

### 

课 程 设 计

**课程名称:** C++课程设计

**设计课题:**  订单管理系统设计

**指导教师:** 龚文辉

**总评成绩：**

**专业：** 软件工程  **班级：** 2202074

**姓名：** 曾日闽  **学号：** 220207435

**二 O 二 四 年 六 月 三 十 日**

目录

[**前言** 1](#_Toc4454)

[第一章 课设目标及内容和要求 2](#_Toc10510)

[1.1 课设目标 2](#_Toc12392)

[1.2 课程设计内容和要求 2](#_Toc27590)

[1.2.1 需求分析 2](#_Toc8242)

[1.2.2 系统功能 2](#_Toc17782)

[1.2.3 设计要求 3](#_Toc29186)

[第二章 课设功能模块组成及流程图 4](#_Toc11144)

[2.1 功能模块 4](#_Toc17640)

[第三章 课设主要数据结构及函数列表 5](#_Toc23713)

[3.1 数据结构 5](#_Toc2263)

[3.2 函数列表 6](#_Toc3400)

[第四章 课设代码及运行结果 8](#_Toc31533)

[4.1 程序代码 8](#_Toc15494)

[4.2 运行结果 23](#_Toc23218)

[第五章 课程设计总结与体会 28](#_Toc4298)

[第六章 参考文献 29](#_Toc21400)

前言

随着信息技术的飞速发展，企业对订单管理的效率和准确性要求越来越高。一个高效、准确的订单信息管理系统不仅能够帮助企业快速响应市场变化，还能提高企业的管理水平和服务质量。本课程设计旨在开发一个基于计算机的订单信息管理系统，以满足现代企业管理的需求。

C++是一种使­­­­­用非常广泛的程序设计语言。它是一种静态数据类型检查的，支持多范型的通用程序设计语言。C++支持过程化程序设计、数据抽象化、面向对象程序设计、泛型程序设计、基于原则设计等多种程序设计风格。

本系统利用Lightly线上IDE来进行代码编写，Git进行项目版本管理，实现了订单系统订单的添加，订单的浏览，订单条件查询，订单删除，订单统计，对订单进行文件读写等功能，该系统GitHub地址: https://github.com/ZRMYDYCG。

本系统主要有1个类：订单（Order）类。该系统已进行全面的系统测试，能够很好的运行、基本达到预期效果。

1. 课设目标及内容和要求

**[1.1 课设目标](#_Toc20897)**

[要求学生达到熟练掌握C++语言的基本知识和技能；基本掌握C++课程设计的思想和方法；能够利用所学的基本知识和技能，解决简单的面向对象程序设计问题，从而提高动手编程解决实际问题的能力。](#_Toc20897)

[1．通过本课程设计，培养上机动手编程能力，使学生巩固《C++程序设计》课程学习的内容，掌握面向对象程序设计语言里面类的定义，对象等的知识点的强化，强化上机动手能力，闯过编程关；](#_Toc20897)

[2．为后续各门计算机课程的学习打下坚实基础。](#_Toc20897)

**[1.2 课程设计内容和要求](#_Toc9298)**

**[1.2.1 需求分析](#_Toc7843)**

订单信息管理系统是为了简化和自动化企业订单处理流程而设计的系统。该系统旨在提供一个集中的平台，以方便地管理订单地各个关键信息，包括订单编号、公司名称、产品名称、产品定价、订购数量和订单金额等。

**[1.2.2 系统功能](#_Toc8692)**

1. 用户能够输入订单地基本信息，系统计算订单金额并保存订单记录；
2. 查询订单：

1）能够根据订单编号确查订单信息；

2）能够根据订单名称查询订单信息；

3）能够根据公司名称查询订单信息；

1. 根据订单进行金额统计；
2. 能够进行订单地删除；
3. 能够进行订单数量的追加；
4. 能够进行订单文件的读写；

**[1.2.3 设计要求](#_Toc9443)**

（1）应用系统分析，建立该系统的功能模块框图以及界面的组织和设计；

（2）分析系统中的各个实体及它们之间的关系；

（3）根据问题描述，设计系统的类层次；

（4）完成类层次中各个类的描述；

（5）完成类中各个方法的定义；

（6）完成系统的应用模块；

（7）功能调试；

1. 课设功能模块组成及流程图

**[2.1 功能模块](#_Toc4134)**

订单信息管理系统主要包含订单信息的录入保存，显示全部订单信息，查询某条订单信息，统计订单金额，订单信息，追加订单数量，退出系统。功能模块如图2.1所示。

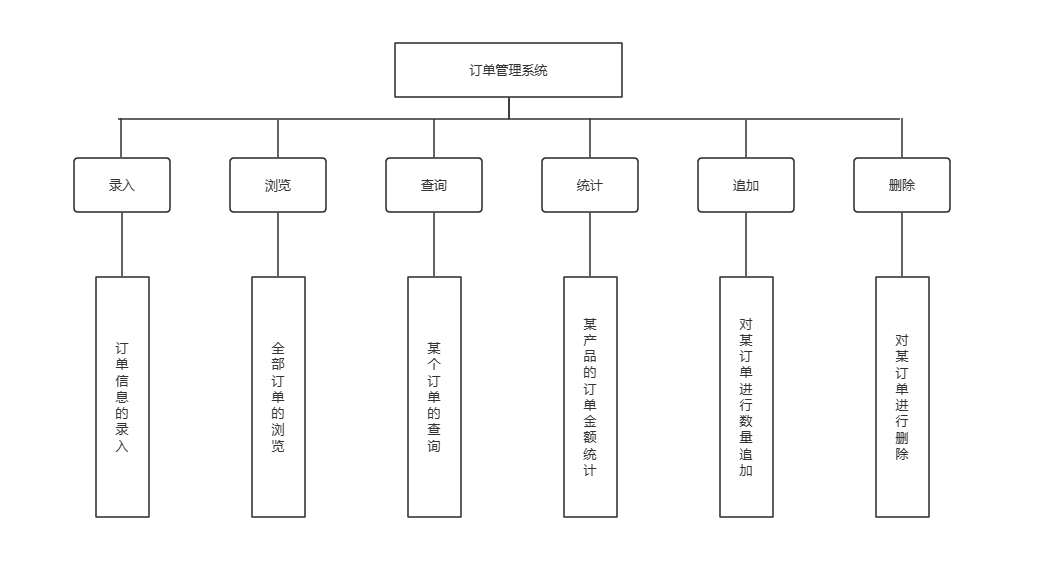


图 2.1 功能模块

[**2.2 流程图**](#_Toc6274) **如图2.2所示**

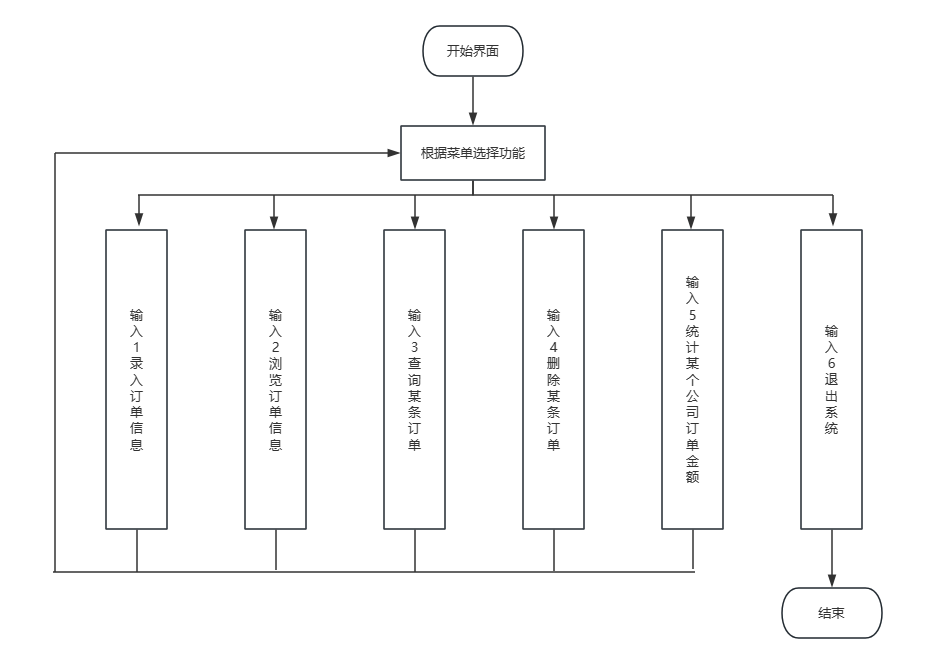


图 2.2 流程图

1. 课设主要数据结构及函数列表

**[3.1 数据结构](#_Toc13749)**

本系统共有一类：Order类

class Order {

private:

char order\_id[20];

char company\_name[50];

char product\_name[50];

double product\_price;

int order\_quantity;

double order\_amount;

public:

void add\_order() {

cout << "请输入订单编号：";

cin >> order\_id;

cout << "请输入公司名称：";

cin >> company\_name;

cout << "请输入产品名称：";

cin >> product\_name;

cout << "请输入产品定价：";

cin >> product\_price;

cout << "请输入订购数量：";

cin >> order\_quantity;

order\_amount = product\_price \* order\_quantity;

cout << "订单金额为：" << order\_amount << endl;

}

void display\_order() {

cout << "订单编号：" << order\_id << endl;

cout << "公司名称：" << company\_name << endl;

cout << "产品名称：" << product\_name << endl;

cout << "产品定价：" << product\_price << endl;

cout << "订购数量：" << order\_quantity << endl;

cout << "订单金额：" << order\_amount << endl;

}

char\* get\_order\_id() {

return order\_id;

}

char\* get\_company\_name() {

return company\_name;

}

double get\_order\_amount() {

return order\_amount;

}

void write\_order() {

ifstream fin("orders.txt");

ofstream fout("temp.txt");

bool found = false;

string line;

while (getline(fin, line)) {

istringstream iss(line);

string id, cname, pname;

double price;

int quantity;

iss >> id >> cname >> pname >> price >> quantity;

if (id == order\_id && cname == company\_name && pname == product\_name) {

found = true;

quantity += order\_quantity;

fout << id << " " << cname << " " << pname << " " << price << " " << quantity << " " << price \* quantity << endl;

} else {

fout << line << endl;

}

}

fin.close();

if (!found) {

fout << order\_id << " " << company\_name << " " << product\_name << " " << product\_price << " " << order\_quantity << " " << order\_amount << endl;

}

fout.close();

remove("orders.txt");

rename("temp.txt", "orders.txt");

}

void read\_order() {

ifstream fin("orders.txt");

while (fin >> order\_id >> company\_name >> product\_name >> product\_price >> order\_quantity >> order\_amount) {

display\_order();

cout << endl;

}

fin.close();

}

void search\_order(char\* id) {

ifstream fin("orders.txt");

bool found = false;

while (fin >> order\_id >> company\_name >> product\_name >> product\_price >> order\_quantity >> order\_amount) {

if (strcmp(order\_id, id) == 0) {

display\_order();

found = true;

break;

}

}

if (!found) {

cout << "未找到该订单！" << endl;

}

fin.close();

}

void delete\_order(char\* id) {

ifstream fin("orders.txt");

ofstream fout("temp.txt");

bool found = false;

while (fin >> order\_id >> company\_name >> product\_name >> product\_price >> order\_quantity >> order\_amount) {

if (strcmp(order\_id, id) != 0) {

fout << order\_id << " " << company\_name << " " << product\_name << " " << product\_price << " " << order\_quantity << " " << order\_amount << endl;

} else {

found = true;

}

}

fin.close();

fout.close();

remove("orders.txt");

rename("temp.txt", "orders.txt");

if (found) {

cout << "订单已删除！" << endl;

} else {

cout << "未找到该订单！" << endl;

}

}

void total\_order\_amount(char\* company) {

ifstream fin("orders.txt");

double total = 0;

while (fin >> order\_id >> company\_name >> product\_name >> product\_price >> order\_quantity >> order\_amount) {

if (strcmp(company\_name, company) == 0) {

total += order\_amount;

}

}

fin.close();

cout << "公司 " << company << " 的所有订单总额为：" << total << endl;

}

};

**[3.2 函数列表](#_Toc30870)**

/\*\*

\* 添加订单信息

\* @description 收集用户输入的订单信息，并计算订单金额。

\* @complexity 时间复杂度：O(1)，仅涉及基本的输入输出操作。

\*/

void add\_order() {}

/\*\*

\* 显示订单详情

\* @description 打印订单的所有详细信息。

\* @complexity 时间复杂度：O(1)，输出操作。

\*/

void display\_order() {}

/\*\*

\* 获取订单编号

\* @return char\* 返回订单编号的字符串。

\* @description 返回订单编号的指针。

\* @complexity 时间复杂度：O(1)。

\*/

char\* get\_order\_id() {}

/\*\*

\* 获取公司名称

\* @return char\* 返回公司名称的字符串。

\* @description 返回公司名称的指针。

\* @complexity 时间复杂度：O(1)。

\*/

char\* get\_company\_name() {}

/\*\*

\* 获取订单金额

\* @return double 返回订单的总金额。

\* @description 返回计算后的订单金额。

\* @complexity 时间复杂度：O(1)。

\*/

double get\_order\_amount() {}

/\*\*

\* 写入订单到文件

\* @description 将订单信息追加到文本文件orders.txt中。

\* @complexity 时间复杂度：O(1)，假设文件写入操作是常数时间。

\*/

void write\_order() {}

/\*\*

\* 从文件读取并显示所有订单

\* @description 从orders.txt文件中读取所有订单信息，并逐条打印。

\* @complexity 时间复杂度：O(n)，n为文件中的订单数量。

\*/

void read\_order() {}

/\*\*

\* 根据订单编号搜索订单

\* @param id 要搜索的订单编号。

\* @description 在orders.txt文件中搜索订单编号，并显示匹配的订单信息。

\* @complexity 时间复杂度：O(n)，n为文件中的订单数量。

\*/

void search\_order(char\* id) {}

/\*\*

\* 根据订单编号删除订单

\* @param id 要删除的订单编号。

\* @description 从orders.txt文件中删除指定的订单编号。

\* @complexity 时间复杂度：O(n)，n为文件中的订单数量。

\*/

void delete\_order(char\* id) {}

/\*\*

\* 统计指定公司的订单总额

\*/

void total\_order\_amount(char\* company) {}

1. 课设代码及运行结果

**[4.1 程序代码](#_Toc26637)**

#include <iostream>

#include <fstream>

#include <sstream>

#include <cstring>

using namespace std;

class Order {

private:

char order\_id[20];

char company\_name[50];

char product\_name[50];

double product\_price;

int order\_quantity;

double order\_amount;

public:

/\*\*

\* 添加订单信息

\* @description 收集用户输入的订单信息，并计算订单金额。

\* @complexity 时间复杂度：O(1)，仅涉及基本的输入输出操作。

\*/

void add\_order() {

cout << "请输入订单编号：";

cin >> order\_id;

cout << "请输入公司名称：";

cin >> company\_name;

cout << "请输入产品名称：";

cin >> product\_name;

cout << "请输入产品定价：";

cin >> product\_price;

cout << "请输入订购数量：";

cin >> order\_quantity;

order\_amount = product\_price \* order\_quantity;

cout << "订单金额为：" << order\_amount << endl;

}

/\*\*

\* 显示订单详情

\* @description 打印订单的所有详细信息。

\* @complexity 时间复杂度：O(1)，输出操作。

\*/

void display\_order() {

cout << "订单编号：" << order\_id << endl;

cout << "公司名称：" << company\_name << endl;

cout << "产品名称：" << product\_name << endl;

cout << "产品定价：" << product\_price << endl;

cout << "订购数量：" << order\_quantity << endl;

cout << "订单金额：" << order\_amount << endl;

}

/\*\*

\* 获取订单编号

\* @return char\* 返回订单编号的字符串。

\* @description 返回订单编号的指针。

\* @complexity 时间复杂度：O(1)。

\*/

char\* get\_order\_id() {

return order\_id;

}

/\*\*

\* 获取公司名称

\* @return char\* 返回公司名称的字符串。

\* @description 返回公司名称的指针。

\* @complexity 时间复杂度：O(1)。

\*/

char\* get\_company\_name() {

return company\_name;

}

/\*\*

\* 获取订单金额

\* @return double 返回订单的总金额。

\* @description 返回计算后的订单金额。

\* @complexity 时间复杂度：O(1)。

\*/

double get\_order\_amount() {

return order\_amount;

}

/\*\*

\* 写入订单到文件

\* @description 将订单信息追加到文本文件orders.txt中。

\* @complexity 时间复杂度：O(1)，假设文件写入操作是常数时间。

\*/

void write\_order() {

ifstream fin("orders.txt");

ofstream fout("temp.txt");

bool found = false;

string line;

while (getline(fin, line)) {

istringstream iss(line);

string id, cname, pname;

double price;

int quantity;

iss >> id >> cname >> pname >> price >> quantity;

if (id == order\_id && cname == company\_name && pname == product\_name) {

found = true;

quantity += order\_quantity;

fout << id << " " << cname << " " << pname << " " << price << " " << quantity << " " << price \* quantity << endl;

} else {

fout << line << endl;

}

}

fin.close();

if (!found) {

fout << order\_id << " " << company\_name << " " << product\_name << " " << product\_price << " " << order\_quantity << " " << order\_amount << endl;

}

fout.close();

remove("orders.txt");

rename("temp.txt", "orders.txt");

}

/\*\*

\* 从文件读取并显示所有订单

\* @description 从orders.txt文件中读取所有订单信息，并逐条打印。

\* @complexity 时间复杂度：O(n)，n为文件中的订单数量。

\*/

void read\_order() {

ifstream fin("orders.txt");

while (fin >> order\_id >> company\_name >> product\_name >> product\_price >> order\_quantity >> order\_amount) {

display\_order();

cout << endl;

}

fin.close();

}

/\*\*

\* 根据订单编号搜索订单

\* @param id 要搜索的订单编号。

\* @description 在orders.txt文件中搜索订单编号，并显示匹配的订单信息。

\* @complexity 时间复杂度：O(n)，n为文件中的订单数量。

\*/

void search\_order(char\* id) {

ifstream fin("orders.txt");

bool found = false;

while (fin >> order\_id >> company\_name >> product\_name >> product\_price >> order\_quantity >> order\_amount) {

if (strcmp(order\_id, id) == 0) {

display\_order();

found = true;

break;

}

}

if (!found) {

cout << "未找到该订单！" << endl;

}

fin.close();

}

/\*\*

\* 根据订单编号删除订单

\* @param id 要删除的订单编号。

\* @description 从orders.txt文件中删除指定的订单编号。

\* @complexity 时间复杂度：O(n)，n为文件中的订单数量。

\*/

void delete\_order(char\* id) {

ifstream fin("orders.txt");

ofstream fout("temp.txt");

bool found = false;

while (fin >> order\_id >> company\_name >> product\_name >> product\_price >> order\_quantity >> order\_amount) {

if (strcmp(order\_id, id) != 0) {

fout << order\_id << " " << company\_name << " " << product\_name << " " << product\_price << " " << order\_quantity << " " << order\_amount << endl;

} else {

found = true;

}

}

fin.close();

fout.close();

remove("orders.txt");

rename("temp.txt", "orders.txt");

if (found) {

cout << "订单已删除！" << endl;

} else {

cout << "未找到该订单！" << endl;

}

}

/\*\*

\* 统计指定公司的订单总额

\* @param company 要统计的公司名称。

\* @description 统计并打印指定公司所有订单的总额。

\* @complexity 时间复杂度：O(n)，n为文件中的订单数量。

\*/

void total\_order\_amount(char\* company) {

ifstream fin("orders.txt");

double total = 0;

while (fin >> order\_id >> company\_name >> product\_name >> product\_price >> order\_quantity >> order\_amount) {

if (strcmp(company\_name, company) == 0) {

total += order\_amount;

}

}

fin.close();

cout << "公司 " << company << " 的所有订单总额为：" << total << endl;

}

};

/\*\*

\* 主函数

\* @description 提供用户界面，允许用户选择不同的订单管理操作。

\* @complexity 时间复杂度：O(1)，主循环中的操作依赖于用户输入。

\*/

int main() {

Order order;

int choice;

char id[20], company[50];

while (true) {

cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 主菜单 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"<< endl;

cout << "\* \*"<< endl;

cout << "\* 请选择操作： \*"<< endl;

cout << "\* \*"<< endl;

cout << "\* 1. 添加订单 \*"<< endl;

cout << "\* 2. 浏览订单 \*"<< endl;

cout << "\* 3. 查询订单 \*"<< endl;

cout << "\* 4. 删除订单 \*"<< endl;

cout << "\* 5. 统计订单总额 \*"<< endl;

cout << "\* 6. 退出 \*"<< endl;

cout << "\* \*"<< endl;

cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"<< endl;

cin >> choice;

switch (choice) {

case 1:

order.add\_order();

order.write\_order();

break;

case 2:

order.read\_order();

break;

case 3:

cout << "请输入要查询的订单编号：";

cin >> id;

order.search\_order(id);

break;

case 4:

cout << "请输入要删除的订单编号：";

cin >> id;

order.delete\_order(id);

break;

case 5:

cout << "请输入要统计的公司名称：";

cin >> company;

order.total\_order\_amount(company);

break;

case 6:

return 0;

default:

cout << "输入错误，请重新输入！" << endl;

break;

}

}

return 0;

}

[**4.2 运行结果**](#_Toc28985)

运行界面图：如图4.1所示

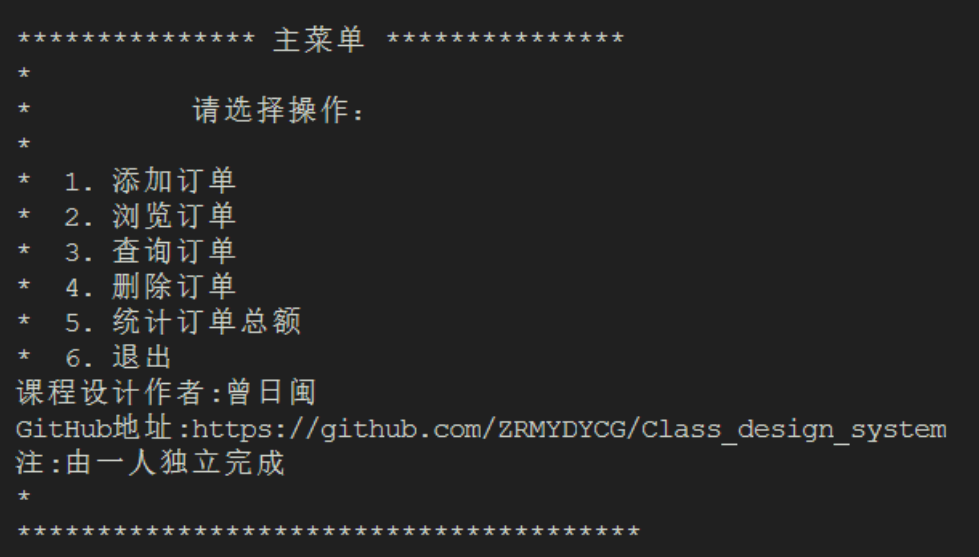


图 4.1运行界面图

功能一：输入订单信息（订单编号，公司名称，产品名称，产品定价，订购数量，邮箱）并且保存入orders.txt文件中。如图4.2所示

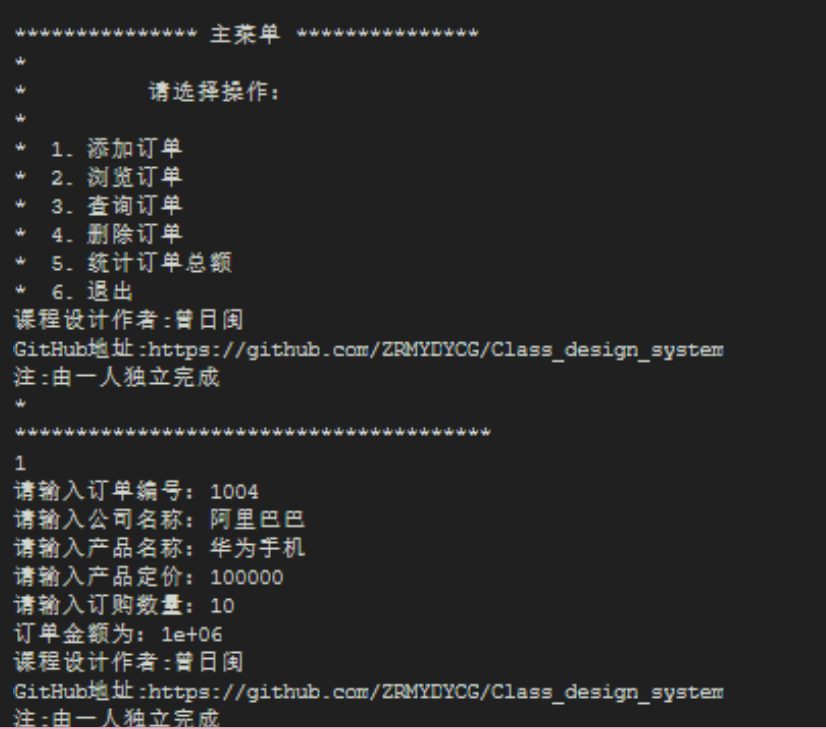


图 4.2 功能一效果图

功能二：浏览订单。如图4.3所示

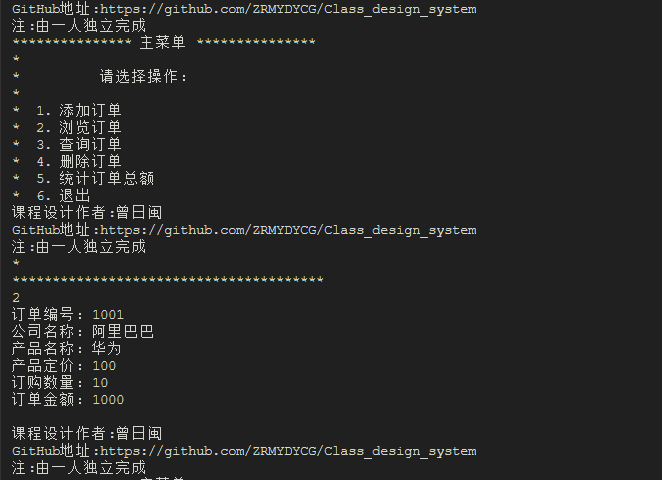


图 4.3 功能二效果图

功能三：查询订单信息。如图4.4所示

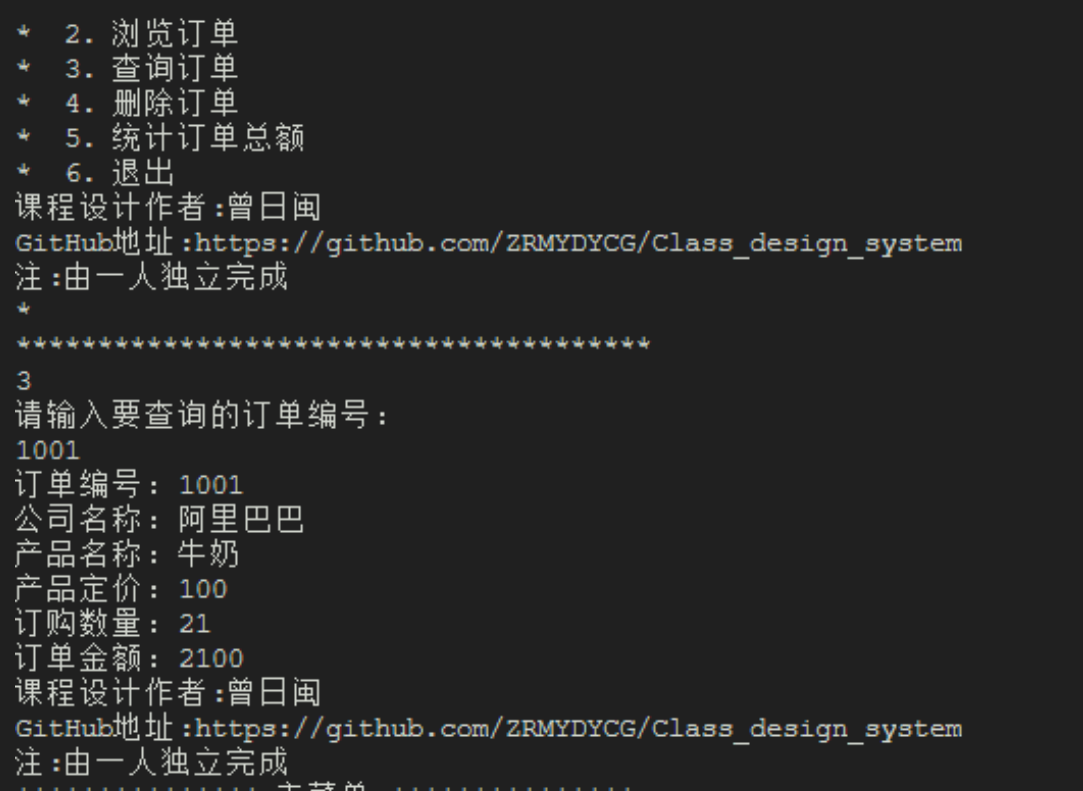


图 4.4 功能三效果图

功能四：删除订单。如图4.4所示



图 4.5 功能四效果图

功能五：统计订单总额。如图4.6所示



图 4.6 功能五效果图

文件读写情况：如图4.7所示

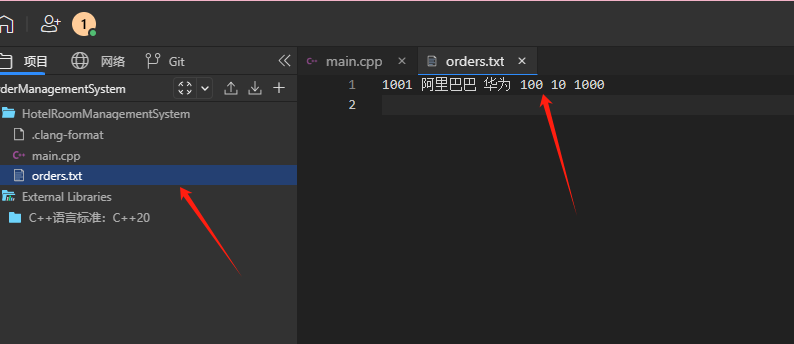


图 4.7 文件读写情况

1. 课程设计总结与体会

在这次课程设计中，我们首先对系统的整体功能进行了构思，然后用结构化分析方法进行分析，将整个系统清楚的划分为几个模块，再根据每个模块的功能编写代码。尽可能的将模块细分，同时，我们也认为上机实验是学习程序设计语言必不可少的实践环节，特别是C++语言灵活、简洁，更需要通过编程的实践来真正掌握它。对于程序设计语言的学习目的，可以概括为学习语法规定、掌握程序设计方法、提高程序开发能力，这些都必须通过充分的实际上机操作才能完成。

在学习中，也对C++语言有更加的了解。在整个编程过程中，可以说遇到了很多难题，在书和同学的帮助下最后解决了这些问题,我觉得编写程序需要很大的耐心，当然细心也很需要，在编程的过程中有很多错误都是自己的粗心造成的，所以我明白了做事需要认真再认真。这次的课程设计也是一次很好的对自我的检查，使我受益匪浅！

1. 参考文献
2. 谭浩强.C++面向对象程序设计[M].3版.北京:清华大学出版社,2020.
3. 吴乃陵,况迎辉,李海文著.VC++程序设计[M].北京：高等教育出版社,2003.8.
4. Leen Ammeraal . C +＋程序设计教程［ M ]，刘瑞挺，等译．3版北京：中国铁道出版社，2003.
5. Csdn-专业开发者社区.