

编号: 1-1



山东师范大学
SHANDONG NORMAL UNIVERSITY

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:	张泽浩
学号:	202111000212
班级:	计工本 2102
导师:	张庆科
时间:	2021 年 10 月 7 日



《面向对象程序设计》实验报告

基本要求：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)。报告撰写完毕后请提交 PDF 格式版本报告到课程云班课系统。

一、实验目的

1. 熟悉 Visual Studio 构建 C++ 项目程序的方法
2. 掌握 Git 的基本原理及常见的 Git 命令用法
3. 理解面向过程和面向对象程序设计的差异
4. 掌握条件编译及多文件编程的基本原理

二、实验内容

1. 注册 Gitee 账号
2. 在 Gitee 里创建名为 "Object-oriented Programming" 的仓库
3. 在仓库里添加文件夹 "第一章 实验任务", "第二章 实验任务"
4. 在上述各实验任务文件夹内添加对应章节中任意一个完整的 C++ 源程序
5. 提交本地源代码到远程仓库内, 检查仓库内容与本地内容是否一致

三、实验过程

1. 注册 gitee 账号, 并下载安装 git
2. 配置 git 用户信息:
 - (1) 右键打开 git bash here, 通过控制台窗口配置 git 用户名称
 - (2) 右键打开 git bash here, 通过控制台窗口配置 git 用户邮箱

```
MINGW64/c/Users/HP/Desktop
HP@LAPTOP-K7K7GRRH MINGW64 ~/Desktop
$ git config --global user.name "zehao Zhang"
HP@LAPTOP-K7K7GRRH MINGW64 ~/Desktop
$ git config --global user.email "ze_hao2002@163.com"
HP@LAPTOP-K7K7GRRH MINGW64 ~/Desktop
$ |
```

图 1 配置用户信息



3. 在 gitee 里面创建仓库



图 2 创建仓库操作



图 3 创建成功后的结果

4. 使用 git 提交源码

(1) 克隆该仓库



图 4 复制克隆 HTTPS

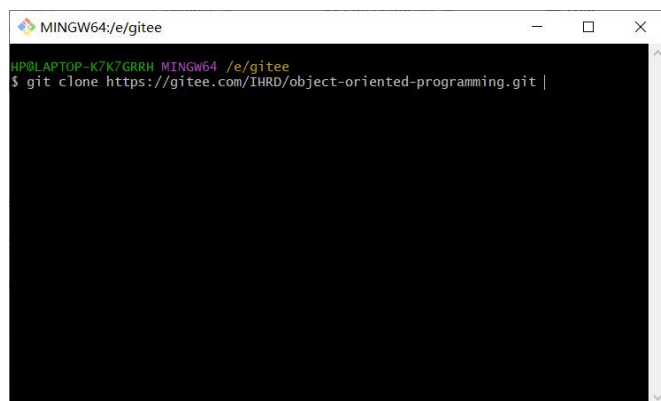


图 5 本地操作

(2) 对仓库源代码进行修改（将大象装进冰箱）



图 6 编写代码

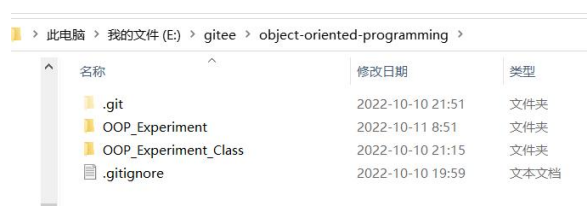


图 7 对本地仓库进行修改

(3) 将修改后版本通过 git 命令上传到 gitee 远程仓库

常用的 git 命令：

- ① git clone HTTPS 链接 -----创建本地仓库操作
- ② git status -----查看仓库的修改状态
- ③ git add . -----合并修改



④ `git commit -m "修复了 bug"` -----相关说明

⑤ `git push` -----提交代码托管平台

```
MINGW64~/e/gitee/object-oriented-programming
HPBLAPTOP-K7K7GRRH MINGW64 /e/gitee/object-oriented-programming (master)
$ git status
On branch master
Your branch is up to date with 'origin/master'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    OOP_Experiment/Experiment-1_1/

nothing added to commit but untracked files present (use "git add" to track)
$ git add .
$ git commit -m "新增Experiment-1_1"
[master 94c784c] 新增Experiment-1_1
25 files changed, 251 insertions(+), 0 deletions(-)
create mode 100644 OOP_Experiment/Experiment-1_1/.vs/Experiment-1_1/v17/.suo
create mode 100644 OOP_Experiment/Experiment-1_1/.vs/Experiment-1_1/v17/Browse.
VC.db
create mode 100644 OOP_Experiment/Experiment-1_1/.vs/Experiment-1_1/v17/ipch/Av
toPCH/55614a782304e866/PUT_ELEPHANT_TO_REFRIGERATOR.ipch
```

图 8 查看当前状态并提交



图 9 gitee 提交结果

四、实验结果

1. 程序编写实验结果

本程序采用多文件编程方法设计并实现了一个"大象装冰箱的程序"。程序分别包含两个.cpp 源文件（main.cpp、Put_Elephant_to_Refrigerator.cpp）、一个.h 头文件（Put_Elephant_to_Refrigerator.h）。

其相关代码见附录。

程序运行结果如下：

```
D:\C & C++\Object-oriented Programming\OOP_Experiment\Experiment-1_1\Debug\Experiment-1_1.exe
>> 如何把大象放入冰箱?
    打开冰箱门
    把大象放入冰箱
    关闭冰箱门
请按任意键继续. . .
```

图 10 程序运行结果

2. Gitee 仓库上传结果

将上述源代码及相关头文件代码拷贝到已经建立好的仓库中，并在含有.git 文件夹的目录下右键，选择“git bash here”，并输入相关命令将仓库内容更



新到远程仓库，完成代码托管。



图 11 上传后的仓库内部文件

五、实验总结

通过本次实验,掌握 gitee 的基本原理和 Git 命令的使用方法,掌握了 Visual Studio 构建 C++项目程序的方法,并构建了一个 Experiment-1_1 的项目。在实验过程中,掌握了多文件编程的基本原理,并通过实现“把大象放入冰箱”的程序体会到了多文件编译的优点。

通过回顾上机课所学内容,以及编译实现代码,体会到 Visual Studio 快捷键的方便与高效(快捷键见表 1)。结合上课老师所讲的面向对象与面向过程程序设计的区别,基于对 C 语言的学习和对 C++项目的实现,跟深刻的感受到了两种程序设计的区别和各自的优点。

表 1 一些常用的快捷键

单步调试(走马观花 F10, 步步深入 F11, 配合使用)	逐行定位 BUG
设置断点(F9), 直接运行到光标所在代码行(ctrl + F10)	部分代码调试
注释(ctrl+ K + C), 取消注释(ctrl+ K + U)	优雅的写代码
代码格式化(先选中代码, 然后 ctrl + K + F)	写优雅的代码
定义跳转(F12), 跳回原位(ctrl + 减号)	查看源码好帮手
一键编译调试运行(F5)	简单粗暴又高效



附录：实验源代码（基于 Highlight 软件粘贴带有行号的源码）

代码托管地址：

https://gitee.com/IHRD/object-oriented-programming/tree/master/OOP_Experiment/Experiment-1_1

main.cpp文件

```
01 #include <iostream>
02 #include "Put_Elephant_to_Refrigerator.h"
03 using namespace std;
04
05 int main() {
06     cout << ">> 如何把大象放入冰箱?" << endl;
07     Put_Elephant_to_Refrigerator();
08     system("pause");
09     return 0;
10 }
```

Put_Elephant_to_Refrigerator.h文件

```
01 void Put_Elephant_to_Refrigerator();
```

Put_Elephant_to_Refrigerator.cpp文件

```
01 #include <iostream>
02 using namespace std;
03
04 void Open_the_Door() {
05     cout << "\t打开冰箱门" << endl;
06 }
07
08 void Put_Exlephant() {
09     cout << "\t把大象放入冰箱" << endl;
10 }
11
12 void Close_the_Door() {
13     cout << "\t关闭冰箱门" << endl;
14 }
15
16 void Put_Elephant_to_Refrigerator() {
17     Open_the_Door();
18     Put_Exlephant();
19     Close_the_Door();
20 }
```