fenshu operator \*(Fenshu A, Fenshu B)

{

Fenshu C;

C.Fenzi = A.Fenzi\*B.Fenzi;

C.Fenmu = A.Fenmu\*B.Fenmu;

return C;

}

Fenshu operator /(Fenshu A, Fenshu B)

{

Fenshu C;

C.Fenzi = A.Fenzi\*B.Fenmu;

C.Fenmu = A.Fenmu\*B.Fenzi;

return C;

}

ostream &operator <<(ostream &output, const Fenshu &A)

{

output << A.Fenzi << "/" << A.Fenmu ; //重载插入运算符；进行分数输出

return output;

}

istream &operator >>(istream &input, Fenshu &A)

{

input >> A.Fenzi >> A.Fenhao >> A.Fenmu; //重载提取运算符；进行分数输入

if (A.Fenmu == 0)

{

cout << "错误！分母不能为零" << endl;

abort();

}

return input;

}

void Fenshu::Display()

{

cout << "计算结果是：" << Fenzi << "/" << Fenmu << endl;

}

void Fenshu::JiSuan(Fenshu a1, Fenshu a2, char fuhao)

{

Fenshu A, B, C;

if (a1.Fenmu == a2.Fenmu)

{

A.Fenmu = B.Fenmu = a1.Fenmu; //判断是否需要进行通分加减

A.Fenzi = a1.Fenzi;

B.Fenzi = a2.Fenzi;

}

else

{

A.Fenzi = a1.Fenzi\*a2.Fenmu;

A.Fenmu = a1.Fenmu\*a2.Fenmu;

B.Fenzi = a2.Fenzi\*a1.Fenmu;

B.Fenmu = A.Fenmu;

}

switch (fuhao) //根据选择进行运算

{

case '+':

C = A + B;

break;

case '-':

C = A - B;

break;

case '\*':

C = A\*B;

break;

case '/':

C = A / B;

break;

case 1:

C = A + B;

cout << a1 << " + " << a2 << " = " << endl;

break;

case 2:

C = A - B;

cout << a1 << " - " << a2<< " = " << endl;

break;

case 3:

C = A\*B;

cout << a1 << " \* " << a2 << " = " << endl;

break;

case 4:

C = A / B;

cout << a1 << " / " << a2 << " = " << endl;

break;

}

if (C.Fenmu == 0)

{

cout << "错误！分母不能为零" << endl;

abort();

}

else

{

this->Fenzi = C.Fenzi;

this->Fenmu = C.Fenmu;

}

}

void Fenshu::SShow()

{

Fenshu D; //真分数化简

D.Fenzi = Fenzi;

D.Fenmu = Fenmu;

if (D.Fenzi<D.Fenmu)

cout << "化简结果为：" << D << endl;

else

{

int n;

n = D.Fenzi / D.Fenmu; //假分数化为代分数

D.Fenzi = D.Fenzi - n\*D.Fenmu;

if (D.Fenzi == 0)

cout << n << endl;

else

cout << "化简结果为：" << n << "+" << D << endl;

}

}

void Fenshu::Huajian()

{

int r;

for (r = Fenmu; r>1; r--) // 寻找分子分母的最大公约数

{

if (Fenzi%r == 0 && Fenmu%r == 0)

break;

}

this->Fenzi /= r;

this->Fenmu /= r;

}

void Fenshu::Show(Fenshu A)

{

if (Fenzi == A.Fenzi&&Fenmu == A.Fenmu)

cout << "You are right!" << endl;

else

cout << "正确答案为：" << Fenzi << "/" << Fenmu << endl;

}

int main()

{

cout << endl;

cout << "\t\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*欢迎使用分数计算器\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

cout << "\t\* \*" << endl;

cout << "\t\* 计算你的式子(1) \*" << endl;

cout << "\t\* \*" << endl;

cout << "\t\* 随 机 计 算 (2) \*" << endl;

cout << "\t\* \*" << endl;

cout << "\t\*注释： \*" << endl;

cout << "\t\*(1)输入你所要计算的两个分式，然后选择所要进行的运算,\*" << endl;

cout << "\t\* 及选择是否需要化简。 \*" << endl;

cout << "\t\*(2)系统随机给出一个算式，输入你的计算结果判断对错 \*" << endl;

cout << "\t\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

cout << endl << "请选择：";

int k;

cin >> k;

if (k == 1)

{

Fenshu F1, F2, F3;

char yunsuan;

for (int ch = 0; ch != 3;)

{

cout << "分别输入两个分式：\n";

cin >> F1 >> F2;

for (; ch != 2 && ch != 3;)

{

cout << "选择你所要进行的运算：";

cin >> yunsuan;

F3.JiSuan(F1, F2, yunsuan);

F3.Display();

cout << "是否需要化简(1是；0否）：";

int i;

cin >> i;

if (i == 1)

{

F3.Huajian();

F3.SShow();

system("pause");

cout << "对该分式重新进行运算（1）" << endl; //选择功能

cout << "重新输入分式进行运算（2）" << endl;

cout << "退出计算器（3）" << endl;

cin >> ch;

if (ch == 2 && ch == 3)

break;

}

else

{

cout << "对该分式重新进行运算（1）" << endl; //选择功能

cout << "重新输入分式进行运算（2）" << endl;

cout << "退出计算器（3）" << endl;

cin >> ch;

if (ch == 2 && ch == 3)

break;

}

}

if (ch == 3)

break;

ch = 0; //初始化ch值，防止陷入死循环

}

return 0;

}

else

{

for (int ch1 = 1;;)

{

cout << "输入分式最大的整数范围值：";

int i;

cin >> i;

srand((unsigned)time(NULL));

Zhengshu A(1 + rand() % i, 1 + rand() % i);

Zhengshu B(1 + rand() % i, 1 + rand() % i);

Fenshu F1(A);

Fenshu F2(B);

Fenshu F3;

char ch = 1 + rand() % 4;

F3.JiSuan(F1, F2, ch);

F3.Huajian();

cout << "输入你的答案：";

Fenshu F4;

cin >> F4;

F3.Show(F4);

cout << "是否继续（1）是（0）否：";

cin >> ch1;

if (ch1 == 1)

continue;

else

break;

}

return 0;

}

cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*谢谢使用\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

}

### 3.2 具体实现

代码实现如下列图：

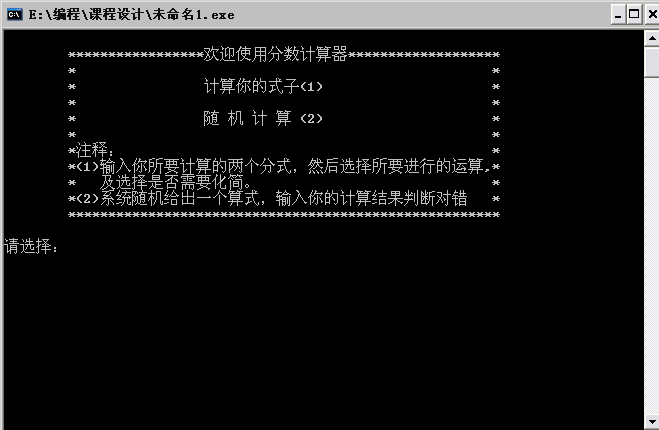


图8

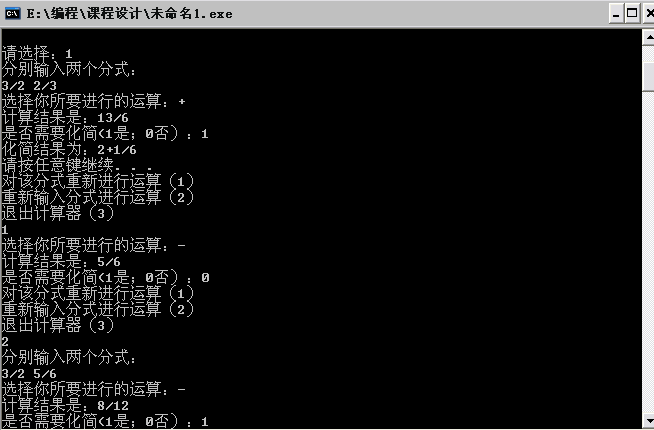


图9

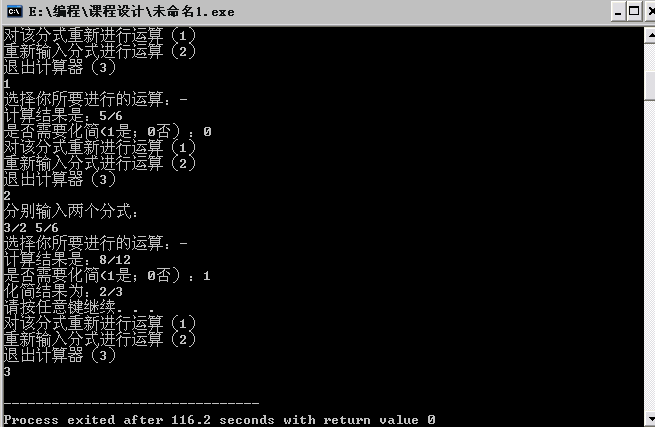


图10



图11

## 总结

本次课程设计题目较为简单，所以在完成时间上较快。这次设计对运算符重载要求较高，做完实验后能充分的熟悉并掌握相关知识。其次也就是在回顾与复习一下switch,rand等函数的用法。