面向对象的程序设计

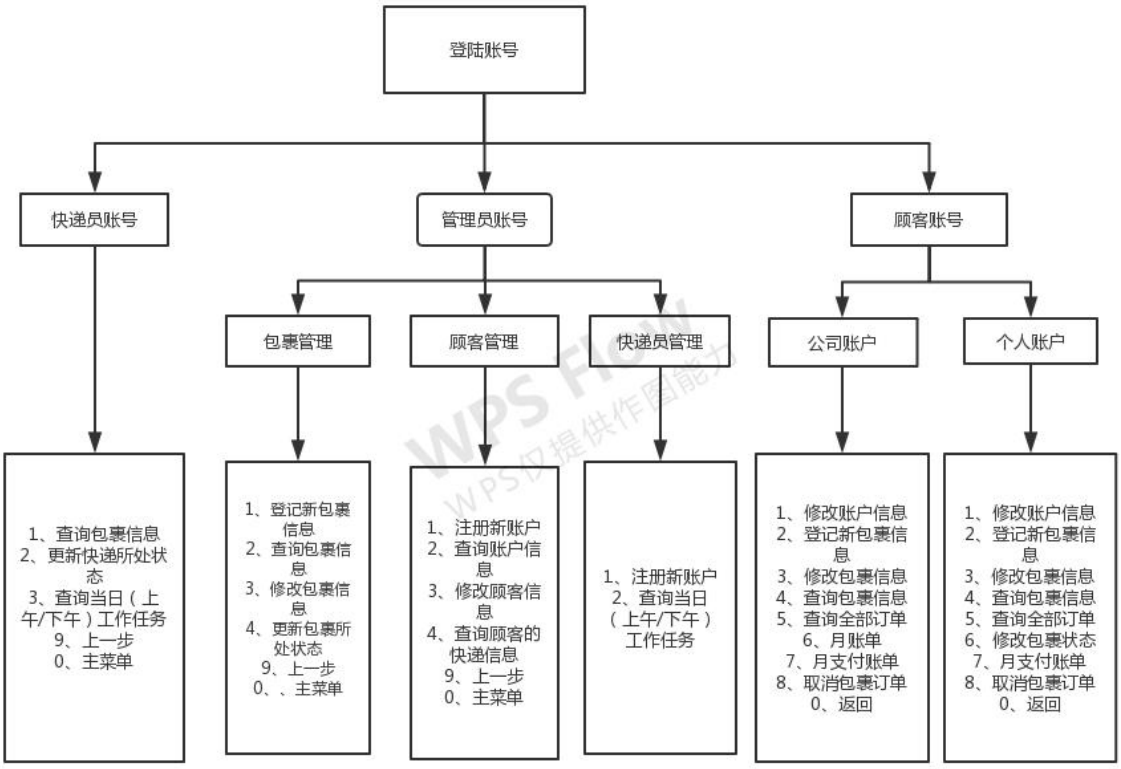
——大作业报告

报告人：

物联网 81 王旭 2186412635

物试 51 李昂 2150201074

物联网 81 皮家恺 2184214240

题目理解	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；
设计思路	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； 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； 

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;　　没有达到预期效果

fflush(stdin);

getline(cin,a);

历史版本 4.2：　string a;　　能够清空 cin 缓冲区，但是如果 cin 缓

cin.clear();　　冲区本就为空，该语句需要输入一个值

cin.ignore(numeric_limits<streamsize>::max(),'\n');

getline(cin.a);

现版本 4.3：　string a;　　当 cin 缓冲区不为空时，若 cin 函数运转

getline(cin, a);　　正常，则 cin 缓冲区最后的符号必为换

if(a == "")　　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

getline(cin, a);　　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;
```

```
50              cout << endl;
51          }
52          void display()
53          {
54              if (month != 0)
55                  cout << "The " << name << " is " << month << "/" << date << " " << hour << ":" << minute <<
56  endl
57                  << endl;
58              else
59              {
60                  cout << "No " << name << " information available!" << endl
61                       << endl;
62              }
63          }
64  };
65
66  class Package
67  {
68  public:
69          friend class Payment;
70          friend class List;
71          friend int modify(Package &a, int mo);
72          Time fetchtime{ "fetchtime" }; //用户输入（上门取件）
73
74          string getplace;              //
75          Time gettime{ "gettime" }; //实际取到寄件（系统记录 0->1)
76
77          Time deliverytime{ "deliverytime" }; //开始寄出（仓库端）
78
79          Time taketime{ "taketime" }; //实际收货人得到寄件 1->0
80
81          Time paytime{ "paytime" };
82          int paytype;
83          string takeplace;
84
85          // takedistance//函数
86
87          double weight; //用户输入
88
89          double ware;   //金额，由函数得出
90          int state;          //(-2，-1，0，1，2)表示准备提货、提货、到达仓库、发货、已发货
91          string curplace; //当前所在位置
92          int speed;              //(-1,0,1)(三小时/当天/过夜）用户确定
93          int distance;   //距离
94          int cancel;
95          int flag; //标记包裹是否存在;
96
97          Package()
98          {
```

```
99          cancel = 0;
100         flag = 0;
101     }
102
103     // printsign（）
104     // printpayment（）
105     // taketime（）
106     // cancel（）//用户取消 包删除
107     // enum livesize{}
108     // enum liveweight{}
109
110     void set_curplace()
111     {
112         fflush(stdin);
113         cout << "Please enter the current place." << endl;
114         getline(cin, curplace);
115         if (curplace == "")
116         {
117             getline(cin, curplace);
118         }
119         cout << "Successfully set current place!" << endl
120             << endl;
121     }
122     void display_curplace()
123     {
124         cout << "The package is currently at " << curplace << "." << endl
125             << endl;
126     }
127
128     void set_paytype()
129     {
130         cout << "Please enter the paytype.(-1: individual shipments; 1:monthly invoices)" << endl;
131         cin >> paytype;
132         cout << "Successfully set paytype!" << endl
133             << endl;
134     }
135
136     void display_paytype()
137     {
138         switch (paytype)
139         {
140         case -1:
141             cout << "The paytype is individual shipments. " << endl
142                 << endl;
143             break;
144
145         case 1:
146             cout << "The paytype is monthly invoices. " << endl
147                 << endl;
```

```
148                 break;
149
150         default:
151                 cout << "No paytype information available." << endl
152                         << endl;
153                 break;
154         }
155     }
156
157     void set_ware()
158     {
159         double w;
160         distance = (getplace.length() + takeplace.length()) * 10;
161
162         cout << "Please enter the weight." << endl;
163         cin >> w;
164         weight = w;
165         ware = weight * 10 * distance;
166         cout << endl;
167     };
168
169     void display_ware()
170     {
171         cout << "The price is " << ware << "." << endl
172                 << endl;
173     }
174
175     void set_getplace()
176     {
177         fflush(stdin);
178         cout << "Please enter the getplace." << endl;
179         getline(cin, getplace);
180         if (getplace == "")
181         {
182             getline(cin, getplace);
183         }
184         cout << "Successfully set getplace!" << endl
185                 << endl;
186     }
187
188     void display_getplace()
189     {
190         cout << "The getplace is " << getplace << "." << endl
191                 << endl;
192     }
193     void set_takeplace()
194     {
195         fflush(stdin);
196         cout << "Please enter the takeplace." << endl;
```

```
197             getline(cin, takeplace);
198             if (takeplace == "")
199             {
200                 getline(cin, takeplace);
201             }
202             cout << "Successfully set takeplace!" << endl
203                 << endl;
204         }
205
206         void display_takeplace()
207         {
208             cout << "The destination is " << takeplace << "." << endl
209                 << endl;
210         }
211
212         void set_state()
213         {
214             int s;
215
216             cout << "Please enter the state (-2:ready to pick up; -1:picked up; 0:arrived at warehouse; 1:out for
217 delivery 2:delivered)"
218                 << endl;
219             cin >> s;
220             state = s;
221             cout << "Successfully set status!" << endl
222                 << endl;
223         }
224
225         void display_state()
226         {
227             switch (state)
228             {
229             case -2:
230                 cout << "The package is ready to pick up." << endl
231                     << endl;
232                 break;
233             case -1:
234                 cout << "The package is picked up." << endl
235                     << endl;
236                 break;
237             case 0:
238                 cout << "The package arrived at warehouse." << endl
239                     << endl;
240                 break;
241             case 1:
242                 cout << "The package is out for delivery." << endl
243                     << endl;
244                 break;
245             case 2:
```

```
246                 cout << "The package is delivered." << endl
247                     << endl;
248                 break;
249
250             default:
251                 cout << "No status information available." << endl
252                     << endl;
253                 break;
254             }
255         }
256
257         void set_speed()
258         {
259             int temp;
260             cout << "Please enter the service speed (-1:in 3 hours; 0:today; 1:overnight)" << endl;
261             cin >> temp;
262             speed = temp;
263             cout << "Successfully set speed!" << endl
264                 << endl;
265         }
266
267         void display_speed()
268         {
269             switch (speed)
270             {
271             case -1:
272                 cout << "The service speed request is: in 3 hours." << endl
273                     << endl;
274
275                 break;
276             case 0:
277                 cout << "The service speed request is: today." << endl
278                     << endl;
279                 break;
280             case 1:
281                 cout << "The service speed request is: overnight." << endl
282                     << endl;
283                 break;
284             default:
285                 cout << "No service speed information available." << endl
286                     << endl;
287                 break;
288             }
289         }
290
291         void set()
292         {
293             fetchtime.set();
294             gettime.set();
```

```
295
296             set_getplace();
297             set_curplace();
298             curplace = getplace;
299             set_takeplace();
300             set_speed();
301             set_paytype();
302
303             set_ware();
304
305             set_state();
306
307             flag = 1;
308         }
309
310         void print_information()
311         {
312             system("clear");
313             fetchtime.display();
314             gettime.display();
315             display_getplace();
316             display_curplace();
317             display_takeplace();
318             display_speed();
319             display_paytype();
320             display_ware();
321             display_state();
322             paytime.display();
323             taketime.display();
324         }
325
326         void display_payment()
327         {
328             display_paytype();
329             paytime.display();
330             display_ware();
331         }
332
333         void cancel_package()
334         {
335             cancel = 1;
336             flag = 0;
337             cout << "The package has been cancelled" << endl;
338         }
339 };
340
341 class Customer
342 {
343 public:
```

```
344     string company;
345     string linkman;
346     string telephone;
347     string adress;
348     string name;
349     int flag;
350
351     Customer()
352     {
353         flag = 0;
354     }
355
356     void set_name()
357     {
358         string nm;
359
360         cout << "Please enter your name" << endl
361             << endl;
362         getline(cin, nm); //此处输入用户名称
363         if (nm == "")
364         {
365             getline(cin, nm);
366         }
367
368         name = nm;
369
370         cout << "Successfully set name!" << endl
371             << endl;
372     }
373
374     void set_company()
375     {
376         string cpy;
377
378         cout << "Please enter your company name ( if you are individual user, please enter 1 )" << endl
379             << endl;
380         getline(cin, cpy); //此处输入公司名称
381         if (cpy == "")
382         {
383             getline(cin, cpy);
384         }
385
386         company = cpy;
387
388         cout << "Successfully set company!" << endl
389             << endl;
390     }
391
392     void set_linkman()
```

```
393     {
394         string lkm;
395 
396         cout << "Please enter the linkman's name" << endl
397             << endl;
398         getline(cin, lkm); //此处输入联系人名称
399         if (lkm == "")
400         {
401             getline(cin, lkm);
402         }
403 
404         linkman = lkm;
405 
406         cout << "Successfully set linkman!" << endl
407             << endl;
408     }
409 
410     void set_telephone()
411     {
412         string tlp;
413 
414         cout << "Please enter your telephone number" << endl
415             << endl;
416         getline(cin, tlp); //此处输入联系人电话
417         if (tlp == "")
418         {
419             getline(cin, tlp);
420         }
421 
422         telephone = tlp;
423 
424         cout << "Successfully set telephone!" << endl
425             << endl;
426     }
427 
428     void set_adress()
429     {
430         string ads;
431 
432         cout << "Please enter the your shipping address" << endl
433             << endl;
434         getline(cin, ads); //此处输入地址
435         if (ads == "")
436         {
437             getline(cin, ads);
438         }
439 
440         adress = ads;
441 
```

```
442             cout << "Successfully set adress!" << endl
443                 << endl;
444         }
445
446         void set()
447         {
448             set_name();
449             set_company();
450             set_linkman();
451             set_telephone();
452             set_adress();
453
454             cout << "Successfully set all customer messages!" << endl
455                 << endl;
456             flag = 1;
457         }
458
459         void display_company()
460         {
461             if (company == "1")
462             {
463                 cout << "The " << name << " is a individual customer" << endl
464                     << endl;
465             }
466             else
467             {
468                 cout << "company:\t" << company << endl
469                     << endl;
470             }
471         }
472
473         void display_linkman()
474         {
475             cout << "linkman:\t" << linkman << endl
476                 << endl;
477         }
478
479         void display_telephone()
480         {
481             cout << "telephone:\t" << telephone << endl
482                 << endl;
483         }
484
485         void display_adress()
486         {
487             cout << "adress:\t" << adress << endl
488                 << endl;
489         }
490
```

```
491         void display()
492         {
493             cout << "The " << name << "'s all messages:"
494                 << "\ncompany:\t" << company
495                 << "\nlinkman:\t" << linkman
496                 << "\ntelephone:\t" << telephone
497                 << "\nadress:\t" << adress << endl
498                 << endl;
499         }
500 };
501
502 class Deliveryman
503 {
504 public:
505         string deliveryman_phone;
506         string deliveryman_name;
507         string dname;
508         string phone;
509         int flag;
510         Deliveryman()
511         {
512             flag = 0;
513         }
514         void set()
515         {
516
517             cout << "Please enter the diliveryman_name"
518                 << endl;
519             getline(cin, dname); //此处输入联系人名称
520             if (dname == "")
521             {
522                 getline(cin, dname);
523             }
524             deliveryman_name = dname;
525
526             cout << "Please enter the " << dname << "'s telephone number" << endl
527                 << endl;
528             getline(cin, phone); //此处输入联系人电话
529             if (phone == "")
530             {
531                 getline(cin, phone);
532             }
533             deliveryman_phone = phone;
534
535             cout << "Successfully set diliveryman!" << endl
536                 << endl;
537             flag = 1;
538         }
539         void display()
```

```
540         {
541             cout << "deliveryman:\t" << dname
542                 << "\nphone:\t" << phone
543                 << endl;
544         }
545 };
546
547 class List
548 {
549 public:
550         int paytype;
551         double weightsum = 0;
552         int numbersum = 0;
553         double waresum = 0;
554         double paidsum = 0;
555         double paid = 0;
556         double debtsum = 0;
557         string name;
558
559         void sum(Package a)
560         {
561             paytype = a.paytype;
562             weightsum = weightsum + a.weight;
563             numbersum = numbersum + 1;
564             if (paytype == -1)
565             {
566                 paid = a.ware;
567             }
568             else if (paytype == 1)
569             {
570                 paid = 0;
571             }
572             waresum = waresum + a.ware;
573             paidsum = paidsum + paid;
574             debtsum = waresum - paidsum;
575         }
576
577         void display_paytype()
578         {
579             cout << "paytype:\t" << paytype << endl
580                 << endl;
581         }
582
583         void display_weightsum()
584         {
585             cout << "weightsum:\t" << weightsum << endl
586                 << endl;
587         }
588
```

```
589         void display_numbersum()
590         {
591             cout << "numbersum:\t" << numbersum << endl
592                 << endl;
593         }
594
595         void display_waresum()
596         {
597             cout << "waresum:\t" << waresum << endl
598                 << endl;
599         }
600
601         void display_debtsum()
602         {
603             cout << "debtsum:\t" << debtsum << endl
604                 << endl;
605         }
606
607         void display()
608         {
609             cout << "The " << name << "'s list:"
610                 << "\npaytype:\t" << paytype
611                 << "\nweightsum:\t" << weightsum
612                 << "\nnumbersum:\t" << numbersum
613                 << "\nwaresum:\t" << waresum
614                 << "\ndebtsum:\t" << debtsum << endl
615                 << endl;
616         }
617 };
618
619 //Package package[N];
620 //Customer customer[N];
621 //Deliveryman deliveryman[N];
622 Customer customer[10];
623 Package package[100];
624 Deliveryman deliveryman[10];
625 List list[10];
626
627 int login();
628 void register_customer();
629 void register_deliveryman();
630 void modify_package();
631 void package_manage();
632 void modify_customer(int cnum);
633 void customer_manage();
634 void dilivery_manage();
635 void account_administrator();
636 void account_deliveryman();
637 void account_company();
```

```
638 void account_person();
639 //void search_package(Time t);
640
641 void search_package(Time t)
642 {
643     int i;
644     int morning = 0;
645     if (t.hour >= 12)
646         morning = 1;          //判断输入的时间是上午还是下午
647     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
648 裹的时间
649     {
650         if (package[i].fetchtime.month == t.month && package[i].fetchtime.date == t.date) //此处不用判断
651 flag
652         {
653             if (morning == 0 && package[i].fetchtime.hour < 12)
654                 package[i].print_information();
655             if (morning == 1 && package[i].fetchtime.hour >= 12)
656                 package[i].print_information();
657         }
658         // switch (t.flag)
659         // {
660         // case 1:
661         //     if (package[i].taketime.hour <= 12)
662         //     {
663         //         package[i].print_information();
664         //     }
665         //     break;
666         // case 2:
667         //     if (package[i].taketime.hour <= 12)
668         //     {
669         //         package[i].print_information();
670         //     }
671         //     break;
672         // case 0:
673         //     package[i].print_information();
674         // }
675     }
676 }
677
678 void register_customer()
679 {
680     int cnum = 0;
681     while (customer[cnum].flag == 1)
682     {
683         ++cnum;
684     }
685     customer[cnum].set();
686     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;
```

```
736             cout << "Enter the package number" << endl;
737             cin >> pnum;
738             cout << "1_fetchtime\n";
739             cout << "2_gettime\n";
740             cout << "3_getplace\n";
741             cout << "4_curplace\n";
742             cout << "5_takeplace\n";
743             cout << "6_speed\n";
744             cout << "7_state\n";
745             cout << "9_Back To Previous" << endl;
746             cout << "0_Back To Mainmenu" << endl;
747             mo = 0;
748             cin >> mo;
749             switch (mo)
750             {
751             case 1:
752                 package[pnum].fetchtime.set();
753                 cout << "Successfully modified!";
754                 break;
755             case 2:
756                 package[pnum].gettime.set();
757                 cout << "Successfully modified!";
758                 break;
759             case 3:
760                 package[pnum].set_getplace();
761                 cout << "Successfully modified!";
762                 break;
763             case 4:
764                 package[pnum].set_curplace();
765                 cout << "Successfully modified!";
766                 break;
767             case 5:
768                 package[pnum].set_takeplace();
769                 cout << "Successfully modified!";
770                 break;
771             case 6:
772                 package[pnum].set_speed();
773                 cout << "Successfully modified!";
774                 break;
775             case 7:
776                 package[pnum].set_state();
777                 cout << "Successfully modified!";
778                 break;
779             case 9:
780                 cccc = 1;
781                 ccc = 0;
782                 break;
783             case 0:
784                 cccc = 1;
```

```
785                     ccc = 1;
786                     cc = 0;
787                     break;
788             default:
789                     cout << "order error, enter any value to return" << endl;
790                     cin >> skip;
791                     break;
792             }
793     }
794     cccc = 0;
795 }
796
797 void modify_customer(int cnum)
798 {
799     while (cccc == 0)
800     {
801             cout << "1_company" << endl;
802             cout << "2_linkman" << endl;
803             cout << "3_telephone" << endl;
804             cout << "4_adress" << endl;
805             cout << "5_name" << endl;
806             cout << "9_Back To Previous" << endl;
807             cout << "0_Back To Mainmenu" << endl;
808             int mo = 0;
809             cin >> mo;
810             switch (mo)
811             {
812             case 1:
813                     customer[cnum].set_company();
814                     break;
815             case 2:
816                     customer[cnum].set_linkman();
817                     break;
818             case 3:
819                     customer[cnum].set_telephone();
820                     break;
821             case 4:
822                     customer[cnum].set_adress();
823                     break;
824             case 5:
825                     customer[cnum].set_name();
826                     break;
827             case 9:
828                     cccc = 1;
829                     ccc = 0;
830                     break;
831             case 0:
832                     cccc = 1;
833                     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;
```

```
883                 package[pnum].print_information();
884             }
885             break;
886             case 3: //修改包裹信息
887                 modify_package();
888                 break;
889             case 4:
890                 cout << "Enter the package number" << endl;
891                 int dcnum2;
892                 cin >> dcnum2;
893                 package[dcnum2].set_state();
894                 break;
895             case 9:
896             {
897                 ccc = 1;
898                 cc = 0;
899             }
900             break;
901             case 0:
902             {
903                 ccc = 1;
904                 cc = 0;
905             }
906             break;
907             default:
908             {
909                 cout << "order error, enter any value to return" << endl;
910                 cin >> skip;
911             }
912             break;
913             }
914         }
915         ccc = 0;
916 }
917
918 void customer_manage()
919
920 { //用户管理系统
921         while (ccc == 0)
922         {
923             cout << "1_register customer information" << endl;
924             cout << "2_query customer information" << endl;
925             cout << "3_modify customer information" << endl;
926             cout << "4_query package information" << endl;
927             cout << "9_Back To Previous" << endl;
928             cout << "0_Back To Mainmenu" << endl;
929             int con;
930             cin >> con;
931             switch (con)
```

```
932             {
933             case 1:
934                 register_customer();
935                 break;
936             case 2:
937                 cout << "Enter the customer number" << endl;
938                 int cnum2;
939                 cin >> cnum2;
940                 customer[cnum2].display();
941                 break;
942             case 3:
943                 cout << "Enter the customer number" << endl;
944                 int cnum3;
945                 cin >> cnum3;
946                 modify_customer(cnum3);
947                 break;
948             case 4:
949                 cout << "Enter the customer number" << endl;
950                 int cnum4;
951                 cin >> cnum4;
952                 cnum4 = 10 * cnum4;
953                 while (package[cnum4].flag == 1)
954                 {
955                     package[cnum4].print_information();
956                     ++cnum4;
957                 }
958             case 9:
959                 ccc = 1;
960                 cc = 0;
961                 break;
962             case 0:
963                 ccc = 1;
964                 cc = 0;
965                 break;
966             default:
967                 cout << "order error, enter any value to return" << endl;
968                 cin >> skip;
969                 break;
970             }
971         }
972         ccc = 0;
973 }
974
975 void dilivery_manage()
976 { //快递员管理系统
977         while (ccc == 0)
978         {
979             cout << "1_register dilivery information" << endl;
980             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)
            {
```

```
1030                customer_manage();
1031            }
1032            else if (order == 3)
1033            {
1034                dilivery_manage();
1035            }
1036            else if (order == 0)
1037            {
1038                cc = 1;
1039                c = 0;
1040            }
1041            else
1042            {
1043                cout << "order error, enter any value to return" << endl;
1044                cin >> skip;
1045            }
1046        }
1047        cc = 0;
1048    }
1049
1050    void account_deliveryman() //快递员系统
1051    {
1052        int order;
1053        while (cc == 0)
1054        {
1055            cout << "1_query package information" << endl;
1056            cout << "2_modify package status" << endl;
1057            cout << "3_query package information in this period" << endl;
1058            cout << "0_Relanding" << endl;
1059            order = 0;
1060            Time period("period");
1061            cin >> order;
1062
1063            switch (order)
1064            {
1065            case 1:
1066                cout << "Enter the package number" << endl;
1067                int dcnum1;
1068                cin >> dcnum1;
1069                package[dcnum1].print_information();
1070                break;
1071            case 2:
1072                cout << "Enter the package number" << endl;
1073                int dcnum2;
1074                cin >> dcnum2;
1075                package[dcnum2].set_state();
1076                break;
1077            case 3:
1078                cout << "Enter the period" << endl;
```

```
1079
1080                    period.set();
1081                    search_package(period);
1082
1083                    break;
1084            case 0:
1085                    cc = 1;
1086                    c = 0;
1087                    break;
1088            default:
1089                    cout << "order error, enter any value to return" << endl;
1090                    cin >> skip;
1091                    break;
1092            }
1093    }
1094    cc = 0;
1095 }
1096
1097 void account_company()
1098 {
1099    int order;
1100    while (cc == 0)
1101    {
1102            cout << "Enter company number" << endl;
1103            int cpnum;
1104            cin >> cpnum;
1105            if (customer[cpnum].flag == 0)
1106            {
1107                    register_customer();
1108            }
1109            cout << "1_modify account" << endl;
1110            cout << "2_register package information" << endl;
1111            cout << "3_modify package information" << endl;
1112            cout << "4_query package information" << endl;
1113            cout << "5_query order records" << endl;
1114            cout << "6_query monthly statement" << endl;
1115            cout << "7_months to be paid bill" << endl;
1116            cout << "8_cancel the package" << endl;
1117            cout << "0_relanding" << endl;
1118            order = 0;
1119            cin >> order;
1120            switch (order)
1121            {
1122            case 1:
1123                    modify_customer(cpnum);
1124                    break;
1125            case 2:
1126                    int pnum2;
1127                    pnum2 = cpnum * 10;
```

```
1128            while (package[pnum2].flag == 1)
1129            {
1130                ++pnum2;
1131            }
1132            package[pnum2].set();
1133            list[cpnum].sum(package[pnum2]);
1134            break;
1135        case 3:
1136            modify_package();
1137            break;
1138        case 4:
1139            cout << "Enter the package number" << endl;
1140            int pnum4;
1141            cin >> pnum4;
1142            package[pnum4].print_information();
1143            break;
1144        case 5:
1145            cpnum = 10 * cpnum;
1146            int i;
1147            for (i = 0; i < 10; i++)
1148            {
1149                if (package[cpnum].flag == 1 || package[cpnum].cancel == 0) //                    {
1150                                                                    //输出全部订单
1151                    cout << "pack" << cpnum << ":";
1152                    package[cpnum].print_information();
1153                    ++cpnum;
1154                }
1155            }
1156            break;
1157        case 6:
1158            list[cpnum].display();
1159            break;
1160        case 7:
1161            list[cpnum].display_debtsum();
1162            break;
1163        case 8:
1164            cout << "Enter the package number" << endl;
1165            while (package[cpnum].flag == 1)
1166            {
1167                cout << "pack" << cpnum << ":";
1168                package[cpnum].print_information();
1169                cout << endl;
1170                ++cpnum;
1171            }
1172            int pnum8;
1173            cin >> pnum8;
1174            if (package[pnum8].state == -2)
1175            {
1176                package[pnum8].cancel_package();
```

```
1177                    }
1178                    else
1179                            cout << "Sorry,it cannot be cancelled" << endl;
1180                    break;
1181            case 0:
1182                    cc = 1;
1183                    c = 0;
1184                    break;
1185            default:
1186                    cout << "order error, enter any value to return" << endl;
1187                    cin >> skip;
1188                    break;
1189            }
1190    }
1191    cc = 0;
1192 }
1193
1194 void account_person()
1195 {
1196    cout << "Enter person number" << endl;
1197    int psnum;
1198    cin >> psnum;
1199    if (customer[psnum].flag == 0)
1200    {
1201            register_customer();
1202    }
1203    int order;
1204    while (cc == 0)
1205    {
1206            cout << "1_modify account" << endl;
1207            cout << "2_register package information" << endl;
1208            cout << "3_modify package information" << endl;
1209            cout << "4_query package information" << endl;
1210            cout << "5_querying records" << endl;
1211            cout << "6_modify package status" << endl;
1212            cout << "7_months to be paid bill" << endl;
1213            cout << "8_cancel the package" << endl;
1214            cout << "0_Relanding" << endl;
1215            order = 0;
1216            cin >> order;
1217            switch (order)
1218            {
1219            case 1:
1220                    modify_customer(psnum);
1221                    break;
1222            case 2:
1223                    while (package[psnum].flag == 1)
1224                    {
1225                            ++psnum;
```

```
1226            }
1227            package[psnum].set();
1228            list[psnum].sum(package[psnum]);
1229            break;
1230        case 3:
1231            modify_package();
1232            break;
1233        case 4:
1234            cout << "Enter the package number" << endl;
1235            int pnum4;
1236            cin >> pnum4;
1237            package[pnum4].print_information();
1238            break;
1239        case 5:
1240            psnum = 10 * psnum;
1241            int i;
1242            for (i = 0; i < 10; i++)
1243            {
1244                if (package[psnum].flag == 1 || package[psnum].cancel == 1)
1245                { //输出全部订单
1246                    cout << "pack" << psnum << ":";
1247                    package[psnum].print_information();
1248                    ++psnum;
1249                }
1250            }
1251            break;
1252        case 6:
1253            for (i = 0; i < 10; i++)
1254            {
1255                if (package[psnum].flag == 1)
1256                { //输出全部订单
1257                    cout << "pack" << psnum << ":";
1258                    package[psnum].print_information();
1259                    ++psnum;
1260                }
1261            }
1262            cout << "Enter the package number" << endl;
1263            int psnum6;
1264            cin >> psnum6;
1265            package[psnum6].set_state();
1266            break;
1267        case 7:
1268            list[psnum].display_debtsum();
1269            break;
1270        case 8:
1271            cout << "Enter the package number" << endl;
1272            while (package[psnum].flag == 1)
1273            {
1274                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;
```

```
1324            case 0:
1325                c = 1;
1326                break;
1327            default:
1328                cout << "login_ID is error" << endl;
1329                break;
1330            }
1331
1332            cout << "Successfully exit" << endl << endl;
1333            login_account = 0;
1334        }
1335        c = 0;
1336        return 0;
1337    }
```