**C++课程设计实验**

**实验题目：订单信息管理系统**

**姓名：叶璇**

**学号：201806060825**

**班级：健行1802班**

**时间：2019年6月10日**

1. **实验题目**

基本要求

订单信息包括：订单编号、公司名称、产品名称、订单定价、订购数量、订单金额等。

功能要求

1. 订单以菜单方式工作；
2. 添加：输入各信息并计算订单金额；修改订单信息；删除某份不需要的订单；
3. 信息浏览及查询；按某个条件查询订单；
4. 统计：输入某个公司名称，统计并输出所有订单总额；
5. 对所有订单信息读写文件。
6. 运行环境

软件环境

编译器：GCC-6.3.0-1

编辑器：Sublime Text 3.2.1(Build 3207)

硬件环境

电脑：联想ThinkPad S1 2017

CPU：第七代智能英特尔®酷睿™i5-7200U处理器(2.5GHz睿频至3.1GHz,3MB)

操作系统：Windows 10 家庭版

远程git仓库

https://github.com/TuDouuuuu/Homework-Course-Design-of-C-Programming-Design/tree/homework\_ver

1. 实验课题分析

需求分析

1. 对公司，需要进行访问，修改名称。
2. 对产品，需要进行访问，修改名称，修改单价。
3. 对订单，需要进行访问，修改公司和产品，修改数量，删除，用已知单价和数量计算订单金额。
4. 对链表，能够从头到尾进行访问。

结构功能分析

需要三个类：公司，产品，订单。

共三个模块：控制模块，具体操作模块，异常处理模块。

1. 实验的主要模块、流程图

头文件设计

公司类Company.h

#ifndef \_HEADER\_COMPANY\_

#define \_HEADER\_COMPANY\_

#include <string>

using std::string;

class Company{

protected:

string name;

public:

Company(string n="No Name");//构造函数

void ReName(string n);//修改公司名字

string ReturnName();//返回公司名字

};

#endif

产品类product.h：公司中的name可以继承给产品用故产品public继承了公司类。

#ifndef \_HEADER\_PRODUCT\_

#define \_HEADER\_PRODUCT\_

#include"Company.h"

class Product:public Company{

private:

int prize;

public:

Product(string n="No Name",int p=0);//构造函数

void RePrize(int p);//修改产品价格

void Re(string n,int p);//修改产品名和产品价格

int ReturnPrize();//返回产品价格

};

#endif

订单类order.h：订单中含有公司和产品两项元素。

#ifndef \_HEADER\_ORDER\_

#define \_HEADER\_ORDER\_

#include"Product.h"

#include"Company.h"

#include"ex.h"

#include<string>

#include<iostream>

#include<cstring>

#include<iomanip>

using namespace std;

class Order{

private:

int no;

Company c;Product p;

int num;int sum;

Order \*next=NULL;

void display(bool x);

void connectNode(Order \*a,Order \*b);

int str\_int(string x);

public:

Order(int nono,string n1="No Name",string n2="No Name",int p=0,int n=0);

Order\* createFromFile(Order\* x,int nono,string n1="No Name",string n2="No Name",int p=0,int n=0);

Order\* createFromKeyboard(int nono);

void allDisplay();

void search();

void del();

void revise();

void companyCnt();

string ReturnCompanyName();

string ReturnProductName();

int ReturnProductPrize();

int ReturnNo();

int ReturnNum();

Order\* ReturnNext();

};

#endif

处理异常的类ex.h

#ifndef \_HEADER\_EX\_

#define \_HEADER\_EX\_

class ex{

public:

virtual string getWhat()=0;

};

class exOp:public ex{

public:

string getWhat(){return "Unkown Operation.";}

};

class exNull:public ex{

public:

string getWhat(){return "Not Find";}

};

#endif

功能细节设计

公司类company.cpp：要求能够返回名字和修改名字

#include"Company.h"

//构造函数

Company::Company(string n){

name=n;

}

//操作函数

void Company::ReName(string n){

name=n;

}

//返回函数读取数值

string Company::ReturnName(){

return name;

}

产品类product.cpp：要求能够修改产品的价格和名字，能够返回产品的价格，继承了公司类的能返回名字。

#include"Product.h"

//构造函数

Product::Product(string n,int p){

name=n;prize=p;

}

//操作系列

void Product::RePrize(int p){//修改产品价格

prize=p;

}

void Product::Re(string n,int p){//修改价格和名字

name=n;prize=p;

}

//返回数据系列

int Product::ReturnPrize(){//返回

return prize;

}

订单类order.cpp：订单以链表的形式组织起来。要求能够进行打印操作，创建新节点操作，遍历操作，查询操作，删除操作。每次进行操作都从根结点进行遍历。

#include"order.h"

//构造函数

Order::Order(int nono,string n1,string n2,int pp,int n){

no=nono;

c.ReName(n1);p.ReName(n2);p.RePrize(pp);num=n;sum=n\*pp;

}

//辅助函数

void Order::connectNode(Order \*a,Order \*b){

a->next=b;

}

void Order::display(bool x){//0是格式输出

if(x)cout<<setiosflags(ios::left)<<"|"<<setw(15)<<no<<"|"<<setw(15)<<c.ReturnName()<<"|"<<setw(15)<<p.ReturnName()<<"|"<<setw(15)<<p.ReturnPrize()<<"|"<<setw(15)<<num<<"|"<<setw(15)<<sum<<"|"<<endl;

else cout<<setiosflags(ios::left)<<"|"<<setw(15)<<"Order ID"<<"|"<<setw(15)<<"Company's Name"<<"|"<<setw(15)<<"Product's Name"<<"|"<<setw(15)<<"Prize"<<"|"<<setw(15)<<"Quantity"<<"|"<<setw(15)<<"Sum"<<"|"<<endl;

}

int Order::str\_int(string s){//string型转int型

int len=s.length();

int ans=0;

for(int i=len-1;i>=0;i--){

ans=ans\*10+s[i]-'0';

}

return ans;

}

//操作函数集：

//创造新节点的函数

Order\* Order::createFromFile(Order\* x,int nono,string n1,string n2,int p,int n){

Order\* v=new Order(nono,n1,n2,p,n);

next=v;

return v;

}

Order\* Order::createFromKeyboard(int nono){//,string n1,string n2,int p,int n){

string s1,s2;int p,n;//读入数据暂存使用

cout<<"Please input company's name:";cin>>s1;

cout<<"Please input product's name:";cin>>s2;

cout<<"Please input product's prize:";cin>>p;

cout<<"Please input product's number:";cin>>n;

Order\* v=new Order(nono,s1,s2,p,n);

next=v;

return v;

}

//打印所有结点

void Order::allDisplay(){

Order\* u=next;

cout<<"There are results:"<<endl;

display(0);//cout<<setiosflags(ios::left)<<"|"<<setw(15)<<"Order ID"<<"|"<<setw(15)<<"Company's Name"<<"|"<<setw(15)<<"Product's Name"<<"|"<<setw(15)<<"Prize"<<"|"<<setw(15)<<"Quantity"<<"|"<<setw(15)<<"Sum"<<"|"<<endl;

while(u!=NULL){

u->display(1);

u=u->next;

}

}

//查询函数

void Order::search(){

Order \*u=next;

//获取修改的参数

int op;string n;

//menu

cout<<" ===================================================="<<endl;

cout<<" Welcome to the order search system!!"<<endl;

cout<<" 1.Abuut the ID of the order"<<endl;

cout<<" 2.About the name of the company"<<endl;

cout<<" 3.About the name of the product"<<endl;

cout<<" 4.About the prize of the product"<<endl;

cout<<" ===================================================="<<endl;

cout<<"Please input the charastic you want to search:";cin>>op;

cout<<"Please input the parameter of the charastic:";cin>>n;

switch(op){

case 1://-N

cout<<"There are results:"<<endl;

display(0);//cout<<setiosflags(ios::left)<<"|"<<setw(15)<<"Order ID"<<"|"<<setw(15)<<"Company's Name"<<"|"<<setw(15)<<"Product's Name"<<"|"<<setw(15)<<"Prize"<<"|"<<setw(15)<<"Quantity"<<"|"<<setw(15)<<"Sum"<<"|"<<endl;

while(u!=NULL){

if(u->ReturnNo()==str\_int(n)){

u->display(1);break;

}

u=u->next;

}

break;

case 2://-C

cout<<"There are results:"<<endl;

display(0);//cout<<setiosflags(ios::left)<<"|"<<setw(15)<<"Order ID"<<"|"<<setw(15)<<"Company's Name"<<"|"<<setw(15)<<"Product's Name"<<"|"<<setw(15)<<"Prize"<<"|"<<setw(15)<<"Quantity"<<"|"<<setw(15)<<"Sum"<<"|"<<endl;

while(u!=NULL){

if(u->ReturnCompanyName()==n)u->display(1);

u=u->next;

}

break;

case 3://-PN

cout<<"There are results:"<<endl;

display(0);//cout<<setiosflags(ios::left)<<"|"<<setw(15)<<"Order ID"<<"|"<<setw(15)<<"Company's Name"<<"|"<<setw(15)<<"Product's Name"<<"|"<<setw(15)<<"Prize"<<"|"<<setw(15)<<"Quantity"<<"|"<<setw(15)<<"Sum"<<"|"<<endl;

while(u!=NULL){

if(u->ReturnProductName()==n)u->display(1);

u=u->next;

}

break;

case 4://-PP

cout<<"There are results:"<<endl;

display(0);//cout<<setiosflags(ios::left)<<"|"<<setw(15)<<"Order ID"<<"|"<<setw(15)<<"Company's Name"<<"|"<<setw(15)<<"Product's Name"<<"|"<<setw(15)<<"Prize"<<"|"<<setw(15)<<"Quantity"<<"|"<<setw(15)<<"Sum"<<"|"<<endl;

while(u!=NULL){

if(u->ReturnProductPrize()==str\_int(n))u->display(1);

u=u->next;

}

break;

default:throw exNull();

}

//以下为魔改命令行版

/\*if(op=="-C"){//查询公司名称

cout<<"There are results:"<<endl;

display(0);//cout<<setiosflags(ios::left)<<"|"<<setw(15)<<"Order ID"<<"|"<<setw(15)<<"Company's Name"<<"|"<<setw(15)<<"Product's Name"<<"|"<<setw(15)<<"Prize"<<"|"<<setw(15)<<"Quantity"<<"|"<<setw(15)<<"Sum"<<"|"<<endl;

while(u!=NULL){

if(u->ReturnCompanyName()==n)u->display(1);

u=u->next;

}

}

else if(op=="-PN"){//查询产品名称

cout<<"There are results:"<<endl;

display(0);//cout<<setiosflags(ios::left)<<"|"<<setw(15)<<"Order ID"<<"|"<<setw(15)<<"Company's Name"<<"|"<<setw(15)<<"Product's Name"<<"|"<<setw(15)<<"Prize"<<"|"<<setw(15)<<"Quantity"<<"|"<<setw(15)<<"Sum"<<"|"<<endl;

while(u!=NULL){

if(u->ReturnProductName()==n)u->display(1);

u=u->next;

}

}

else if(op=="-PP"){//查询产品单价

cout<<"There are results:"<<endl;

display(0);//cout<<setiosflags(ios::left)<<"|"<<setw(15)<<"Order ID"<<"|"<<setw(15)<<"Company's Name"<<"|"<<setw(15)<<"Product's Name"<<"|"<<setw(15)<<"Prize"<<"|"<<setw(15)<<"Quantity"<<"|"<<setw(15)<<"Sum"<<"|"<<endl;

while(u!=NULL){

if(u->ReturnProductPrize()==str\_int(n))u->display(1);

u=u->next;

}

}

else if(op=="-N"){//查询某个编号

cout<<"There are results:"<<endl;

display(0);//cout<<setiosflags(ios::left)<<"|"<<setw(15)<<"Order ID"<<"|"<<setw(15)<<"Company's Name"<<"|"<<setw(15)<<"Product's Name"<<"|"<<setw(15)<<"Prize"<<"|"<<setw(15)<<"Quantity"<<"|"<<setw(15)<<"Sum"<<"|"<<endl;

while(u!=NULL){

if(u->ReturnNo()==str\_int(n)){

u->display(1);break;

}

u=u->next;

}

}

else if(op=="-D"){//输出所有

cout<<"There are results:"<<endl;

display(0);//cout<<setiosflags(ios::left)<<"|"<<setw(15)<<"Order ID"<<"|"<<setw(15)<<"Company's Name"<<"|"<<setw(15)<<"Product's Name"<<"|"<<setw(15)<<"Prize"<<"|"<<setw(15)<<"Quantity"<<"|"<<setw(15)<<"Sum"<<"|"<<endl;

while(u!=NULL){

u->display(1);

u=u->next;

}

}

else{

cout<<"Unknow opeartion!!"<<endl;

}\*/

}

//删除函数

void Order::del(){

//获取修改的参数

int x,op;string n;

cout<<"Please know the ID of order first!"<<endl;

cout<<"Please input the ID of the order:";cin>>x;

//确认

while(1){

string flag;

cout<<"Are you sure?[Y/N]";cin>>flag;

if(flag=="N")return ;

else if(flag=="Y")break;

else cout<<"For Y is yes,N is no."<<endl;

}

Order\* v=NULL;

Order\* u=next;

while(u!=NULL){

if(u->ReturnNo()==x){

v->next=u->next;

delete u;

return ;

}

v=u;

u=u->next;

}

throw exNull();

//其它魔改功能

/\*if(op=="-C"){//删除公司名称

while(u!=NULL){

if(u->ReturnCompanyName()==n){

v->next=u->next;

delete u;

u=u->next;

}

}

}

else if(op=="-PN"){//删除产品名称

while(u!=NULL){

if(u->ReturnProductName()==n){

v->next=u->next;

delete u;

u=u->next;

}

}

}

else if(op=="-PP"){//删除产品单价

while(u!=NULL){

if(u->ReturnProductPrize()==str\_int(n)){

v->next=u->next;

delete u;

u=u->next;

}

}

}

else if(op=="-N"){//删除某个编号

while(u!=NULL){

if(u->ReturnNo()==str\_int(n)){

v->next=u->next;

delete u;

u=u->next;

}

}

}

else{

cout<<"Unknow opeartion!!"<<endl;

}\*/

}

//修改订单信息

void Order::revise(){

//获取修改的参数

int x,op;string n;

cout<<"Please know the ID of order first!"<<endl;

cout<<"Please input the ID of the order:";cin>>x;

//menu

cout<<" ===================================================="<<endl;

cout<<" Welcome to the order revise system!!"<<endl;

cout<<" 1.Change the name of the company"<<endl;

cout<<" 2.Change the name of the product"<<endl;

cout<<" 3.Change the prize of the product"<<endl;

cout<<" 4.Change the Quantity of the product"<<endl;

cout<<" ===================================================="<<endl;

cout<<"Please input the place you want to change:";cin>>op;

cout<<"Please input the parameter you want to change to:";cin>>n;

//确认

while(1){

string flag;

cout<<"Are you sure?[Y/N]";cin>>flag;

if(flag=="N")return ;

else if(flag=="Y")break;

else cout<<"For Y is yes,N is no."<<endl;

}

Order \*u=next;

while(u!=NULL){

//cout<<"u->ReturnNo()="<<u->ReturnNo()<<endl;

if(u->ReturnNo()==x){

switch(op){

case 1:u->c.ReName(n);break;

case 2:u->p.ReName(n);break;

case 3:u->p.RePrize(str\_int(n));u->sum=u->p.ReturnPrize()\*u->num;break;

case 4:u->num=str\_int(n);u->sum=u->p.ReturnPrize()\*u->num;break;

default:cout<<"Unknow opeartion!!"<<endl;

}

return ;

//以下为魔改命令行组件

/\*if(op=="-C")u->c.ReName(n);

else if(op=="-PN")u->p.ReName(n);

else if(op=="-PP")u->p.RePrize(str\_int(n));

else cout<<"Unknow opeartion!!"<<endl;

return ;\*/

}

u=u->next;

}

throw exNull();

}

void Order::companyCnt(){

string temp;cout<<"Enter the company's name:";

cin>>temp;

int res=0;

Order \*u=next;

while(u!=NULL){

if(u->ReturnCompanyName()==temp){

u->display(1);

res+=u->ReturnProductPrize()\*u->ReturnNum();

}

u=u->next;

}

if(res==0)throw exNull();

else cout<<"The sum is:"<<res<<endl;

}

//返回值的函数

string Order::ReturnCompanyName(){

return c.ReturnName();

}

string Order::ReturnProductName(){

return p.ReturnName();

}

int Order::ReturnProductPrize(){

return p.ReturnPrize();

}

int Order::ReturnNo(){

return no;

}

int Order::ReturnNum(){

return num;

}

Order\* Order::ReturnNext(){

return next;

}

控制模块由main.cpp实现

1. main先读入已经储存好的txt文件，创建链表。
2. 循环显示菜单，输入需要进行的操作。若输入不正确则抛出异常。
3. 若输入1，则为创建新节点选项；输入2，更改订单信息；输入3，打印所有订单；输入4，根据所给信息查找订单；输入5删除具体编号的订单；输入6，通过给定公司名输出所有订单及统计订单总额；输入7，保存并退出。

#include<fstream>

#include"company.h"

#include"product.h"

#include"order.h"

#include"ex.h"

void menu(){//输出菜单

cout<<endl;

cout<<" ===================================================="<<endl;

cout<<" Welcome to the order manage system!!"<<endl;

cout<<" 1.Add a new order"<<endl;

cout<<" 2.Change an order's information"<<endl;

cout<<" 3.Display all orders"<<endl;

cout<<" 4.Find some orders"<<endl;

cout<<" 5.Delete an order"<<endl;

cout<<" 6.Count a company's order"<<endl;

cout<<" 7.Save and exit"<<endl;

cout<<" ===================================================="<<endl;

cout<<"What opearation do you want to do:";

}

int main(){

cout<<"Loading..."<<endl;

ifstream inFile;

inFile.open("..\\\\data\_example\\data.txt");

//构建链表的头（空的

static int cnt=0;

Order root(cnt);cnt++;

string s1,s2;int temp,p,n,s;//读入数据暂存使用

//读入已有数据

Order\* u=&root;

while(inFile>>temp>>s1>>s2>>p>>n){

u=u->createFromFile(u,temp,s1,s2,p,n);

cnt=temp+1;

}

inFile.close();

//开始进行菜单操作

string op;

int flag=0;

while(1){

try{menu();

cin>>op;

if(op.length()>1)throw exOp();

if(op[0]<'1'||op[0]>'7')throw exOp();

int opp;opp=op[0]-'0';

switch(opp){

case 1:u=u->createFromKeyboard(cnt);cnt++;break;

case 2:root.revise();break;

case 3:root.allDisplay();break;

case 4:root.search();break;

case 5:root.del();break;

case 6:root.companyCnt();break;

case 7:flag=1;break;

default:throw exOp();

}

if(flag)break;

menu();

}

catch(ex& e){

cout<<e.getWhat()<<endl;//menu();

}

}

//输出文件用：

ofstream outFile;

outFile.open("..\\\\data\_example\\data.txt");

u=&root;u=u->ReturnNext();

while(u!=NULL){

outFile<<u->ReturnNo()<<'\t'<<u->ReturnCompanyName()<<'\t'<<u->ReturnProductName()<<'\t'<<u->ReturnProductPrize()<<'\t'<<u->ReturnNum()<<endl;

u=u->ReturnNext();

}

outFile.close();

//ofstream cout("..\\\\data\_example\\test1.txt");

}

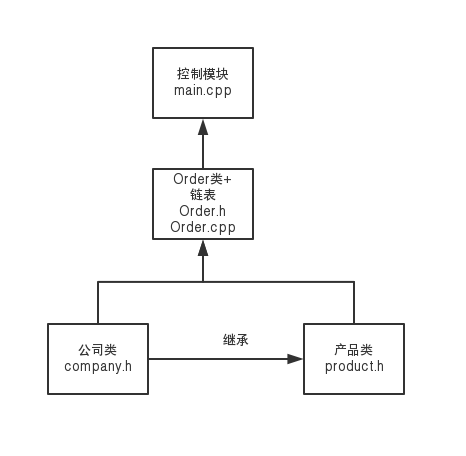


图 1工程文件组织

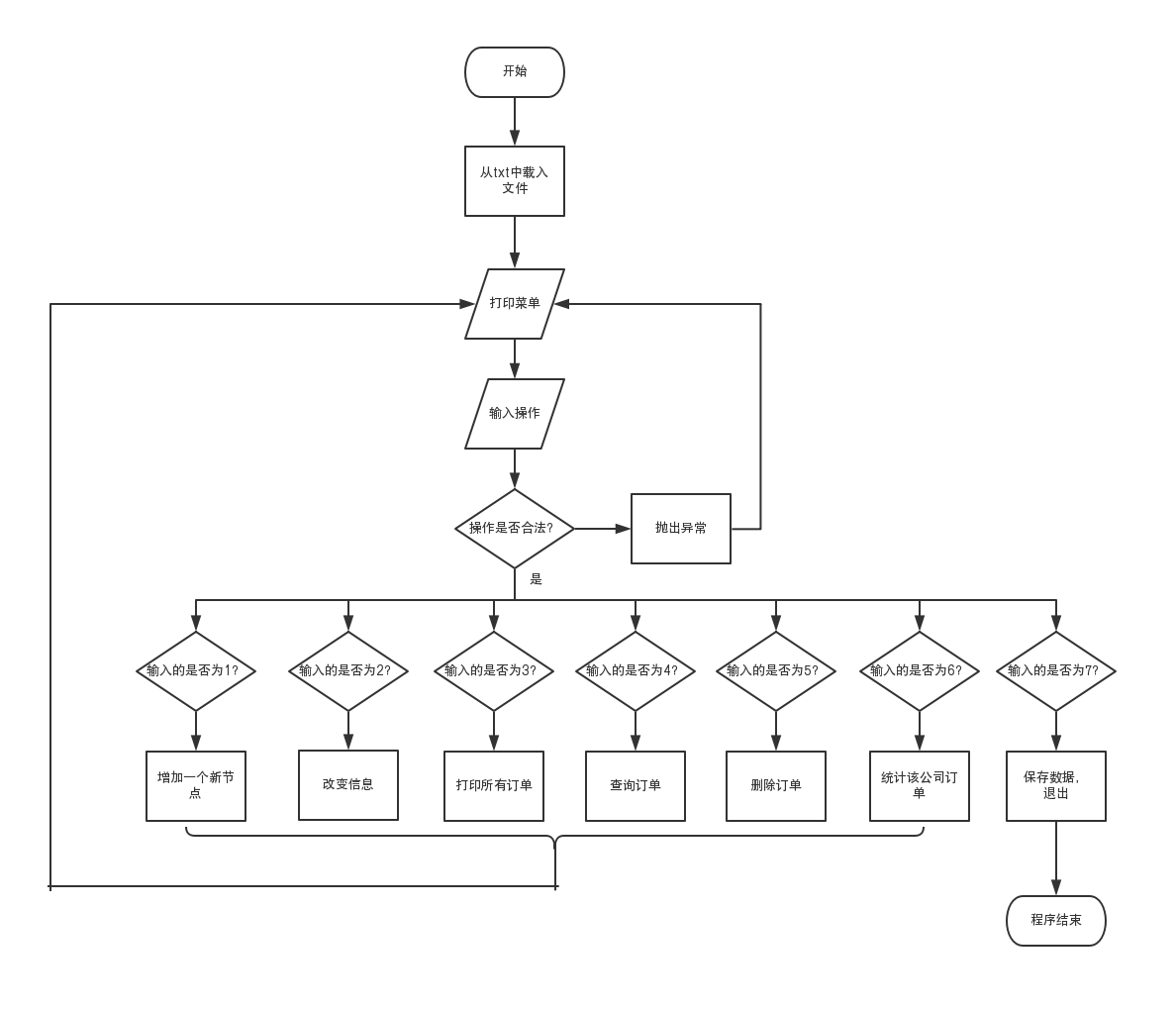


图 2流程图

1. 实验调试、测试、运行记录及分析

实验调试

实验文件的构建顺序为company.h->company.cpp->product.h->product.cpp->order.h->order.cpp->main.cpp.学习异常处理后新增ex.h文件。

实验调试日记



测试数据

测试数据由人工手动输入，保存在data\_example中

1 cyz M\_E 32 8

2 cyz Daisychu 5 13

3 zka Hugin 250 250

4 hwh nb 10 27

7 cyz makerspace 300000 1

8 tudou poi 3123 123

1. 实验总结

优点

增加了异常处理功能，能够处理用户的一些异常输入内容，增加了程序的稳定性。

采用面向对象的编程，既能够保证数据的私密性，增加数据的安全，又能进行访问，设计类与类之间的关系，节省编程所需时间。

通过采用链表的方式，给每个结点进行一个编号，通过类似数据库处理的方式，重复的编号不会出现。

不足

因为开始时间较早，知识掌握不足，在处理方式上不够便捷。如果使用抽象类构建上层能够使得程序主体更加清晰。未将结点和链表进行分离，使得Order类读起来需要一些想象能力。

收获及体会

通过本次实验实践了学期所学的面向对象化编程，体会到了面向对象编程的意义——对数据更加安全，也更加具有结构感。

通过本次实验也增强了对数据结构：链表的理解。上学期通过结构体实现链表，这学期通过类来实现链表。

本次实验在进行算法设计上有些不足，有寻求帮助。

通过本次实验，也增强了利用互联网解决问题的能力。

对课程的看法与体会

课程通过分类，给每个同学不同的任务，既可以相互交流，又保证了多样性，非常有趣。课程。课程任务单描述清楚，也方便了功能的实现。

附录

注：实验全部的源代码在每个文件夹下。输入输出数据也在各个.txt文件中。