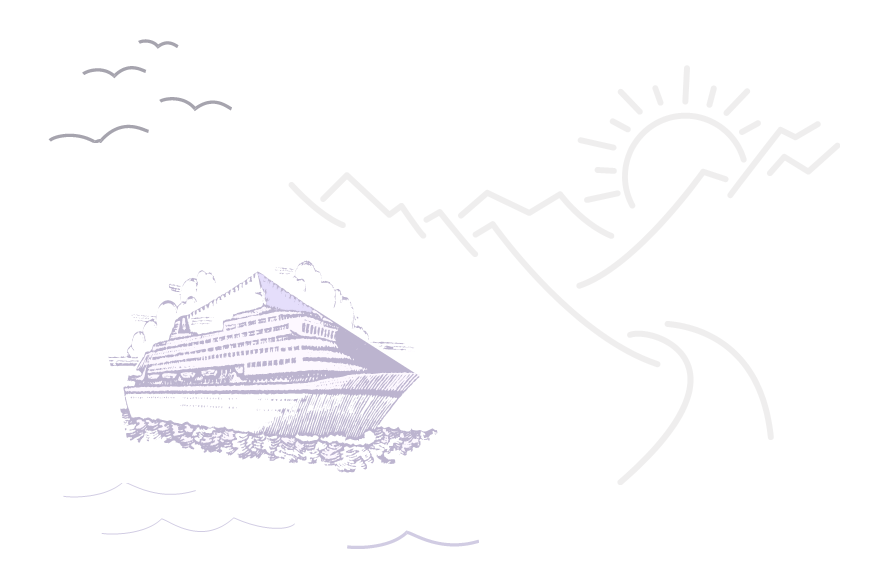






**专业班级：计算机82班**

**课 题：快递管理系统**

**小组成员：高浩翔 学号：2181411962**

**薛杰锋 学号：2181411984**

**施炎江 学号：2186113847**

**2019.10.17**

# 题目翻译：

作为一家大型国际快递公司的雇员，比尔·威利几乎每天都要与运送和接收包裹的公司接触。他经常被问到是否可以在同一天递送当地的包裹。几个月后，他注意到，他所在的城市对快递服务的需求似乎很大。他决定成立自己的快递公司，并当场付诸实践。比尔先把他的手机号码列在黄页上。他还向所有那些要求当日速递服务的公司寄信，提供他之前的公司无法提供的服务。他希望，通过良好的服务和广告增长业务。他还开始了其他的广告和营销活动，以促进他的服务。起初，比尔通过他的商务手机收到了送货请求。然而，没过多久，他的客户就开始询问他是否有一个网站可以让他们下订单发货。他意识到如果他能在网上露面，他就能增加他的曝光率，促进他的生意发展。比尔刚开业几个月，就发现他需要更多的人手。他雇了另一个人来帮忙递送和取包裹。看到业务增长是件好事，但多一个人也增加了协调提货和配送的复杂性。随着一个新人的加入，他再也不能把他的货车当做出货的仓库。他现在需要一个中央仓库，以便他组织和分发包裹。他认为，如果他的业务支撑得起，他还需要有人在仓库协调所有包裹的到达和分发。

此处描述了包裹取件和递送过程的细节。当比尔接到订单时，起初，他只在电话上记录何时接到订单，何时可以取货。有时，顾客要求立即提货;有时，他们会打电话约定当天晚些时候来取车。比尔一到取货地点，就开始收包裹。对于客户来说，有几个包裹要投递是很常见的。除了送货地点的名称和地址，他还记录了取货的时间。他指出了所需的交货时间、交货地点和包裹的重量，以确定快递成本。当他拿起包裹时，他用便携式打印机打印了一个标签，并把它放在了送货车上。一开始，比尔要求客户在取货的时候付款，但是他很快发现有些老客户更喜欢收到他们所有货物的月度账单。他希望能够满足这些客户，在票据在收到时付款。为了帮助跟踪所有的包裹，比尔认为他需要扫描每个包裹，因为它在仓库里被分类。这将使他能够很好地控制他的包裹，避免丢失或延误。递送包裹相当简单在交付时，他将记录关于何时交付以及谁收到了交付的信息。因为有些包是有价值的，所以在这些情况下需要有人为包签名

在进行快递服务过程中有很多改变。一开始，比尔用他的手机接收到取包裹的请求，并将这些信息记录在日志中，然后在当天晚些时候开车去取所有的包裹。然而，他很快发现，他很难与另一个员工协调货车的使用。没过多久，他就重组了公司，把仓库员工变成了司机。然后，他自己呆在仓库里，由他的两名员工负责提货和送货。这很有效，因为他可以更好地控制和协调货车和交货。对他来说，坐在办公桌前接受取货请求也比开着货车更容易。

当比尔想到他的生意是如何发展的以及他为顾客提供的服务时，他开始逐条列出他需要维护的各种信息。

当然，他需要维护客户的信息。他的一些客户是企业;一些人的个人。他需要知道每个客户的基本地址和联系方式。此外，对于他的企业客户，他需要确定一个主要联系人。主要是他的企业客户希望每月收到一份清单，列出当月的所有发货和总成本。比尔需要区分哪些顾客付了现金，哪些顾客想要每月结帐单。事实上，对于那些按月付款的人，他需要对诸如最后一次付款的时间、以及任何未付余额等事项保持流水账户。最后，当收到付款时，无论是单独的发货还是每月的发票，他需要记录关于付款的信息:付款类型、日期和金额。虽然这不是一个复杂的帐单和支付系统，比尔认为它会满足他的需要。

接下来，他开始考虑他的包裹和运输。在收到要求提货的请求时，他需要将其作为某种类型的送货请求或送货单进行跟踪。此时，比尔最需要知道的是客户是谁，取货地点在哪里，以及准备取货的日期和时间。他还记录了收到订单的日期和时间。送货单被认为是“开放的”，直到货车到达取货地点，所有包裹都被取走。此时，交付订单全部完成。

一旦检索到包，就需要惟一地标识每个包。比尔需要知道是什么时候取走的，是哪个送货员取走的。其他重要信息是“发送到”实体名称和地址。他还需要确定交付的类型。有些包裹是高优先级的，需要当天送达，有的人可以隔夜送达。当然，重量和成本都记录下来了，这样客户就可以要么付款，要么把它加到每月的票据上。

在快递业务中，最重要的信息需求之一是日期和时间戳。对于每一个包裹，重要的是要知道它是什么时候被取走的，什么时候到达仓库的，什么时候返回，什么时候交付。在可能的情况下，将这些事件的名称关联起来也很重要。

随着快递服务的不断发展，比尔发现如果他利用一些现有的技术，他可以为客户提供更好的服务。例如，它将使他与他的送货卡车保持频繁的通信，这可以节省运输和劳动力成本，使取货和送货操作更有效率。这将使他能更好地为顾客服务。当然，需要一个更复杂的系统，但是比尔的开发顾问向他保证可以开发一个简单而不太复杂的解决方案。

接下来是比尔希望他的企业如何运作。每辆卡车将有上午和下午的送货和取货。每个驱动程序都有一个带触摸屏的便携式数字设备。司机将能够看到他或她预定的皮卡和交付运行。(注意:这个过程将需要一个新的用例—敏捷开发方法预测会发生的事情。)然而，由于卡车将通过电话互联网接入与家庭办公室保持频繁联系，因此可以实时更新取货/送货时间表——即使在运行期间也是如此。与其保持经常的联系，比尔决定只要数字设备在取货或送货时与家庭办公室同步就足够了。在这些时间点，可以用适当的信息更新路线时间表。

在此之前，客户可以打电话到现场要求领取包裹，也可以访问该公司的网站安排领取包裹。一旦客户登录，他们可以进入一个网页，允许他们输入关于每个包裹的信息，包括“收件人”地址、大小和重量类别信息，以及请求的服务类型。现场提供“三小时”、“当日”、“通宵”服务。为了方便客户自助服务，在现场不需要精确的重量和尺寸，但有预定义的尺寸和重量类别，客户可以从中选择。

一旦客户输入了所有包裹的信息，系统就会计算成本，然后打印邮寄标签和收据。根据所要求的服务类型和送货卡车的远近，该系统将安排立即接机或当天晚些时候接机。它将显示此信息，以便客户立即知道何时需要取货。

挑选包是一个相当简单的过程。但是根据系统中的信息以及包裹是否已经贴了标签，会发生什么变化呢?到达预定的取货地点后，司机会让系统显示该客户可用的任何包裹信息。如果系统已经有了包上的信息，驱动程序将简单地验证包在系统中的信息是否正确。驱动程序还可以进行诸如纠正地址、删除包或添加新包等更改。如果这是一个现金客户，司机将收集任何钱并输入系统。司机可以用车上的便携式打印机为顾客打印收据。如果有新的包不在系统中，驱动程序将输入所需的信息，并使用便携式打印机打印邮件标签。

客户需要的另一项服务是能够跟踪他们的包裹的交付状态。系统需要跟踪一个包的状态，从它第一次发现这个包到包被交付。“准备取货”、“取货”、“到达仓库”、“送出”和“已送”等状态非常重要。通常，一个包会遵循所有的状态，但是由于调度和交付过程的复杂性，一个包有时会在相同的交付运行中被挑选和交付。比尔还决定为那些计划取走但最终没有发送的包裹添加“已取消”状态。

# 题目解析

在整个快递系统中，有来自客户的各种要求，也有送货人在送货过程发现的各种问题。具体要求实现的功能如下

一．送货人：

1. 交货时间、交货地点和包裹的重量，以确定快递成本；
2. 将记录关于何时交付以及谁收到了交付的信息；
3. 需要知道每个客户的基本地址和联系方式；
4. 判断支付类型并跟踪订单；
5. 客户是谁，取货地点在哪里，以及准备取货的日期和时间；
6. 惟一地标识每个包，什么时候取走，交付的类型以及客户优先级；

二、客户方：

1、选择支付方式

2、填写包裹类型

三、包裹：

1、包含自身来自客户的和送货人的各项属性

# 设计思路

根据不同要求及其涉及到的主体，同时迎合C++类和对象的编程理念，我们分了送货方，客户方，以及包裹，并据此安排任务，完成不同主体下的操作。每个模块完成一个特性的子功能，把着些模块结合起来组成一个整体，逐一实现各种功能。在不同模块，即不同的类之间通过友元加强其联系。

包裹类，正如名称一般，包裹是物，拥有自身属性但却是不能进行自主操作的，对包裹的操作都通过客户和送货方来实现，因此包裹类中含有与包裹相关的各项属性，同时声明客户和送货员都是其友元，可以对它进行操作。

用户类，最基本的一点，用户是人，同样有着各项和快递服务相关的属性，除此之外，客户有着更强的自主性，作为快递服务的的接受者，他也有着和服务提供方送货员的共同点，首先，他们都得在快递服务中有一个身份，即快递管理系统中的ID，证明他们是参与者，因此二者都有与登录注册相关的友元函数。而他作为快递服务中的对立方，二者自然有着不同之处，众所周知，顾客是上帝，因此顾客类拥有各种选择的权利，比如结账方式上的现结或者生成账单，根据顾客选择的不同，将会有不同的应答，其中自然包括对包裹的操作。

送货人，这个群体当然也有着自己的信息，但是在快递服务中，他们的信息并不具备什么实质性的作用，所以这个类不需要基本属性，用现在流行的话，他不配拥有姓名。但是，作为服务的一方，没有姓名，也得劳作，他包括了各种各样的操作，既要对包裹操作，也要对用户负责，因此他需要进行的操作有包裹的修改和查看，还要跟踪交易过程，仅有的与自己相关的过程也是查看自己负责的包裹，还是与其他类紧密联系。

三个大类基本确定，主函数主要面向有自主性的两大主体：客户及送货方。通过操作者的不同选择执行相应的操作。

# 源代码

#include<iostream>

using namespace std;

int main();

class pack;

class staff;

class Client;// 包裹 客户 员工三大类提前声明

struct running\_account//记录交易流水的结构体

{

string object\_name;//交易名称

double price;//运费

string type\_of\_payment;//付款方式

string pay\_date;//付款日期

bool ispaid;//用来判断是否付款的标记

};

class pack

{

private:

int pack\_id;//本包裹的id

int id;//寄件人的id

string recipient;//收件人姓名

string recipient\_contact;//收件人电话

string recipient\_add;//收件人地址

string pick\_add;

int weight;//重量

int price;//价格

int pack\_state;// 包裹状态 0：公司尚未揽收； 1：以揽收，正在运送；2：已经签收尚未付款；3：已经签收，交易完成

string pick\_time;//揽收时间

string sign\_time;//签收时间

int deliveryman;//送货员编号 0/1

public:

friend class staff;

friend class Client;//将员工与客户声明为包裹的友元

pack()

{id=0;} //无参数的构造函数

pack(int f,int idd,string a,string b,string c,string d,string e);//含参构造函数

void display();//展示包裹信息

};

pack::pack(int f,int idd,string a,string b,string c,string d,string e)

{

pack\_id=f;

id=idd;

recipient=a;

recipient\_contact=b;

recipient\_add=c;

pick\_add=e;

pick\_time=d;

pack\_state=0;

}

void pack::display()//展示包裹信息

{

cout<<"------------------------------------------------------------------------"<<endl;

cout<<"包裹id为："<<pack\_id<<endl;

cout<<endl;

cout<<"收件人姓名："<<recipient<<endl;

cout<<endl;

cout<<"收件人地址："<<recipient\_add<<endl;

cout<<endl;

cout<<"收件人联系方式："<<recipient\_contact<<endl;

cout<<endl;

switch(pack\_state) //根据包裹状态不同输出不同的信息

{

case(0):

cout<<"公包裹目前状态：公司尚未揽收"<<endl;

cout<<endl;

break;

case(1):

cout<<"包裹重量："<<weight<<endl;

cout<<endl;

cout<<"包裹费用："<<price<<endl;

cout<<endl;

cout<<"包裹目前状态：已经揽收，正在运送"<<endl;

cout<<endl;

cout<<"送货员为";

if(deliveryman==0)

cout<<"张三"<<endl;

else

cout<<"李四"<<endl;

break;

case(2):

cout<<"包裹重量："<<weight<<endl;

cout<<endl;

cout<<"包裹费用："<<price<<endl;

cout<<endl;

cout<<"包裹目前状态：已经签收,尚未付款"<<endl;

cout<<endl;

cout<<"送货员为";

if(deliveryman==0)

{

cout<<"张三"<<endl;

}

else

{

cout<<"李四"<<endl;

}

break;

case(3):

cout<<"包裹重量："<<weight<<endl;

cout<<endl;

cout<<"包裹费用："<<price<<endl;

cout<<endl;

cout<<"包裹目前状态：已经签收，交易完成"<<endl;

cout<<endl;

cout<<"送货员为";

if(deliveryman==0)

{

cout<<"张三"<<endl;

}

else

{

cout<<"李四"<<endl;

}

break;

}

cout<<"------------------------------------------------------------------------"<<endl;

}

pack obj\_pack[100000];//可以容纳100000包裹的对象数组

class Client//客户类

{

private:

int ID;//客户id

string name;//客户名字（可以是公司名字或是个人名字）

string basic\_address;//客户地址

string contact\_information;//联系方式

running\_account account[10000];//可以存放10000条交易流水的流水账

int running\_account\_number;//记录流水账数目

string password;

int ownpack[100];//拥有包裹的id

public:

friend void log\_in();

friend void id\_register();

friend void client\_interface(int k);

friend class pack;

Client();

Client(int id,string n,string add,string con,string m);//构造函数

int operator ==(const int a);//重载运算符用于判断对象数组是否有元素

string get\_name();//获得客户名称

void add\_running\_account(string n,double p);//添加一条流水账

void finish\_running\_account(string n, string fun,string d);//完成一条流水 n：交易名称 fun：交易方式 d：交易日期

void show\_running\_account();//展示客户流水账

void add\_pack(int k);//添加包裹

void display\_pack();//显示自己的包裹信息

};

Client::Client()

{

name="张三";

basic\_address="西安交通大学" ;

contact\_information="10000000";

password=1111111;

}

Client::Client(int id,string n,string add,string con,string m)//构造函数

{

ID=id;

name=n;

basic\_address=add;

contact\_information=con;

password=m;

running\_account\_number=0;

for(int jishu=0;jishu<=99;jishu++)

{

ownpack[jishu]=0;

}

}

int Client::operator ==(const int a)//重载运算符用于判断对象数组是否有元素

{

if(name=="张三"&&contact\_information=="10000000" )

return 1;

else

return 0;

}

string Client::get\_name()//获得客户名称

{

return name;

}

void Client::add\_running\_account(string n,double p)//添加一条流水账

{

account[running\_account\_number].object\_name=n;

account[running\_account\_number].price=p;

account[running\_account\_number].ispaid=false;

running\_account\_number++;

}

void Client::finish\_running\_account(string n, string fun,string d)//完成一条流水 n：交易名称 fun：交易方式 d：交易日期

{

for(int i=0;i<running\_account\_number;i++)

if(account[i].object\_name==n)

{

account[i].ispaid=true;

account[i].type\_of\_payment=fun;

account[i].pay\_date=d;

}

}

void Client::show\_running\_account()//展示客户流水账

{

for(int i=0;i<running\_account\_number;i++)

{

cout<<"交易名称："<<account[i].object\_name<<" "<<"运费"<<account[i].price<<" ";

if(account[i].ispaid==true)

cout<<"付款日期："<<account[i].pay\_date<<"付款方式："<<account[i].type\_of\_payment<<endl;

else

cout<<"未付款"<<endl;

}

cout<<"------------------------------------------------------------------------"<<endl;

cout<<"共"<<running\_account\_number<<"条流水账"<<endl;

}

void Client::add\_pack(int k)//添加包裹

{

int pid;

for(pid=1;pid<=100000;pid++)

{

if(obj\_pack[pid].id==0)

break;

}

string a;

string b;

string c;

string d;

string e;

string delimiter;

getline(cin,delimiter);

cout<<"请输入收件人姓名："<<endl;

getline(cin,a);

cout<<endl;

cout<<"请输入收件人联系方式："<<endl;

getline(cin,b);

cout<<endl;

cout<<"请输入收件人地址："<<endl;

getline(cin,c);

cout<<endl;

cout<<"请输入您期待的上门取货时间："<<endl;

getline(cin,d);

cout<<endl;

cout<<"请输入您期待的取货地址："<<endl;

getline(cin,e);

cout<<endl;

obj\_pack[pid]=pack(pid,k,a,b,c,d,e);

for(int y=0;y<=99;y++)

{

if(ownpack[y]==0)

{

ownpack[y]=pid;

break;

}

}

system("cls");

cout<<"------------------------------------------------------------------------"<<endl;

cout<<"添加包裹成功！"<<endl;

cout<<"------------------------------------------------------------------------"<<endl;

system("pause");

}

void Client::display\_pack()//显示自己的包裹信息

{

for(int u=0;u<=99;u++)

{

if(ownpack[u]==0)

{

break;

}

else

{

obj\_pack[ownpack[u]].display();

}

}

}

Client obj[100];//可以容纳100客户的对象数组

class staff

{

public:

staff(){}//无参构造函数

staff(int i,string a)//含参构造函数

{

staffid=i;

staff\_password=a;

}

void staff\_interface();//客户界面

void check\_pack();//查看未揽收的包裹

void amend\_pack(int i);//修改包裹信息

void check\_ownpack();//查看自己负责的包裹

void add\_running\_tally();//添加流水

void close\_deal();//完成交易

friend void log\_in();

friend int main();

private:

int staffid;//员工id

string staff\_password;//员工密码，默认123456

};

staff obj\_staff[2];//两个员工的对象数组

void staff::staff\_interface()

{

cout<<"------------------------------------------------------------------------"<<endl;

cout<<"尊敬的员工，您好！"<<endl;

cout<<endl;

cout<<"请选择您想要执行的操作:"<<endl;

cout<<endl;

cout<<"查看未揽收的包裹 请输入1"<<endl;

cout<<endl;

cout<<"查看我负责运送的包裹 请输入2"<<endl;

cout<<endl;

cout<<"更新某个包裹的信息 请输入3"<<endl;

cout<<endl;

cout<<"为某一客户添加一条流水 请输入4"<<endl;

cout<<endl;

cout<<"为某一客户完成一条流水 请输入5"<<endl;

cout<<endl;

cout<<"退出程序 请输入6"<<endl;

cout<<"------------------------------------------------------------------------"<<endl;

switch(getchar())

{

case('1'):

system("cls");//清屏操作 system函数

obj\_staff[staffid].check\_pack();

system("cls");

staff\_interface();

case('2'):

system("cls");

obj\_staff[staffid].check\_ownpack();

system("cls");

staff\_interface();

case('3'):

system("cls");

cout<<"请输入包裹id："<<endl;

int k;

cin>>k;

obj\_staff[staffid].amend\_pack(k);

system("cls");

getchar();

staff\_interface();

case('4'):

system("cls");

add\_running\_tally();

system("cls");

getchar();

staff\_interface();

case('5'):

system("cls");

close\_deal();

system("cls");

staff\_interface();

case('6'):

system("cls");

getchar();

system("cls");

main();

break;

default:

system("cls");

cout<<"输入错误 请输入正确的数字"<<endl;

cout<<endl;

cout<<endl;

system("pause");

staff\_interface();

}

}

void staff::add\_running\_tally()

{

cout<<"------------------------------------------------------------------------"<<endl;

cout<<"请输入用户id"<<endl;

int ki;

cin>>ki;

string a;

cout<<"请输入该流水涉及的包裹id"<<endl;

getchar();

getline(cin,a);

cout<<"请输入流水价钱"<<endl;

double jiaqian;

cin>>jiaqian;

obj[ki].add\_running\_account(a,jiaqian);

cout<<"------------------------------------------------------------------------"<<endl;

system("cls");

cout<<"------------------------------------------------------------------------"<<endl;

cout<<"流水信息更新成功！"<<endl;

cout<<"------------------------------------------------------------------------"<<endl;

system("pause");

}

void staff::close\_deal()

{

cout<<"------------------------------------------------------------------------"<<endl;

cout<<"请输入用户id"<<endl;

int ki;

cin>>ki;

string a;

string b;

string c;

cout<<"请输入包裹id"<<endl;

getchar();

getline(cin,a);

cout<<"请输入交易方式"<<endl;

getline(cin,b);

cout<<"请输入交易日期"<<endl;

getline(cin,c);

obj[ki].finish\_running\_account(a,b,c);

cout<<"------------------------------------------------------------------------"<<endl;

system("cls");

cout<<"------------------------------------------------------------------------"<<endl;

cout<<"流水信息更新成功！"<<endl;

cout<<"------------------------------------------------------------------------"<<endl;

system("pause");

}

void staff::check\_pack()

{

system("cls");

for(int i=1;i<=10000;i++)

{

if(obj\_pack[i].id==0)

{

cout<<"------------------------------------------------------------------------"<<endl;

cout<<"暂无包裹信息！"<<endl;

cout<<"------------------------------------------------------------------------"<<endl;

system("pause");

break;

}

else

{

if(obj\_pack[i].pack\_state==0)

{

obj\_pack[i].display();

}

}

}

char c;

getchar();

}

void staff::amend\_pack(int i)

{

system("cls");

cout<<"------------------------------------------------------------------------"<<endl;

cout<<"请选择您想要更新的数据:"<<endl;

cout<<"包裹送货员 请输入1"<<endl;

cout<<"包裹重量 请输入2"<<endl;

cout<<"包裹状态 请输入3"<<endl;

cout<<"包裹费用 请输入4"<<endl;

cout<<"返回 请输入5"<<endl;

cout<<"------------------------------------------------------------------------"<<endl;

getchar();

switch(getchar())

{

case('1'):

cout<<"------------------------------------------------------------------------"<<endl;

cout<<"请输入您的送货员id:"<<endl;

cin>>obj\_pack[i].deliveryman;

system("cls");

cout<<"------------------------------------------------------------------------"<<endl;

cout<<"数据更新成功！"<<endl;

cout<<"------------------------------------------------------------------------"<<endl;

system("pause");

break;

case('2'):

cout<<"------------------------------------------------------------------------"<<endl;

cout<<"请输入包裹重量:"<<endl;

cin>>obj\_pack[i].weight;

system("cls");

cout<<"------------------------------------------------------------------------"<<endl;

cout<<"数据更新成功！"<<endl;

cout<<"------------------------------------------------------------------------"<<endl;

system("pause");

break;

case('3'):

cout<<"------------------------------------------------------------------------"<<endl;

cout<<"请输入包裹新状态:"<<endl;

cout<<"已揽收，正在运送 输入1"<<endl;

cout<<"已签收，未付款 输入2"<<endl;

cout<<"已签收，交易完成 输入3"<<endl;

cout<<"------------------------------------------------------------------------"<<endl;

cin>>obj\_pack[i].pack\_state;

if(obj\_pack[i].pack\_state==1)

{

string delimiter;

cout<<"------------------------------------------------------------------------"<<endl;

cout<<"请输入揽收时间:"<<endl;

getline(cin,delimiter);

getline(cin,obj\_pack[i].pick\_time);

}

else

{

if(obj\_pack[i].pack\_state==2)

{

string delimiter1;

cout<<"------------------------------------------------------------------------"<<endl;

cout<<"请输入签收时间:"<<endl;

getline(cin,delimiter1);

getline(cin,obj\_pack[i].sign\_time);

}

else

{getchar();}

}

system("cls");

cout<<"------------------------------------------------------------------------"<<endl;

cout<<"数据更新成功！"<<endl;

cout<<"------------------------------------------------------------------------"<<endl;

system("pause");

break;

case('4'):

cout<<"------------------------------------------------------------------------"<<endl;

cout<<"请输入包裹费用:"<<endl;

cin>>obj\_pack[i].price;

system("cls");

cout<<"------------------------------------------------------------------------"<<endl;

cout<<"数据更新成功！"<<endl;

cout<<"------------------------------------------------------------------------"<<endl;

system("pause");

break;

case('5'):

amend\_pack(i);

default:

system("cls");

cout<<"------------------------------------------------------------------------"<<endl;

cout<<"命令无效 请输入正确的数字"<<endl;

cout<<"------------------------------------------------------------------------"<<endl;

system("pause");

}

}

void staff::check\_ownpack()

{

system("cls");

for(int i=1;i<=10000;i++)

{

if(obj\_pack[i].id==0)

{

cout<<"------------------------------------------------------------------------"<<endl;

cout<<"暂无包裹信息！"<<endl;

cout<<"------------------------------------------------------------------------"<<endl;

system("pause");

break;

}

else

{

if(obj\_pack[i].deliveryman==staffid)

{

obj\_pack[i].display();

}

}

}

getchar();

}

void client\_interface(int k)//客户主界面

{

cout<<"------------------------------------------------------------------------"<<endl;

cout<<"尊敬的客户，您好！"<<endl;

cout<<endl;

cout<<"请选择您想要执行的操作:"<<endl;

cout<<endl;

cout<<"我要寄件 请输入1"<<endl;

cout<<endl;

cout<<"查看正在运送的包裹 请输入2"<<endl;

cout<<endl;

cout<<"查看我的账单 请输入3"<<endl;

cout<<endl;

cout<<"退出程序 请输入4"<<endl;

cout<<"------------------------------------------------------------------------"<<endl;

switch(getchar())

{

case('1'):

system("cls");

obj[k].add\_pack(k);

client\_interface(k);

case('2'):

system("cls");

obj[k].display\_pack();

getchar();

client\_interface(k);

case('3'):

system("cls");

obj[k].show\_running\_account();

getchar();

client\_interface(k);

case('4'):

system("cls");

getchar();

main();

break;

default:

system("cls");

cout<<"------------------------------------------------------------------------"<<endl;

cout<<"命令无效 请输入正确的数字"<<endl;

cout<<"------------------------------------------------------------------------"<<endl;

system("pause");

}

}

void log\_in ()//登录界面

{

system("cls");

cout<<"------------------------------------------------------------------------"<<endl;

cout<<"请输入您的ID："<<endl;

int id;

string delimiter;

string mima;

cin>>id;

getline(cin,delimiter);

cout<<"请输入您的密码："<<endl;

getline(cin,mima);

if(obj[id].password==mima)

{

system("cls");

client\_interface(id);

}

else

{

if(obj\_staff[id].staff\_password==mima)

{

system("cls");

obj\_staff[id].staff\_interface();

}

else

{

system("cls");

cout<<"------------------------------------------------------------------------"<<endl;

cout<<"ID或密码错误，请重新登录"<<endl;

cout<<"------------------------------------------------------------------------"<<endl;

system("pause");

main();

}

}

}

void id\_register()//用户注册界面

{

string delimiter;

string name;

string address;

string contact;

string password1;

system("cls");

cout<<"------------------------------------------------------------------------"<<endl;

getline(cin,delimiter);

cout<<"请输入您的姓名："<<endl;

getline(cin,name);

cout<<endl;

cout<<"请输入您的住址："<<endl;

getline(cin,address);

cout<<endl;

cout<<"请输入您的电话号码："<<endl;

getline(cin,contact);

cout<<endl;

cout<<"请设置您的密码："<<endl;

getline(cin,password1);

int i=2;

for(i=2;i<100;i++)

{

if(obj[i]==0)

{

obj[i]=Client(i,name,address,contact,password1);

break;

}

}

system("cls");

cout<<"------------------------------------------------------------------------"<<endl;

cout<<"注册成功!您的ID是："<<i<<endl;

cout<<endl;

cout<<"请重新登录"<<endl;

cout<<"------------------------------------------------------------------------"<<endl;

system("pause");

log\_in ();

}

int main()//主函数

{

obj\_staff[0].staffid=1;

obj\_staff[1].staffid=1;

obj\_staff[0].staff\_password="123456";

obj\_staff[1].staff\_password="123456";

cout<<"------------------------------------------------------------------------"<<endl;

cout<<"欢迎来到无敌螺旋物流!!!"<<endl;

cout<<endl;

cout<<"请选择您想要执行的操作:"<<endl;

cout<<endl;

cout<<"登录已有帐号 请输入1"<<endl;

cout<<endl;

cout<<"注册账号 请输入2"<<endl;

cout<<endl;

cout<<" 版本v1.0.2"<<endl;

cout<<" 版权所有：XJTU 高浩翔 施炎江 薛杰锋"<<endl;

cout<<"------------------------------------------------------------------------"<<endl;

switch(getchar())

{

case('1'):

log\_in();

break;

case('2'):

id\_register();

break;

default:

system("cls");

cout<<"------------------------------------------------------------------------"<<endl;

cout<<"命令无效 请输入正确的数字"<<endl;

cout<<"------------------------------------------------------------------------"<<endl;

system("pause");

main();

}

return 0;

}