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 20 日**

**实验报告正文**

**实验一**

**类与对象（二）**

**一、 实验目的**

1.理解对象数组的定义和使用方法；

2.理解对象指针的概念，掌握指针引用对象的方法；

3.掌握 this 指针的工作方式；

4.理解静态数据成员和静态成员函数的定义，并掌握其使用方法；

5.了解友元与友元函数的概念、作用和使用方法。

**二、 问题的本质和抽象描述**

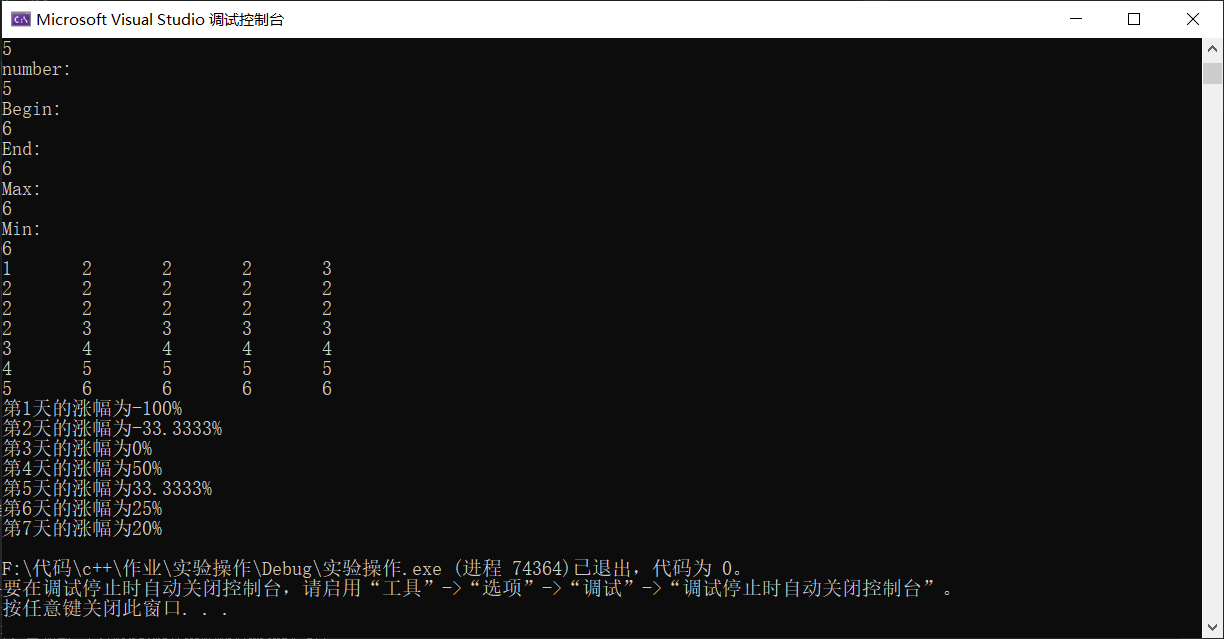
问题的本质是定义一个股票类，通过指针对该类进行引用，并且创建一个一个成员函数对其进行计算涨幅。设置一个静态成员函数计算对象的数量，设置一个友元函数，通过布尔类型测试每日开盘与收盘的差距。

**三、 测试**

**1.方案**

先定义一个含有七个元素的数组对其进行存储，再定义一个stock股票类，设置成员函数计算日涨幅，静态成员函数计算对象个数，友元函数检测每日开盘与收盘价的高低并返回一个布尔类型，设置一个成员函数使用指针对对象赋值。

**2.结果**



**四、 总结与讨论**

在编写过程中，通过对友元函数与静态成员函数的编写与使用，熟悉了友元函数的使用方法。在编写过程中，懂得了使用this指针的含义，程序运行更加迅速快捷，保证了类中数据的安全性。我在这次编写代码的过程中，巩固了许多之前学到的知识。

**附：源代码**

#include<iostream>

using namespace std;

const int N = 7;

class stock

{

public:

static int count;

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

stock(const stock& p);

stock();

~stock();

void stock\_set();

void stock\_copy(stock &);

void stock\_show();

float stock\_End();

friend bool stock\_compare(stock s);

private:

int number;

float Max, Min, Begin, End;

};

int stock::count = 0;

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

{

number = n;

Max = max;

Min = min;

Begin = begin;

End = end;

count = count + 1;

}

stock::stock()

{

number = 0;

Max = 0;

Min = 0;

Begin = 0;

End = 0;

}

stock::stock(const stock& p)

{

number = p.number;

Max = p.Max;

Min = p.Min;

Begin = p.Begin;

End = p.End;

}

stock::~stock()

{

}

void stock::stock\_show()

{

cout << number << "\t";

cout << Max << "\t";

cout << Min << "\t";

cout << Begin << "\t";

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

}

float stock::stock\_End()

{

return(End);

}

void stock::stock\_copy(stock& p)

{

this->Begin = p.Begin;

this->End = p.End;

this->Max = p.Max;

this->Min = p.Min;

cout << this->number << endl;

}

bool stock\_compare(stock s)

{

if (s.Begin<s.End)

{

return true;

}

else

return false;

}

void stock::stock\_set()

{

cout << "number:" << endl;

cin >> number;

cout << "Begin:" << endl;

cin >> Begin;

cout << "End:" << endl;

cin >> End;

cout << "Max:" << endl;

cin >> Max;

cout << "Min:" << endl;

cin >> Min;

}

int main()

{

int i;

stock s1[100] ;

stock\* p;

for (i = 0, p = s1;i<N;p++, i++)

{

p->stock\_set();

}

for (i = 0, p = s1; i < N; p++, i++)

{

p->stock\_show();

}

for (i = 0, p = s1; i < N; i++, p++)

{

cout << "第" << i + 1 << "天的涨幅为" << (p->stock\_End() - (p - 1)->stock\_End()) / (p - 1)->stock\_End() \* 100 << "%" << endl;

}

return 0;

}