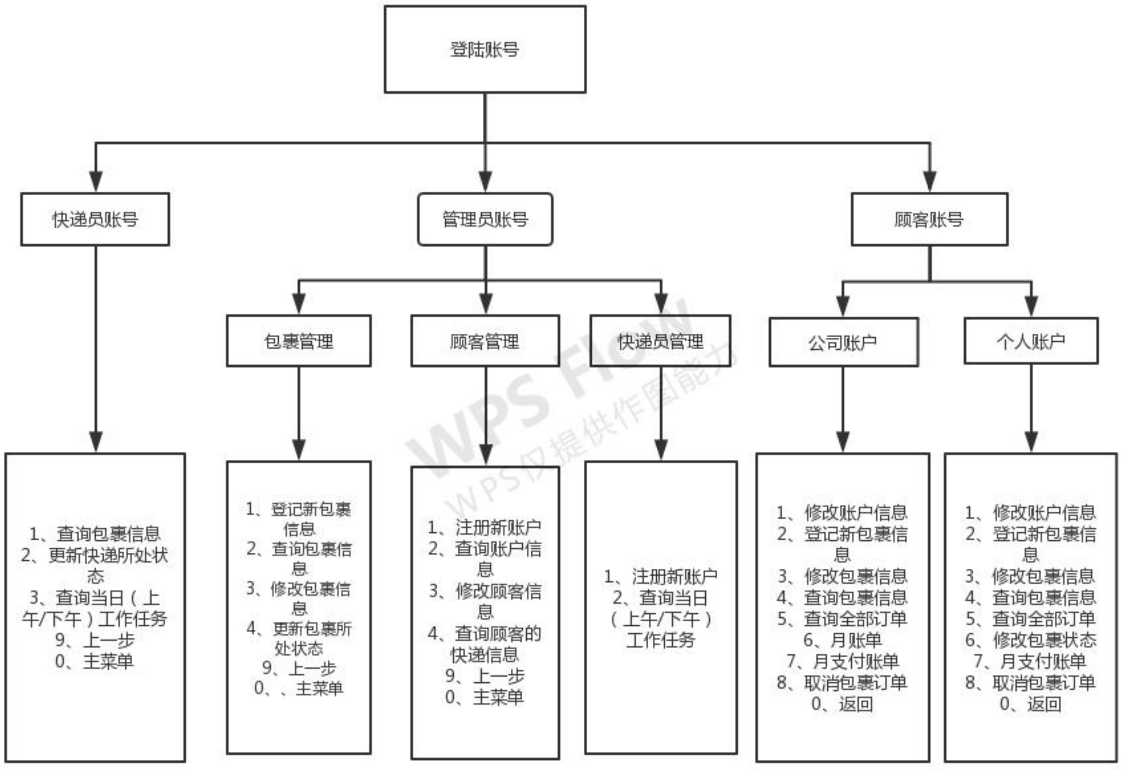
面向对象的程序设计

——大作业报告

报告人：

物联网 81	王旭	2186412635
物试 51	李昂	2150201074
物联网 81	皮家恺	2184214240

<table>
<tr><td>题目理解</td><td>
<p>题目要求创建一个程序，能够实现记录信息、修改信息和查询信息，同时能够对程序进行操作的对象有管理员若干、快递员若干、顾客若干，同时顾客又分为公司客户和个人用户，而需要被记录信息、修改信息和查询信息的对象则有包裹若干、顾客若干、快递员若干、账单若干，在后端，对对象的操作由类的成员函数来实现，事物的信息则封装在实例化的对象中 ，在前端，用户发出指令则由前端的交互界面函数实现；</p>
</td></tr>
<tr><td>设计思路</td><td>
<p>从结构上来说，我们这个程序需要包括时间类、包裹类、顾客类、快递员类、账单类，而信息则由类的数据成员记录，每有一个新的订单就实例化一个新的包裹，建立对象数组来存放这些不同的包裹，然后通过建立类似哈希表的方式将包裹和用户绑定。不同的使用程序的用户则应该能够调用不同的函数来进行不同的操作，就此也就确定了程序的主体结构；</p>
<p>在此基础之上，我们在通过在类中编写函数来对类中的数据进行操作，而在主体程序中编写函数来实现程序和用户之间的交互，也就是完成程序的细节功能；</p>


</td></tr>
</table>

讨论过程	程序开始讨论到设计完成共历时 27 天，最初是确定分工，李昂同学负责时间类、货物类，皮家恺同学负责顾客和账单类，王旭同学负责快递员类并编写实现人机交互的函数； 本次大作业我们多次在图书馆研读间讨论，我们认为做好本次大作业不仅需要良好的 C++编程能力，更重要的是当好 product manager，理解用户需求，认真设计程序功能。我们每过几天就聚在一起交流代码和功能设计。 在设计类的过程中：第一次讨论中，本打算再设计一个付款类用于单独记录每一次的付款信息，但考虑到每一次付款的信息是严格与每一次包裹信息联系的，所以将每一次付款的信息直接记录到了包裹类中，同时增加一个时间类，作为记录、修改及打印时间信息的工具；第二次讨论中，确定了各个类分别需要实现的功能，同时考虑到为了便于可以在修改信息（即类的数据成员）的过程中，否决了使用构造函数来记录类的数据成员的方式，采用只是写一个空的构造函数，而使用成员函数给数据成员赋值和修改数据成员的方式；第三次讨论中，确定了各个类内的数据成员的限定方式，如包裹类中表示付款类型的数据成员 paytype，值-1 表示货到付款，而值 1 表示按月付款；第六次讨论中，考虑到需要建立多个成员，故定义了对象数组，并通过对象数组的下标确定确定了顾客对象、包裹对象和账单对象的对应关系；第七次讨论到第十一次讨论中，确定了对象在访问对应的其他类的对象的数据成员时，将对应对象作为参数传递到类的函数中，以此来获得正确的对应信息； 在设计交互函数的过程中：第五次讨论中，确定了交互函数所需要实现的所有内容，包括但不限于登录界面、登记信息界面等；第六次讨论中，考虑到账单类内修改数据成员的函数需要调用到每次包裹输入的数据成员，故确定在每次修改或输入包裹对象的数据成员时，紧接着修改一次账单类的数据成员；第十次讨论中，考虑到程序中的类创建的对象间缺乏联系和绑定，因此打算使用数组对象的下标来进行一一对应；第十一次讨论中，确定了以便于返回上级界面的循环嵌套的结构，以及通过 switch 函数对每个界面输入值的判定来决定下一步操作的方式作为整个交互函数体系的基本框架，使人机交互逻辑变得更加人性化；第十七次讨论中，确定添加根据时间是上下午来筛选包裹函数；第十八次讨论中，确定添加注册登录账户的函数； 细节优化的过程中：第四次讨论中，确定所有的字符串数据均申明为 string 型，同时大部分 string 型变量均以 getline 函数来赋值，便于接收含有空格符的字符串；第六次讨论中，确定了类对象的输出内容和输出格式的统一性，同时确定了函数名、变量名等内容的一致性；第十一次讨论过程中，统一了编程风格及和用户交互中的输出格式，同时确定了以 2 位数的编号作为包裹对象的下标，十位数表示用户，个位数表示该用户对应的包裹，共可以建立 10 个用户对象，10 个账单对象，100 个包裹对象，即类似哈希表的方法来完成，以取代原本应该使用数据库完成的功能；第十二次至第十六次讨论主要为 debug；第十九次讨论中，为使程序能够在 Windows、linux 和 macos 环境下都可以使用，删除了 conio 头文件以及对应的 kbhit 函数，并以 cin 来代替；第二十次讨论中，在每个 getline 函数后增添一个判断语句，以排除 cin 缓冲区对 getline 函数的影响；第二十一至第二十三次讨论主要内容均为优化代码以及 debug；

部分代码亮点

字符串的输入：

历史版本 1：	char a[n]; cin >> a[];	
历史版本 2：	string a; cin >>a	优化：方便输入
历史版本 3：	string a; getline(cin,a);	优化：可以接收空格符

历史版本 4.1~现版本 4.3 均是为防止 cin 缓冲区对 getline 接收造成影响

历史版本 4.1：	string a;
	fflush(stdin);
	getline(cin,a);
		没有达到预期效果

历史版本 4.2：	string a;
	cin.clear();
	cin.ignore(numeric_limits<streamsize>::max(),’\n’);
	getline(cin.a);
		能够清空 cin 缓冲区，但是如果 cin 缓冲区本就为空，该语句需要输入一个值

现版本 4.3：	string a;
	getline(cin, a);
	if(a == “”)
	{
	getline(cin, a);
	}
		当 cin 缓冲区不为空时，若 cin 函数运转正常，则 cin 缓冲区最后的符号必为换行符，此时 cin 缓冲区的值将赋给 a，getline(cin, a)接收到的即为空值，因此对 a 值进行判定，若为空值则再次调用 getline(cin, a)对 a 进行赋值则成功赋值

Time 类的构建：

为了便于记录、修改及打印时间信息，需要自己写一个 Time 类，类中包括月份，日期，小时，分钟，同时通过调用成员函数 set 进行数据成员的设置，通过调用 display 函数进行输出信息。另外，由于 C++中类无法在运行时调用自己的名字（编译后所有对象名都变成了地址），因此为了方便输出必须设置一个 string 变量存自己的名字，这样在调用 display 函数时就可完整的输出信息。

Time 类成员有：

```
int month;
int date;
int hour;
int minute;
string name;
int flag;
void set();
void display();
```

嵌套循环结构实现返回上级：

```
int c = 0, cc = 0, ccc = 0;
while (c == 0)
{
    int a = -1;
    cin >> a;
    if (a == 0) { c = 1; }
    else
    {
        while (cc == 0)
        {
            int aa = -1;
            cin >> aa;
            if (aa == 0) { ccc = 1; }
            else
            {
                while (ccc == 0)
                {
                    int aaa = -1;
                    cin >> aaa;
                    if (aaa == 0) { ccc = 1; }
                    else
                    {


                    }
                }
                ccc = 0;
            }
        }
        cc = 0;
    }
}
```

通过对输入值 a,aa,aaa 的判定，决定是否改变 c,cc,ccc 的值，从而决定是否跳出循环，通过控制 c,cc,ccc 的值，可以决定跳出到哪一层循环，同时在每次跳出循环后，让 c,cc,ccc 的值返回初始值，以便于下一次循环时能够继续用于判定

附录 1

本次大作业代码 共 1335 行

```
#include <string>
#include <stdio.h>
#include <ctime>
#include <iostream>
using namespace std;
const int N = 10;
class Package;
class Payment;
class Customer;
class Deliveryman;
class list;
int package_count = 0;
int c = 0;
int cc = 0;
int ccc = 0;
int cccc = 0;
string skip;

class Time
{
public:
    int month;
    int date;
    int hour;
    int minute;
    string name;
    int flag;

    Time(string n)
    {
        name = n;
        month = 0;
        date = 0;
        hour = 0;
        minute = 0;
        flag = 0;
    }
    void set()
    {
        int a, b, c, d;
        cout << "Please enter the " << name << " (month date hour minnute)" << endl;

        cin >> a >> b >> c >> d; // 此处按照顺序输入 月 日 小时 分钟
        month = a;
        date = b;
        hour = c;
        minute = d;
        cout << endl;
    }
    void display()
    {
        if (month != 0)
            cout << "The " << name << " is " << month << "/" << date << " " << hour << ":" << minute <<
endl
            << endl;
        else
        {
            cout << "No " << name << " information available!" << endl
                << endl;
        }
    }
};

class Package
{
public:
    friend class Payment;
    friend class List;
    friend int modify(Package &a, int mo);
    Time fetchtime{ "fetchtime" }; //用户输入（上门取件）

    string getplace;               //
    Time gettime{ "gettime" }; //实际取到寄件（系统记录 0->1)

    Time deliverytime{ "deliverytime" }; //开始寄出（仓库端）

    Time taketime{ "taketime" }; //实际收货人得到寄件 1->0

    Time paytime{ "paytime" };
    int paytype;
    string takeplace;

    // takedistance//函数

    double weight; //用户输入

    double ware;   //金额，由函数得出
    int state;          //(-2，-1，0，1，2)表示准备提货、提货、到达仓库、发货、已发货
    string curplace; //当前所在位置
    int speed;               //(-1,0,1)(三小时/当天/过夜）用户确定
    int distance;   //距离
    int cancel;
    int flag; //标记包裹是否存在；

    Package()
    {
```

```
    cancel = 0;
    flag = 0;
}

// printsign（）
// printpayment（）
// taketime（）
// cancel（）//用户取消 包删除
// enum livesize{}
// enum liveweight{}

void set_curplace()
{
    fflush(stdin);
    cout << "Please enter the current place." << endl;
    getline(cin, curplace);
    if (curplace == "")
    {
        getline(cin, curplace);
    }
    cout << "Successfully set current place!" << endl
        << endl;
}
void display_curplace()
{
    cout << "The package is currently at " << curplace << "." << endl
        << endl;
}

void set_paytype()
{
    cout << "Please enter the paytype.(-1: individual shipments; 1:monthly invoices)" << endl;
    cin >> paytype;
    cout << "Successfully set paytype!" << endl
        << endl;
}

void display_paytype()
{
    switch (paytype)
    {
    case -1:
        cout << "The paytype is individual shipments. " << endl
            << endl;
        break;

    case 1:
        cout << "The paytype is monthly invoices. " << endl
            << endl;
            break;

        default:
            cout << "No paytype information available." << endl
                << endl;
            break;
        }
}

void set_ware()
{
    double w;
    distance = (getplace.length() + takeplace.length()) * 10;

    cout << "Please enter the weight." << endl;
    cin >> w;
    weight = w;
    ware = weight * 10 * distance;
    cout << endl;
};

void display_ware()
{
    cout << "The price is " << ware << "." << endl
        << endl;
}

void set_getplace()
{
    fflush(stdin);
    cout << "Please enter the getplace." << endl;
    getline(cin, getplace);
    if (getplace == "")
    {
        getline(cin, getplace);
    }
    cout << "Successfully set getplace!" << endl
        << endl;
}

void display_getplace()
{
    cout << "The getplace is " << getplace << "." << endl
        << endl;
}
void set_takeplace()
{
    fflush(stdin);
    cout << "Please enter the takeplace." << endl;
```

```
        getline(cin, takeplace);
        if (takeplace == "")
        {
            getline(cin, takeplace);
        }
        cout << "Successfully set takeplace!" << endl
            << endl;
    }

    void display_takeplace()
    {
        cout << "The destination is " << takeplace << "." << endl
            << endl;
    }

    void set_state()
    {
        int s;

        cout << "Please enter the state (-2:ready to pick up; -1:picked up; 0:arrived at warehouse; 1:out for delivery 2:delivered)"
            << endl;
        cin >> s;
        state = s;
        cout << "Successfully set status!" << endl
            << endl;
    }

    void display_state()
    {
        switch (state)
        {
        case -2:
            cout << "The package is ready to pick up." << endl
                << endl;
            break;
        case -1:
            cout << "The package is picked up." << endl
                << endl;
            break;
        case 0:
            cout << "The package arrived at warehouse." << endl
                << endl;
            break;
        case 1:
            cout << "The package is out for delivery." << endl
                << endl;
            break;
        case 2:
            cout << "The package is delivered." << endl
                << endl;
            break;

        default:
            cout << "No status information available." << endl
                << endl;
            break;
        }
    }

    void set_speed()
    {
        int temp;
        cout << "Please enter the service speed (-1:in 3 hours; 0:today; 1:overnight)" << endl;
        cin >> temp;
        speed = temp;
        cout << "Successfully set speed!" << endl
            << endl;
    }

    void display_speed()
    {
        switch (speed)
        {
        case -1:
            cout << "The service speed request is: in 3 hours." << endl
                << endl;

            break;
        case 0:
            cout << "The service speed request is: today." << endl
                << endl;
            break;
        case 1:
            cout << "The service speed request is: overnight." << endl
                << endl;
            break;
        default:
            cout << "No service speed information available." << endl
                << endl;
            break;
        }
    }

    void set()
    {
        fetchtime.set();
        gettime.set();

        set_getplace();
        set_curplace();
        curplace = getplace;
        set_takeplace();
        set_speed();
        set_paytype();

        set_ware();

        set_state();

        flag = 1;
    }

    void print_information()
    {
        system("clear");
        fetchtime.display();
        gettime.display();
        display_getplace();
        display_curplace();
        display_takeplace();
        display_speed();
        display_paytype();
        display_ware();
        display_state();
        paytime.display();
        taketime.display();
    }

    void display_payment()
    {
        display_paytype();
        paytime.display();
        display_ware();
    }

    void cancel_package()
    {
        cancel = 1;
        flag = 0;
        cout << "The package has been cancelled" << endl;
    }
};

class Customer
{
public:
    string company;
    string linkman;
    string telephone;
    string adress;
    string name;
    int flag;

    Customer()
    {
        flag = 0;
    }

    void set_name()
    {
        string nm;

        cout << "Please enter your name" << endl
            << endl;
        getline(cin, nm); //此处输入用户名称
        if (nm == "")
        {
            getline(cin, nm);
        }

        name = nm;

        cout << "Successfully set name!" << endl
            << endl;
    }

    void set_company()
    {
        string cpy;

        cout << "Please enter your company name ( if you are individual user, please enter 1 )" << endl
            << endl;
        getline(cin, cpy); //此处输入公司名称
        if (cpy == "")
        {
            getline(cin, cpy);
        }

        company = cpy;

        cout << "Successfully set company!" << endl
            << endl;
    }

    void set_linkman()
    {
        string lkm;

        cout << "Please enter the linkman's name" << endl
            << endl;
        getline(cin, lkm); //此处输入联系人名称
        if (lkm == "")
        {
            getline(cin, lkm);
        }

        linkman = lkm;

        cout << "Successfully set linkman!" << endl
            << endl;
    }

    void set_telephone()
    {
        string tlp;

        cout << "Please enter your telephone number" << endl
            << endl;
        getline(cin, tlp); //此处输入联系人电话
        if (tlp == "")
        {
            getline(cin, tlp);
        }

        telephone = tlp;

        cout << "Successfully set telephone!" << endl
            << endl;
    }

    void set_adress()
    {
        string ads;

        cout << "Please enter the your shipping address" << endl
            << endl;
        getline(cin, ads); //此处输入地址
        if (ads == "")
        {
            getline(cin, ads);
        }

        adress = ads;

        cout << "Successfully set adress!" << endl
            << endl;
    }

    void set()
    {
        set_name();
        set_company();
        set_linkman();
        set_telephone();
        set_adress();

        cout << "Successfully set all customer messages!" << endl
            << endl;
        flag = 1;
    }

    void display_company()
    {
        if (company == "1")
        {
            cout << "The " << name << " is a individual customer" << endl
                << endl;
        }
        else
        {
            cout << "company:\t" << company << endl
                << endl;
        }
    }

    void display_linkman()
    {
        cout << "linkman:\t" << linkman << endl
            << endl;
    }

    void display_telephone()
    {
        cout << "telephone:\t" << telephone << endl
            << endl;
    }

    void display_adress()
    {
        cout << "adress:\t" << adress << endl
            << endl;
    }

    void display()
    {
        cout << "The " << name << "'s all messages:"
            << "\ncompany:\t" << company
            << "\nlinkman:\t" << linkman
            << "\ntelephone:\t" << telephone
            << "\nadress:\t" << adress << endl
            << endl;
    }
};

class Deliveryman
{
public:
    string deliveryman_phone;
    string deliveryman_name;
    string dname;
    string phone;
    int flag;
    Deliveryman()
    {
        flag = 0;
    }
    void set()
    {

        cout << "Please enter the diliveryman_name"
            << endl;
        getline(cin, dname); //此处输入联系人名称
        if (dname == "")
        {
            getline(cin, dname);
        }
        deliveryman_name = dname;

        cout << "Please enter the " << dname << "'s telephone number" << endl
            << endl;
        getline(cin, phone); //此处输入联系人电话
        if (phone == "")
        {
            getline(cin, phone);
        }
        deliveryman_phone = phone;

        cout << "Successfully set diliveryman!" << endl
            << endl;
        flag = 1;
    }
    void display()
    {
        cout << "deliveryman:\t" << dname
            << "\nphone:\t" << phone
            << endl;
    }
};

class List
{
public:
    int paytype;
    double weightsum = 0;
    int numbersum = 0;
    double waresum = 0;
    double paidsum = 0;
    double paid = 0;
    double debtsum = 0;
    string name;

    void sum(Package a)
    {
        paytype = a.paytype;
        weightsum = weightsum + a.weight;
        numbersum = numbersum + 1;
        if (paytype == -1)
        {
            paid = a.ware;
        }
        else if (paytype == 1)
        {
            paid = 0;
        }
        waresum = waresum + a.ware;
        paidsum = paidsum + paid;
        debtsum = waresum - paidsum;
    }

    void display_paytype()
    {
        cout << "paytype:\t" << paytype << endl
            << endl;
    }

    void display_weightsum()
    {
        cout << "weightsum:\t" << weightsum << endl
            << endl;
    }

```

```
	void display_numbersum()
	{
		cout << "numbersum:\t" << numbersum << endl
			<< endl;
	}

	void display_waresum()
	{
		cout << "waresum:\t" << waresum << endl
			<< endl;
	}

	void display_debtsum()
	{
		cout << "debtsum:\t" << debtsum << endl
			<< endl;
	}

	void display()
	{
		cout << "The " << name << "'s list:"
			<< "\npaytype:\t" << paytype
			<< "\nweightsum:\t" << weightsum
			<< "\nnumbersum:\t" << numbersum
			<< "\nwaresum:\t" << waresum
			<< "\ndebtsum:\t" << debtsum << endl
			<< endl;
	}
};

//Package package[N];
//Customer customer[N];
//Deliveryman deliveryman[N];
Customer customer[10];
Package package[100];
Deliveryman deliveryman[10];
List list[10];

int login();
void register_customer();
void register_deliveryman();
void modify_package();
void package_manage();
void modify_customer(int cnum);
void customer_manage();
void dilivery_manage();
void account_administrator();
void account_deliveryman();
void account_company();
void account_person();
//void search_package(Time t);

void search_package(Time t)
{
    int i;
    int morning = 0;
    if (t.hour >= 12)
        morning = 1;            //判断输入的时间是上午还是下午
    for (i = 0; i < 100; i++) //此处用 taketime 不妥  taketime 是用户实际收到包裹的时间，应改为约定收取包裹的时间
    {
        if (package[i].fetchtime.month == t.month && package[i].fetchtime.date == t.date) //此处不用判断 flag
        {
            if (morning == 0 && package[i].fetchtime.hour < 12)
                package[i].print_information();
            if (morning == 1 && package[i].fetchtime.hour >= 12)
                package[i].print_information();
        }
        // switch (t.flag)
        // {
        // case 1:
        //    if (package[i].taketime.hour <= 12)
        //    {
        //        package[i].print_information();
        //    }
        //    break;
        // case 2:
        //    if (package[i].taketime.hour <= 12)
        //    {
        //        package[i].print_information();
        //    }
        //    break;
        // case 0:
        //    package[i].print_information();
        // }
    }
}

void register_customer()
{
    int cnum = 0;
    while (customer[cnum].flag == 1)
    {
        ++cnum;
    }
    customer[cnum].set();
    cout << "Successful!Your number is :" << cnum << endl;
```

```
}

void register_deliveryman()
{
    int dnum = 0;
    while (deliveryman[dnum].flag == 1)
    {
        ++dnum;
    }
    deliveryman[dnum].set();
    cout << "Successful!Your number is :" << dnum << endl;
}

int login()
{
    int account, state = 0;
    cout << "login_ID" << endl; //administrator account 1-digit dilivery
    // account 3-digit company account 5-digit personal account 7-digit
    cin >> account;
    if (account > 0 && account <= 9)
    {
        state = 1;
    }
    else if (account >= 100 && account <= 999)
    {
        state = 3;
    }
    else if (account >= 10000 && account <= 99999)
    {
        state = 5;
    }
    else if (account >= 1000000 && account <= 9999999)
    {
        state = 7;
    }
    else if (account == 0)
    {
        state = 0;
    }
    return state;
}

void modify_package()
{
    int pnum;
    int mo;
    while (cccc == 0)
    {
        pnum = 0;
        cout << "Enter the package number" << endl;
        cin >> pnum;
        cout << "1_fetchtime\n";
        cout << "2_gettime\n";
        cout << "3_getplace\n";
        cout << "4_curplace\n";
        cout << "5_takeplace\n";
        cout << "6_speed\n";
        cout << "7_state\n";
        cout << "9_Back To Previous" << endl;
        cout << "0_Back To Mainmenu" << endl;
        mo = 0;
        cin >> mo;
        switch (mo)
        {
        case 1:
            package[pnum].fetchtime.set();
            cout << "Successfully modified!";
            break;
        case 2:
            package[pnum].gettime.set();
            cout << "Successfully modified!";
            break;
        case 3:
            package[pnum].set_getplace();
            cout << "Successfully modified!";
            break;
        case 4:
            package[pnum].set_curplace();
            cout << "Successfully modified!";
            break;
        case 5:
            package[pnum].set_takeplace();
            cout << "Successfully modified!";
            break;
        case 6:
            package[pnum].set_speed();
            cout << "Successfully modified!";
            break;
        case 7:
            package[pnum].set_state();
            cout << "Successfully modified!";
            break;
        case 9:
            cccc = 1;
            ccc = 0;
            break;
        case 0:
            cccc = 1;
            ccc = 1;
            cc = 0;
            break;
        default:
            cout << "order error, enter any value to return" << endl;
            cin >> skip;
            break;
        }
    }
    cccc = 0;
}

void modify_customer(int cnum)
{
    while (cccc == 0)
    {
        cout << "1_company" << endl;
        cout << "2_linkman" << endl;
        cout << "3_telephone" << endl;
        cout << "4_adress" << endl;
        cout << "5_name" << endl;
        cout << "9_Back To Previous" << endl;
        cout << "0_Back To Mainmenu" << endl;
        int mo = 0;
        cin >> mo;
        switch (mo)
        {
        case 1:
            customer[cnum].set_company();
            break;
        case 2:
            customer[cnum].set_linkman();
            break;
        case 3:
            customer[cnum].set_telephone();
            break;
        case 4:
            customer[cnum].set_adress();
            break;
        case 5:
            customer[cnum].set_name();
            break;
        case 9:
            cccc = 1;
            ccc = 0;
            break;
        case 0:
            cccc = 1;
            ccc = 1;
            cc = 0;
            break;
        default:
            cout << "order error, enter any value to return" << endl;
            cin >> skip;
            break;
        }
    }
    cccc = 0;
}

void package_manage()

{ //包裹管理系统
    int p;
    while (ccc == 0)
    {
        cout << "1_register package information" << endl;
        cout << "2_query package information" << endl;
        cout << "3_modify package information" << endl;
        cout << "4_modify package status" << endl;
        cout << "9_Back To Previous" << endl;
        cout << "0_Back To Mainmenu" << endl;
        p = 0;
        cin >> p;
        switch (p)
        {
        case 1:
            //登记包裹信息
        {
            cout << "Enter the customer number" << endl;
            int cnum, pnum;
            cin >> cnum;
            pnum = cnum * 10;
            int i = 0;
            while (package[pnum + i].flag == 1)
            {
                ++i;
            }
            pnum = pnum + i; //定位到包裹的位置
            package[pnum].set();
            list[cnum].sum(package[pnum]);
        }
        break;
        case 2: //查询包裹信息
        {
            cout << "Enter the package number" << endl;
            int pnum; //通过订单号查询
            cin >> pnum;
            package[pnum].print_information();
        }
        break;
        case 3: //修改包裹信息
            modify_package();
            break;
        case 4:
            cout << "Enter the package number" << endl;
            int dcnum2;
            cin >> dcnum2;
            package[dcnum2].set_state();
            break;
        case 9:
        {
            ccc = 1;
            cc = 0;
        }
        break;
        case 0:
        {
            ccc = 1;
            cc = 0;
        }
        break;
        default:
        {
            cout << "order error, enter any value to return" << endl;
            cin >> skip;
        }
        break;
        }
    }
    ccc = 0;
}

void customer_manage()

{ //用户管理系统
    while (ccc == 0)
    {
        cout << "1_register customer information" << endl;
        cout << "2_query customer information" << endl;
        cout << "3_modify customer information" << endl;
        cout << "4_query package information" << endl;
        cout << "9_Back To Previous" << endl;
        cout << "0_Back To Mainmenu" << endl;
        int con;
        cin >> con;
        switch (con)
        {
        case 1:
            register_customer();
            break;
        case 2:
            cout << "Enter the customer number" << endl;
            int cnum2;
            cin >> cnum2;
            customer[cnum2].display();
            break;
        case 3:
            cout << "Enter the customer number" << endl;
            int cnum3;
            cin >> cnum3;
            modify_customer(cnum3);
            break;
        case 4:
            cout << "Enter the customer number" << endl;
            int cnum4;
            cin >> cnum4;
            cnum4 = 10 * cnum4;
            while (package[cnum4].flag == 1)
            {
                package[cnum4].print_information();
                ++cnum4;
            }
        case 9:
            ccc = 1;
            cc = 0;
            break;
        case 0:
            ccc = 1;
            cc = 0;
            break;
        default:
            cout << "order error, enter any value to return" << endl;
            cin >> skip;
            break;
        }
    }
    ccc = 0;
}

void dilivery_manage()
{ //快递员管理系统
    while (ccc == 0)
    {
        cout << "1_register dilivery information" << endl;
        cout << "2_query dilivery information" << endl;
        cout << "9_exit" << endl;
        int d;
        cin >> d;

        switch (d)
        {
        case 1:
            register_deliveryman();
            break;
        case 2:
            cout << "Enter the deliveryman number" << endl;
            int dnum2;
            cin >> dnum2;
            deliveryman[dnum2].display();
            break;
        case 9:
            ccc = 1;
            cc = 0;
            break;
        case 0:
            ccc = 1;
            cc = 0;
            break;
        default:
            cout << "order error, enter any value to return" << endl;
            cin >> skip;
            break;
        }
    }
    ccc = 0;
}

void account_administrator()
{
    int order;
    while (cc == 0)
    {
        cout << "1_Package" << endl;
        cout << "2_Customer" << endl;
        cout << "3_Diliveryman" << endl;
        cout << "0_Relanding" << endl;
        order = 0;
        cin >> order;
        if (order == 1)
        {
            package_manage();
        }
        else if (order == 2)
        {
            customer_manage();
        }
        else if (order == 3)
        {
            dilivery_manage();
        }
        else if (order == 0)
        {
            cc = 1;
            c = 0;
        }
        else
        {
            cout << "order error, enter any value to return" << endl;
            cin >> skip;
        }
    }
    cc = 0;
}

void account_deliveryman() //快递员系统
{
    int order;
    while (cc == 0)
    {
        cout << "1_query package information" << endl;
        cout << "2_modify package status" << endl;
        cout << "3_query package information in this period" << endl;
        cout << "0_Relanding" << endl;
        order = 0;
        Time period("period");
        cin >> order;

        switch (order)
        {
        case 1:
            cout << "Enter the package number" << endl;
            int dcnum1;
            cin >> dcnum1;
            package[dcnum1].print_information();
            break;
        case 2:
            cout << "Enter the package number" << endl;
            int dcnum2;
            cin >> dcnum2;
            package[dcnum2].set_state();
            break;
        case 3:
            cout << "Enter the period" << endl;

            period.set();
            search_package(period);

            break;
        case 0:
            cc = 1;
            c = 0;
            break;
        default:
            cout << "order error, enter any value to return" << endl;
            cin >> skip;
            break;
        }
    }
    cc = 0;
}

void account_company()
{
    int order;
    while (cc == 0)
    {
        cout << "Enter company number" << endl;
        int cpnum;
        cin >> cpnum;
        if (customer[cpnum].flag == 0)
        {
            register_customer();
        }
        cout << "1_modify account" << endl;
        cout << "2_register package information" << endl;
        cout << "3_modify package information" << endl;
        cout << "4_query package information" << endl;
        cout << "5_query order records" << endl;
        cout << "6_query monthly statement" << endl;
        cout << "7_months to be paid bill" << endl;
        cout << "8_cancel the package" << endl;
        cout << "0_relanding" << endl;
        order = 0;
        cin >> order;
        switch (order)
        {
        case 1:
            modify_customer(cpnum);
            break;
        case 2:
            int pnum2;
            pnum2 = cpnum * 10;
            while (package[pnum2].flag == 1)
            {
                ++pnum2;
            }
            package[pnum2].set();
            list[cpnum].sum(package[pnum2]);
            break;
        case 3:
            modify_package();
            break;
        case 4:
            cout << "Enter the package number" << endl;
            int pnum4;
            cin >> pnum4;
            package[pnum4].print_information();
            break;
        case 5:
            cpnum = 10 * cpnum;
            int i;
            for (i = 0; i < 10; i++)
            {
                if (package[cpnum].flag == 1 || package[cpnum].cancel == 0) //                    {
                                                                //输出全部订单
                    cout << "pack" << cpnum << ":";
                    package[cpnum].print_information();
                    ++cpnum;
                }
            }
            break;
        case 6:
            list[cpnum].display();
            break;
        case 7:
            list[cpnum].display_debtsum();
            break;
        case 8:
            cout << "Enter the package number" << endl;
            while (package[cpnum].flag == 1)
            {
                cout << "pack" << cpnum << ":";
                package[cpnum].print_information();
                cout << endl;
                ++cpnum;
            }
            int pnum8;
            cin >> pnum8;
            if (package[pnum8].state == -2)
            {
                package[pnum8].cancel_package();
            }
            else
                cout << "Sorry,it cannot be cancelled" << endl;
            break;
        case 0:
            cc = 1;
            c = 0;
            break;
        default:
            cout << "order error, enter any value to return" << endl;
            cin >> skip;
            break;
        }
    }
    cc = 0;
}

void account_person()
{
    cout << "Enter person number" << endl;
    int psnum;
    cin >> psnum;
    if (customer[psnum].flag == 0)
    {
        register_customer();
    }
    int order;
    while (cc == 0)
    {
        cout << "1_modify account" << endl;
        cout << "2_register package information" << endl;
        cout << "3_modify package information" << endl;
        cout << "4_query package information" << endl;
        cout << "5_querying records" << endl;
        cout << "6_modify package status" << endl;
        cout << "7_months to be paid bill" << endl;
        cout << "8_cancel the package" << endl;
        cout << "0_Relanding" << endl;
        order = 0;
        cin >> order;
        switch (order)
        {
        case 1:
            modify_customer(psnum);
            break;
        case 2:
            while (package[psnum].flag == 1)
            {
                ++psnum;
            }
            package[psnum].set();
            list[psnum].sum(package[psnum]);
            break;
        case 3:
            modify_package();
            break;
        case 4:
            cout << "Enter the package number" << endl;
            int pnum4;
            cin >> pnum4;
            package[pnum4].print_information();
            break;
        case 5:
            psnum = 10 * psnum;
            int i;
            for (i = 0; i < 10; i++)
            {
                if (package[psnum].flag == 1 || package[psnum].cancel == 1)
                { //输出全部订单
                    cout << "pack" << psnum << ":";
                    package[psnum].print_information();
                    ++psnum;
                }
            }
            break;
        case 6:
            for (i = 0; i < 10; i++)
            {
                if (package[psnum].flag == 1)
                { //输出全部订单
                    cout << "pack" << psnum << ":";
                    package[psnum].print_information();
                    ++psnum;
                }
            }
            cout << "Enter the package number" << endl;
            int psnum6;
            cin >> psnum6;
            package[psnum6].set_state();
            break;
        case 7:
            list[psnum].display_debtsum();
            break;
        case 8:
            cout << "Enter the package number" << endl;
            while (package[psnum].flag == 1)
            {
                cout << "pack" << psnum << ":";
```

```
                package[psnum].print_information();
                cout << endl;
                ++psnum;
            }
            int pnum;
            cin >> pnum;
            if (package[pnum].state == -2)
            {
                package[pnum].cancel_package();
            }
            else
                cout << "Sorry,it cannot be cancelled" << endl;
            break;
        case 0:
            cc = 1;
            c = 0;
            break;
        default:
            cout << "order error, enter any value to return" << endl;
            cin >> skip;
            break;
        }
    }
    cc = 0;
}

int main()
{

    while (c == 0)
    {
        int login_account;
        cout << "enter 0 to exit" << endl;
        login_account = login();

        switch (login_account)
        {
        case 1:
            account_administrator();
            break;
        case 3:
            account_deliveryman();
            break;
        case 5:
            account_company();
            break;
        case 7:
            account_person();
            break;
        case 0:
            c = 1;
            break;
        default:
            cout << "login_ID is error" << endl;
            break;
        }

        cout << "Successfully exit" << endl << endl;
        login_account = 0;
    }
    c = 0;
    return 0;
}
```