|  |  |
| --- | --- |
| **编号:** | **第1章第1次** |

****

信息科学与工程学院实验报告

《面向对象程序设计》

**Object-Oriented Programming**

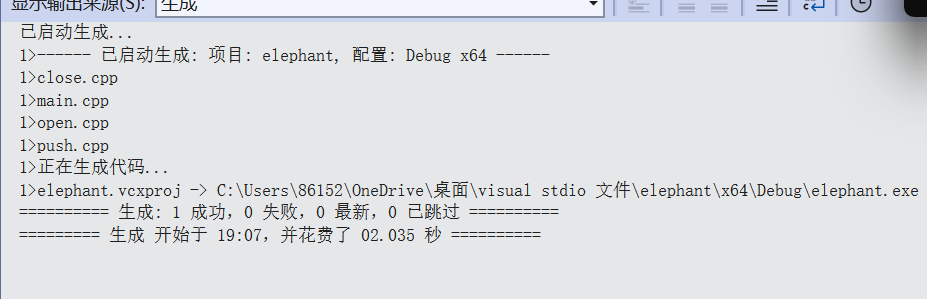
|  |  |
| --- | --- |
| 姓名： | 颜丙超 |
| 学号： | 202311000415 |
| 班级： | 计工本2301 |
| 教师: | 张庆科 |
| 时间： | 2024年9月18日 |

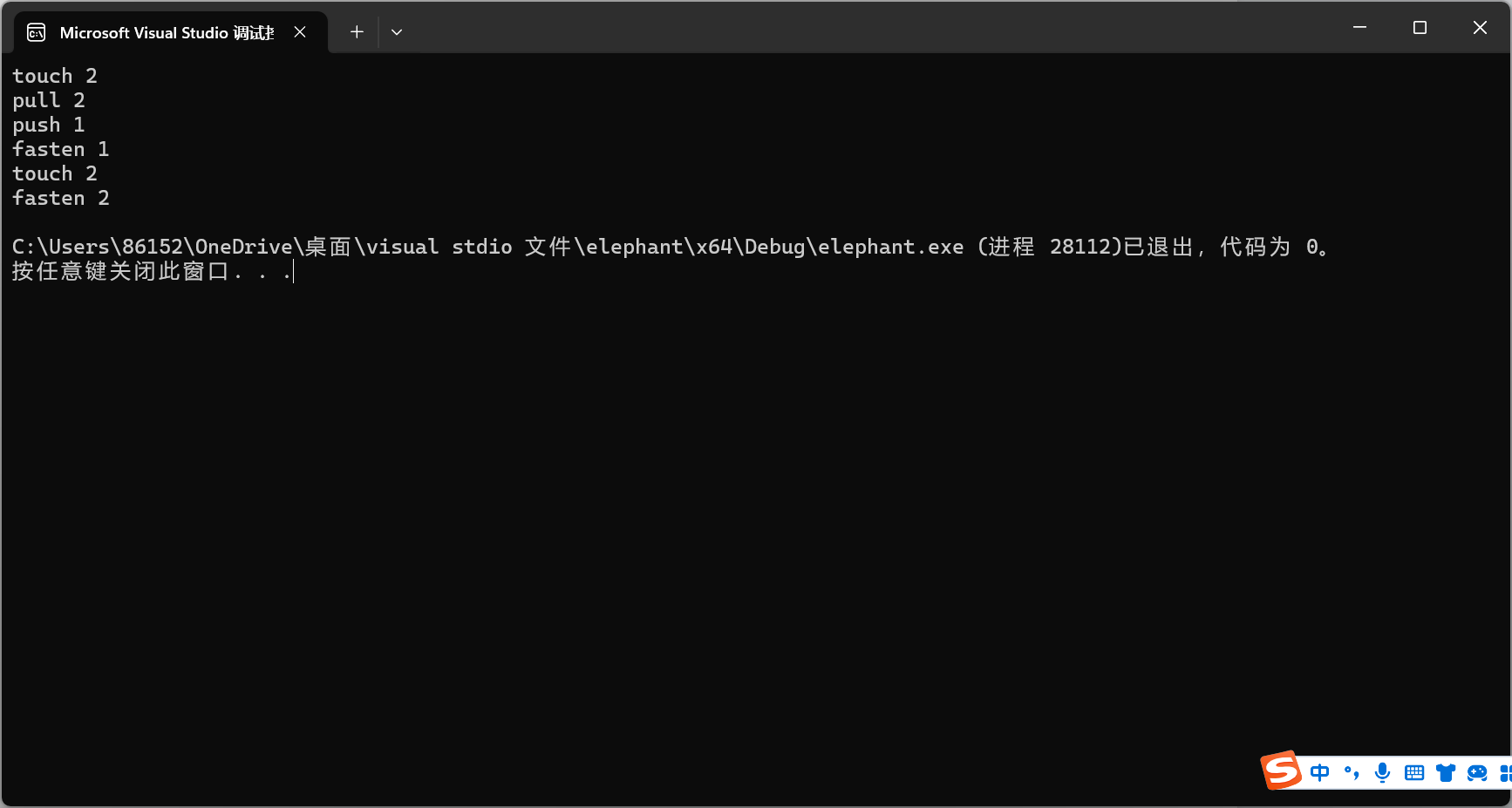
**《面向对象程序设计》实验报告**

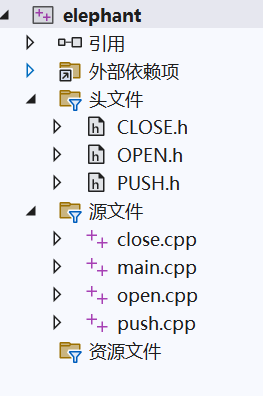
**报告要求：**实验报告包含实验目的、实验内容、实验过程（详细操作流程）、实验结果（程序运行结果高清截图）、实验分析总结五个部分。报告中若涉及代码程序，请在附录部分提供完整程序源码及源码托管地址(基于Highlight软件导入源码)。报告撰写完毕后请将PDF格式版本上传到坚果云作业提交系统。

1. **实验目的**
2. 理解面向过程和面向对象编程的异同点
3. 理解基于c\c++的多文件编程基本原理
4. 能够根据任务需求独立建构多文件程序项目
5. 掌握visual studio c++的项目建构及运行过程
6. **实验内容**
7. 利用c++语言将一个文件进行多文件进行编程
8. 利用多文件编程的实例演示将“大象装进冰箱”分为三步，一是“open”打开冰箱门，二是“push”，把大象装进冰箱 ，三是“close”关上冰箱门。
9. 可以利用番茄插件输入“hh”进行快捷输入。
10. **实验过程**
11. **编写源文件：**首先建立一个源文件main文件先输入三个步骤。
12. **编写头文件：**再建立三个步骤的头文件OPEN PUSH CLOSE,来解释 源文件中三个步骤。
13. **处理：**然后再输入三个步骤的源文件来解释头文件中的代码。
14. **编译：**最后进行程序编程，和一起写的程序是一样的输出结果。

1. **实验结果**







1. **实验总结**

为了实现多文件编程，我们需要将代码分割成多个文件，并在需要的地方包含相应的头文件。这样可以提高代码的可读性和可维护性。多文件编程允许开发者将复杂的程序分解成更小、更易于管理的部分。通过合理地组织代码，可以有效地减少全局变量的使用，从而降低程序的复杂性和出错率。

* **附录：实验源代码（基于Highlight软件粘贴带有行号的源码）**

Main.cpp

#include<stdio.h>

#include"OPEN.h"

#include"PUSH.h"

#include"CLOSE.h"

int elephant = 01, fridge = 02;

int main()

{

/\*1.open the fridge\*/

Touch(fridge);

Pull(fridge);

/\*2.push an elephant into the fridge\*/

Push(elephant);

Fasten(elephant);

/\*3.close the fridge\*/

Touch(fridge);

Close(fridge);

return 0;

}

CLOSE.h

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

\* Head File : CLOSE.h

\* File Usage :

\* Create Time : 2024-9-18/14:55

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

#ifndef \_\_CLOSE\_H\_\_

#define \_\_CLOSE\_H\_\_

void Close(int f);

#include <stdio.h>

#endif

#pragma once

PUSH.h

#ifndef \_\_OPEN\_H\_\_

#define \_\_OPEN\_H\_\_

void Touch(int f);

void Pull(int f);

#include <stdio.h>

#ifndef \_\_PUSH\_H\_\_

#define \_\_PUSH\_H\_\_

void Push(int e);

void Fasten(int e);

#include <stdio.h>

#include "CLOSE.h"

void Close(int f)

{

*printf*("fasten %d\n", f);

}

#include "OPEN.h"

#include <stdio.h>

void Touch(int f)

{

*printf*("touch %d\n", f);

}

void Pull(int f)

{

*printf*("pull %d\n", f);

}

#include "PUSH.h"

#include <stdio.h>

void Push(int e)

{

*printf*("push %d\n", e);

}

void Fasten(int e)

{

*printf*("fasten %d\n", e);

}