**实验六 运算符重载**

**一、实验目的**

1．掌握运算符重载的概念；

2．握使用friend重载运算符的方法；

**二、实验要求**

1．独立完成程序的编辑、编译、调试及运行；

2．对于程序编译及运行中出现的错误，能够进行改正；

**三、实验内容**

1、重载加、减、乘、除运算符，实现复数类complex的四则运算。

#include<iostream>

#include<stdio.h>

#include<stdlib.h>

using namespace std;

class Complex

{

public:

Complex()

{

real=image=0;

}

Complex(double r)

{

real=r;image=0;

}

Complex(double r,double i)

{

real=r;image=i;

}

friend Complex operator +(const Complex &c1,const Complex &c2);

friend Complex operator -(const Complex &c1,const Complex &c2);

friend Complex operator \*(const Complex &c1,const Complex &c2);

friend Complex operator /(const Complex &c1,const Complex &c2);

friend void print(const Complex &c);

private:

double real,image;

};

Complex operator +(const Complex &c1,const Complex &c2)

{

return Complex(c1.real+c2.real,c1.image+c2.image);

}

Complex operator -(const Complex &c1,const Complex &c2)

{

return Complex(c1.real-c2.real,c1.image-c2.image);

}

Complex operator \*(const Complex &c1,const Complex &c2)

{

return Complex(c1.real\*c2.real-c1.image\*c2.image,c1.real\*c2.image+c1.image\*c2.real);

}

Complex operator /(const Complex &c1,const Complex &c2)

{

return Complex((c1.real\*c2.real+c1.image\*c2.image)/(c2.real\*c2.real+c2.image\*c2.image),

(c1.image\*c2.real-c1.real\*c2.image)/(c2.real\*c2.real+c2.image\*c2.image));

}

void print(const Complex &c)

{

if(c.image<0)

cout<<c.real<<c.image<<"i"<<endl;

else

cout<<c.real<<"+"<<c.image<<"i"<<endl;

}

int main()

{

Complex c1(2.5),c2(1.5,2),c3(3,3),c4;

printf("c2+c3:\n");

c4=c2+c3;

print(c4);

c4=c3-c2;

printf("c3-c4:\n");

print(c4);

printf("c1\*c2:\n");

c4=c2\*c1;

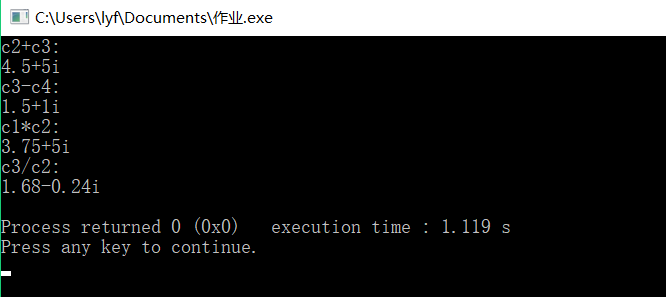
print(c4);

printf("c3/c2:\n");

c4=c3/c2;

print(c4);

}



分析：关键词operator是定义运算符重载的关键词，写了个复数类，实现了类相加的功能。

2、描述时间类，包括小时、分、秒三个数据成员，能够进行显示时间、设置时间等基本操作，在此基础上，重载自增运算符++，能够满足60秒增加1分钟、60分钟增1小时操作，注意能够进行前加和后加的重载。

#include<iostream>

#include<string.h>

#include<stdio.h>

using namespace std;

class Time

{

public:

Time()

{

hour=mint=second=0;

}

Time(int s,int f,int m)

{

hour=s;mint=f;second=m;

}

friend void print(const Time &c);

Time run();

void setTime();

Time operator ++();

Time operator ++(int n);

private:

int hour,mint,second;

};

Time Time::run()

{

second++;

if(second==60)

{

mint++;

second=0;

if(mint==60)

{

hour++;

mint=0;

}

}

return \*this;

}

Time Time::operator ++(int n)//表示a++时进行时的加法计算

{

Time s=\*this;

run();

return s;

}

Time Time::operator ++()//表示++a时，进行的加法计算

{

return run();

}

void print(const Time &c)

{

cout<<c.hour<<':'<<c.mint<<':'<<c.second<<endl;

}

void Time::setTime()

{

int s,f,m;

printf("请输入时间: \n");

scanf("%d:%d:%d",&s,&f,&m);

hour=s;mint=f;second=m;

}

int main()

{

Time c(12,35,59),f;

int i;

f=++c;

printf("初始时间为12:35:59,进行++c之后的时间：\n");

print(f);

c.setTime();

printf("设置时间循环66次c++后的结果:\n");

for(i=0;i<66;i=i+1)

{

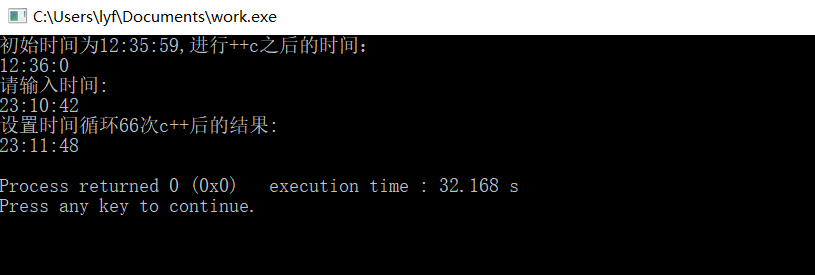
c++;

}

f=c;

print(f);

}



分析:重新重载了++运算符，a++，后面有一个实数，所以用了两个重载。