郑 州 轻 工 业 大 学

**实 验 报 告**

**课程名称： 面向对象程序设计**

**姓 名： 原彬贺**

**学 号： 542001020223**

**专业班级： 计算机科学与技术20-02**

**任课教师： 王岩**

**2021 年 10 月 16 日**

**实验报告正文**

**实验一**

**类与对象（一）**

**一、 实验目的**

1.掌握类的定义方式；

2.掌握类的数据成员和成员函数定义方式；

3.掌握类成员的访问控制方式；

4.理解对象的定义及定义和使用对象的方法；

5.掌握构造函数和析构函数的定义和使用方法；

6.掌握构造函数的重载；

7.了解拷贝构造函数的定义和使用。

**二、 问题的本质和抽象描述**

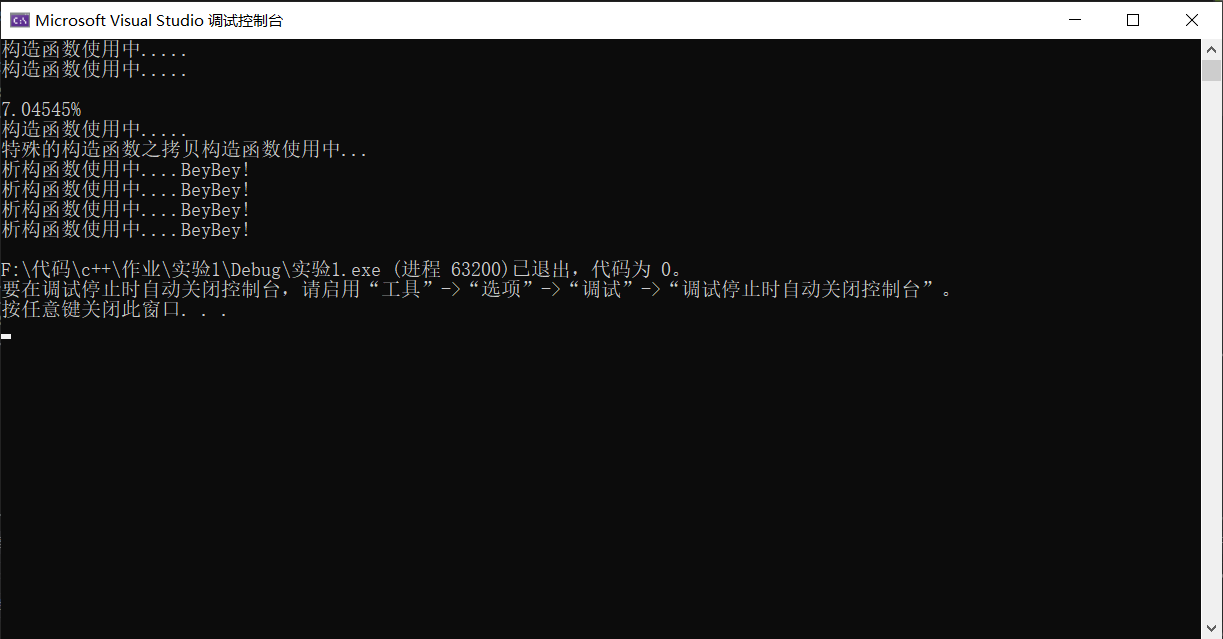
问题的本质是定义一个股票类，进行使用私有和公有数据，再编写构造stock与析构函数 ~stock函数，定义一个类中显示End数据的函数，最后在主函数中进行调用。

**三、 测试**

**1.方案**

定义一个有参数的构造函数和一个无参数的构造函数使其数据都为零，再定义一个拷贝构造函数与输出数据的函数，做一个返回值为End的函数，定义一个析构函数，最后再在主函数中创建三个类的对象。

**2.结果**



**四、 总结与讨论**

在编写过程中，通过对类的定义与使用，熟悉了类的编写。在编写过程中，逐渐懂得了其中代码的意义，比如设置一个函数的返回值为End的函数的重要性，防止数据成员被修改，保证了类中数据的安全性。在这次实验中，收获颇多。

**附：源代码**

#include<iostream>

using namespace std;

class stock

{

public:

stock(int n, float max, float min, float begin, float end)

{

number = n;

Max = max;

Min = min;

Begin = begin;

End = end;

cout << "构造函数使用中....." << endl;

}

stock(const stock& p)

{

number = p.number;

Max = p.Max;

Min = p.Min;

Begin = p.Begin;

End = p.End;

cout << "特殊的构造函数之拷贝构造函数使用中..." << endl;

}

stock()

{

number = 0;

Max = 0;

Min = 0;

Begin = 0;

End = 0;

}

~stock()

{

cout << "析构函数使用中....BeyBey!"<< endl;

}

void stock\_show()

{

cout << number <<"\t" ;

cout << Max << "\t" ;

cout << Min << "\t" ;

cout << Begin << "\t" ;

cout << End << "\t" << endl;

}

float stock\_End()

{

return(End);

}

private:

int number;

float Max, Min, Begin, End;

};

int main()

{

stock s1(1, 4.18, 4.00, 4.05, 4.09);

stock s2(2, 4.41, 4.03, 4.04, 4.40);

cout <<"\n" << (s2.stock\_End() - s1.stock\_End()) / s2.stock\_End() \* 100 << "%" << endl;

stock s3(3, 4.45, 4.05, 4.06, 4.34);

stock s4(s3);

return 0;

}