

**《C++语言程序设计》实验报告**

**实验二：编写自己的类**

**姓 名：李剑寒**

**学 号：Y0914541**

**座 位 号： 92**

**专业班级： 网络空间安全**

**实验日期：2021/4/29**

**安徽大学互联网学院**

**实验内容**

**一、实验要求：**

1. 分数相加是数学中常用的功能，例如，求解方法是先算出两个分数分母的最小公倍数，通分后再求解两个分子之和，最后约简结果分数的分子和分母（例如两个分数相加的结果为，则必须将其约简成最简分数的形式），即分子分母的最大公约数分别被分子分母相除。求解m,n最大公约数的一种方法为：将m，n中最小的那个赋给变量k,然后分别用{k,k-1,k-2,…,1}中的数（递减）去除m和n,第一个能把m和n同时除尽的数就是m和n的最大公约数。假定m和n的最大公约数是v，则它们的最小公倍数就是m×n÷v。请建立一个分数类Fract,完成两个分数相加减的功能，具体要求如下：

（1）类的UML图如下

|  |
| --- |
| **Fract** |
| -num:int // 分子  -den:int // 分母 |
| +Fract(int a=0,int b=1) // 构造函数  +ged(int m,int n):int // 求m,n的最大公约数  +add(Fract f):Fract // 将参数分数f与对象自生相加，返回最简分数  +show():void // 按照num/den的形式在屏幕上显式分数 |

（2）在主函数中定义两个分数对象f1和f2,初值分别为1/5和7/20，通过f1调用成员函数add完成f1和f2的相加，将得到的分数赋给对象f3,显示分数对象f3。

2. 附加题：完成分数的加减乘除运算（感兴趣的同学做）

**二、设计思路及程序代码**

#include<iostream>

using namespace std;

class fract

{

public:

void setnum(int n);

void setden(int m);

fract(int a, int b);

void show();

int getnum();//获取分子

int getden();//获取分母

int ged(int m, int n);//最大公约数

int lcm(int m, int n);//最小公倍数

fract add(fract f);

private:

int num, den;

};

fract::fract(int a=0, int b=1)

{

num = a;

den = b;

}

void fract::show()

{

cout << num << "/" << den << endl;

}

void fract::setnum(int n)

{

num = n;

}

void fract::setden(int m)

{

den = m;

}

int fract::getnum()

{

return num;

}

int fract::getden()

{

return den;

}

int fract::ged(int m, int n)

{

int k;

if (m < n)

k = m;

else

k = n;

while (0 != m%k || 0 != n%k)

{

k =k- 1;

}

return k;

}

int fract::lcm(int m, int n)

{

int k;

if (m < n)

k = m;

else

k = n;

while (0 != m%k || 0 != n%k)

{

k -= 1;

}

return (m\*n)/k;

}

fract fract::add(fract f)

{

int a;

num = num\*f.getden() + f.getnum()\*den;

den = den\*f.getden();

a = ged(den, num);

setnum(num/a);

setden(den/ a);

show();

return f;

}

int main()

{

fract f1(1,5);

f1.show();

fract f2(7,20);

f2.show();

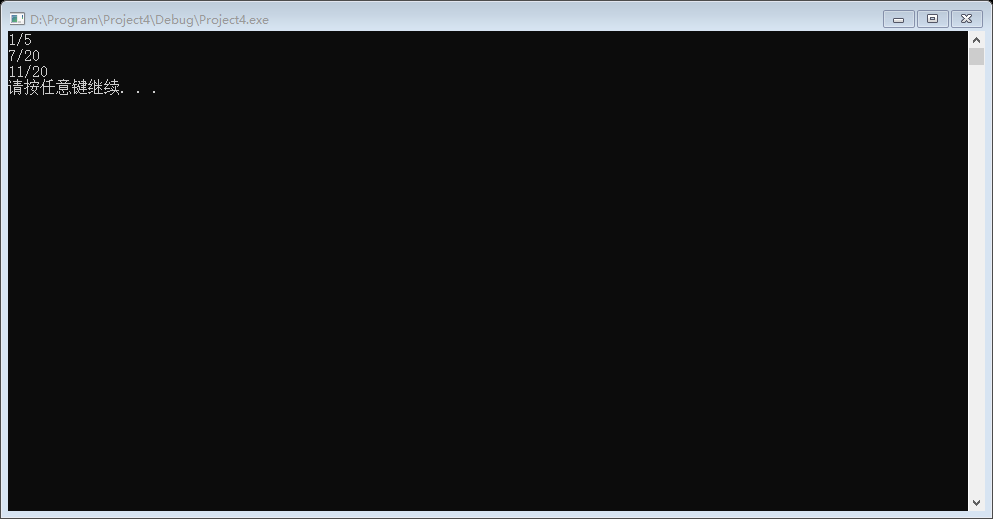
f1.add(f2);

system("pause");

return 0;

}

**三、运行结果及分析**

****