



中央民族大学
MINZU UNIVERSITY OF CHINA

《面向对象程序设计课程设计》 课程设计文档

学院：信息工程学院 专业班级：21 级计算机科学与技术

小组人数：1 人

组员：

学号：21011636

姓名：陈俞宁

设计题目：教务管理系统、仓库管理信息系统

一、教务管理系统

(1) 基本思想及设计思路

这个系统需要有两个基本大类：学生组（Student_group）、课程组（Courses_group）。

先看学生组，学生组类的基本数据是学生个体数组和学生人数，里面需要包含能够添加、删除、查询某一个学生个体的函数，但不涉及学生个体的具体特性。因此，我们需要一个学生个体类来概括学生身上的特性。根据要求，学生个体类包括：基本信息（学号、姓名、性别）、奖惩信息、选课信息（其中包含每门课的课程成绩）。基本信息的存储和改查可以直接做到学生个体类里，而奖惩信息和选课信息需要另外建立两个类来描述。我将学生课程成绩管理做到了学生选课中，在学生选课中可以选择任意一门课来改变其成绩。

再看课程组，课程组类的基本数据是课程个体数组和课程数，里面包含添加、删除、显示课程的函数，但不涉及某一个课程个体的具体内容。课程个体类包括：课程 id、任课老师数、任课老师个体数组、学生数、学生学号数组。课程个体类需要实现的功能有：添加、删除、显示教师，添加、删除学生，显示某一位学生和显示全部学生的函数。

要注意，课程组与学生选课是需要有联系的，当某个课程个体添加了某一位学生，那么这位学生的选课列表中将会自动添加该课程，反之，如果删除，则同理，课程删除学生，学生的选课列表中的该课也会自动删除。当某一个课程停课/不开的时候，学生选课列表也会自动将该课程删除。如此实现联动。

除此之外，我还添加了教师评价系统（扩展附加功能），学生可以对某一课程的某一任课老师进行评分，从而实现双向互评。

(2) 代码说明

1. 学生组：

```
class Student_group
{
public:
    int num;
    Student stu[400];
    Student_group(){num=0;}
    void add_(int id_,string name_,char sex_)
    {
    }
    void delete_(int n)
    {
    }
    void display_all()
    {
    }
    void display_one(int id_)
    {
    }
};
```

分别实现了增删显示的功能。

删 (delete_):

```
void delete_(int n)
{
    for(int i=n+1;i<num;i++)
    {
        stu[i-1]=stu[i];
    }
    num--;
}
```

输入参数为序号，在数组从该序号开始，后一个单元覆盖前一个单元，之后将 num 减去 1，表示删除。

2. 学生个体:

```
class Student
{
public:
    int id;
    string name;
    char sex;
    Pp_group t;
    Courses_selected_group c;
    friend class Courses_group;
    friend class Student_group;
    void change_base(int id_,string name_,char sex_)
    {
    }
    void display_all()
    {
    }
    void display_base()
    {
    }
};
```

Pp_group 是奖惩类，Courses_selected_group 是选课类。

3. 奖惩类:

```

struct Prize_punish
{
    string name;
    char nature;
};
class Pp_group
{
public:
    int num;
    Prize_punish sin[20];
    Pp_group(){num=0;}
    void add_(string name_,char nature_)
    {
    }
    void delete_(int n)
    {
    }
    void change_(int n,string name_,char nature_)
    {
    }
    void display_one(string name_)
    {
    }
    void display_all()
    {
    }
};

```

奖惩类包括奖惩数目、奖惩结构体数组、增删改查函数。

奖惩结构体包括奖惩的名称和性质（奖还是惩），对于 display_one 函数，实现如下：

```

void display_one(string name_)
{
    int i;
    for(i=0;i<num;i++)
    {
        if(sin[i].name==name_)
        {
            cout<<sin[i].name<<" ";
            if(sin[i].nature=='T') cout<<"奖"<<endl;
            else cout<<"惩"<<endl;
            break;
        }
    }
    if(i==num) cout<<"查询失败！"<<endl;
}

```

输入量是奖惩的名称，先从奖惩结构体数组里挨个查询是否有同名的，如果有同名的，则显示其名字和性质，如果没有（i==num）则显示“查询失败！”。

4. 选课组：

```

61 - };
62 struct Courses_selected
63 {
64     int id;
65     double mark;
66 };
67 class Courses_selected_group
68 {
69     public:
70     int num;
71     Courses_selected sin[10];
72     Courses_selected_group(){num=0;}
73     void add_(int id_)
74     {
79     void delete_(int n)
80     {
87     void delete_1(int id_)
88     {
102    void change_mark(int n,double mark_)
103    {
106    void display_all()
107    {
114    };

```

结构基本与奖惩组相同。

5. 学生组类的申请：

```

171 {
183 };
184
185 Student_group a;
186
187 struct Teacher

```

因为在之后的课程管理系统中需要操作学生类的内容，因此在定义课程组类前先定义了学生组类的对象 a。

6. 课程组：

在定义课程组前，使用一个全局变量表来使课程编号和课程名称对应起来，方便在之后的函数中只存编号，需要名称的时候可以随时查表。

```

3 #define Max 6
4 using namespace std;
5 string sta[15]={"离散","线代","模电","数据结构","汇编","数电"};
6 struct Prize_punish

```

```

279  };
280  class Courses_group
281  {
282      public:
283          int num;
284          Courses_group()
285          {
286              num=0;
287          }
288          Courses c[6];
289          void add_(int id_)
290          {
294              void delete_(int n)
295              {
302              void display_all()
303              {
309  };

```

课程组包括课程数和课程个体数组（最大六门课，但是可以扩充）以及增删显示课程的函数。

7. 课程个体：

```

186
187 struct Teacher
188 {
189     int id;
190     string name;
191     double mark;
192 };
193 class Courses
194 {
195     public:
196         int id;
197         int tea_num;
198         Teacher t[5];
199         int stu_num;
200         int stu_list[100];
201 Courses(){
205     void add_tea(int id_,string name_)
206     {
216     void delete_tea(int n)
217     {
224     void display_tea()
225     {
231     void add_stu(int id_)//课程端加学生，学生端也应该加课程
232     {
236     void delete_stu(int n)
237     {
244     void display_stu_one(int id_)//显示学生
245     {
262     void display_stu_all()
263     {
278
279 };

```

在类中我仅实现了简单的添加学生功能，而同步添加算法则放在了主函数里，如下：

```

611 case 8: //课程添加学生，学生端也要添加这个课程
612     int n;
613     int id_;
614     b.display_all();
615     cout<<"请输入课程序号(第一列): ";
616     cin>>n;
617     cout<<"请输入学生id: ";
618     cin>>id_;
619     b.c[n-1].add_stu(id_);
620     for(int i=0;i<a.num;i++)
621     {
622         if(a.stu[i].id==id_)
623         {
624             a.stu[i].c.add_(b.c[n-1].id); //学生端添加课程
625             break;
626         }
627     }
628     break;

```

其功能逻辑为：先显示所有的课程（已标序号），之后提示用户选择要添加

学生的课程以及学生 id，并将此 id 的学生添加到该课程中去。之后遍历全体学生列表，如果 id 和此添加学生的 id 相同，则在此学生的选课表中通过 id 方式添加课程，从而实现同步。

8. 主函数：

```
354 int main()
355 {
356     Courses_group b;
357     int tag;
358     bool f=1;
359     while(f)
360     {
361
362         display_mainmenu();
363         cin>>tag;
364         cout<<endl;
365         switch(tag)
366         {
367             case 1:{
368
369             case 2:{
370
371             case 3:{
372
373             case 0:{
374                 f=0;
375                 break;
376             }
377             default:{
378                 cout<<"输入有误!"<<endl;
379                 break;
380             }
381         }
382     }
383     return 0;
384 }
```

用 while 函数来实现多次读取命令，用 Switch 函数来判别需要执行的操作。

同时设计一级主菜单和二级菜单(学生菜单、课程菜单)，使程序更有条理，更人性化，提高用户使用体验。

(3) 运行结果

1. 添加学生


```
D:\陈俞宁\作业\c++大作业\教务管理系统\EAS.exe
(3)教师评分系统
(0)退出
-----
请输入序号并按回车以选择: 1
-----学生菜单-----
(1)显示所有学生
(2)添加学生
(3)删除学生
(4)查询学生全部信息
(5)修改基本信息
(6)添加奖惩信息
(7)删除奖惩信息
(8)修改奖惩信息
(9)课程成绩录入或修改
(10)课程成绩查询
(0)退出
-----
请输入序号并按回车以选择: 2
请输入学号: 10001
请输入姓名: 张三
请输入性别(男性->M, 女性->F): M
      按
```

2. 添加奖惩

```
-----学生菜单-----
(1)显示所有学生
(2)添加学生
(3)删除学生
(4)查询学生全部信息
(5)修改基本信息
(6)添加奖惩信息
(7)删除奖惩信息
(8)修改奖惩信息
(9)课程成绩录入或修改
(10)课程成绩查询
(0)退出
-----
请输入序号并按回车以选择: 6
1      10001 张三      M
2      10002 李四      F
请输入学生序号(第一列): 1
请输入名称: 专业奖学金
请输入性质(奖->T, 惩->F): T
      按
```

3. 添加课程

```

-----课程菜单-----
(1)显示所有课程
(2)添加课程
(3)删除课程
(4)显示所有任课老师
(5)添加任课老师
(6)删除任课老师
(7)显示所有选某课的学生(选课系统)
(8)添加选某课的学生(选课系统)
(9)删除选某课的学生(选课系统)
(10)查询某一个选此课的学生
(0)退出
-----
请输入序号并按回车以选择: 2

0离散 1线代 2模电 3数据结构 4汇编 5数电
请输入课程代码:
1

```

4. 添加任课老师

```

-----课程菜单-----
(1)显示所有课程
(2)添加课程
(3)删除课程
(4)显示所有任课老师
(5)添加任课老师
(6)删除任课老师
(7)显示所有选某课的学生(选课系统)
(8)添加选某课的学生(选课系统)
(9)删除选某课的学生(选课系统)
(10)查询某一个选此课的学生
(0)退出
-----
请输入序号并按回车以选择: 5

1      离散
2      线代
请输入课程序号(第一列): 1
请输入老师工号: 90001
请输入老师姓名: 王老师

```

5. 对课程添加学生

```

-----课程菜单-----
(1)显示所有课程
(2)添加课程
(3)删除课程
(4)显示所有任课老师
(5)添加任课老师
(6)删除任课老师
(7)显示所有选某课的学生(选课系统)
(8)添加选某课的学生(选课系统)
(9)删除选某课的学生(选课系统)
(10)查询某一个选此课的学生
(0)退出

请输入序号并按回车以选择: 8

1      离散
2      线代
请输入课程序号(第一列): 1
请输入学生id: 10001
          张三

```

6. 显示所有学生

```

-----学生菜单-----
(1)显示所有学生
(2)添加学生
(3)删除学生
(4)查询学生全部信息
(5)修改基本信息
(6)添加奖惩信息
(7)删除奖惩信息
(8)修改奖惩信息
(9)课程成绩录入或修改
(10)课程成绩查询
(0)退出

请输入序号并按回车以选择: 1

1      10001 张三      M
2      10002 李四      F

```

7. 查找学生

```

-----学生菜单-----
(1)显示所有学生
(2)添加学生
(3)删除学生
(4)查询学生全部信息
(5)修改基本信息
(6)添加奖惩信息
(7)删除奖惩信息
(8)修改奖惩信息
(9)课程成绩录入或修改
(10)课程成绩查询
(0)退出
-----
请输入序号并按回车以选择: 4

请输入所查询学生学号: 10001
10001 张三 M
1 离散 -1
2 线代 -1
专业奖学金 奖

```

因为没有录入课程成绩，所以显示-1.

8. 显示所有课程

```

-----课程菜单-----
(1)显示所有课程
(2)添加课程
(3)删除课程
(4)显示所有任课老师
(5)添加任课老师
(6)删除任课老师
(7)显示所有选某课的学生(选课系统)
(8)添加选某课的学生(选课系统)
(9)删除选某课的学生(选课系统)
(10)查询某一个选此课的学生
(0)退出
-----
请输入序号并按回车以选择: 1

1 离散
2 线代
-----菜单-----

```

9. 显示任课老师

```
-----课程菜单-----
(1)显示所有课程
(2)添加课程
(3)删除课程
(4)显示所有任课老师
(5)添加任课老师
(6)删除任课老师
(7)显示所有选某课的学生(选课系统)
(8)添加选某课的学生(选课系统)
(9)删除选某课的学生(选课系统)
(10)查询某一个选此课的学生
(0)退出

请输入序号并按回车以选择：4

1      离散
2      线代
请输入课程序号(第一列)：1
90001 王老师
```

10. 显示选离散的学生

```
-----课程菜单-----
(1)显示所有课程
(2)添加课程
(3)删除课程
(4)显示所有任课老师
(5)添加任课老师
(6)删除任课老师
(7)显示所有选某课的学生(选课系统)
(8)添加选某课的学生(选课系统)
(9)删除选某课的学生(选课系统)
(10)查询某一个选此课的学生
(0)退出

请输入序号并按回车以选择：7

1      离散
2      线代
请输入课程序号(第一列)：1
1      10001 张三 M
2      10002 李四 F
```

11. 查询某一个选离散的学生

```

-----课程菜单-----
(1)显示所有课程
(2)添加课程
(3)删除课程
(4)显示所有任课老师
(5)添加任课老师
(6)删除任课老师
(7)显示所有选某课的学生(选课系统)
(8)添加选某课的学生(选课系统)
(9)删除选某课的学生(选课系统)
(10)查询某一个选此课的学生
(0)退出
-----
请输入序号并按回车以选择: 10

1      离散
2      线代
请输入课程序号(第一列): 1
请输入学号: 10001
10001 张三 M

```

12. 录入成绩

```

-----学生菜单-----
(1)显示所有学生
(2)添加学生
(3)删除学生
(4)查询学生全部信息
(5)修改基本信息
(6)添加奖惩信息
(7)删除奖惩信息
(8)修改奖惩信息
(9)课程成绩录入或修改
(10)课程成绩查询
(0)退出
-----
请输入序号并按回车以选择: 9

1      10001 张三      M
2      10002 李四      F

请输入学生序号(第一列): 1
1      离散 -1
2      线代 -1
请输入课程序号(第一列): 1
请输入成绩: 99

```

13. 查询所选课程及成绩

```

-----学生菜单-----
(1)显示所有学生
(2)添加学生
(3)删除学生
(4)查询学生全部信息
(5)修改基本信息
(6)添加奖惩信息
(7)删除奖惩信息
(8)修改奖惩信息
(9)课程成绩录入或修改
(10)课程成绩查询
(0)退出
-----
请输入序号并按回车以选择: 10

1      10001 张三      M
2      10002 李四      F

请输入学生序号(第一列): 1
1      离散 99
2      线代 95

```

14. 删除选某课的学生

```

-----
请输入序号并按回车以选择: 9

1      离散
2      线代
请输入课程序号(第一列): 1
1      10001 张三 M
2      10002 李四 F

请输入学生序号: 2
-----菜单-----
(0)学生信息管理

```

删除后: 只剩张三

```

-----
请输入序号并按回车以选择: 7

1      离散
2      线代
请输入课程序号(第一列): 1
1      10001 张三 M

```

15. 删除任课老师

```

(5)添加任课老师
(6)删除任课老师
(7)显示所有选某课的学生(选课系统)
(8)添加选某课的学生(选课系统)
(9)删除选某课的学生(选课系统)
(10)查询某一个选此课的学生
(0)退出

```

 请输入序号并按回车以选择: 6

```

1      离散
2      线代
请输入课程序号(第一列): 1
90001 王老师
请输入老师序号(第一列): 1

```

16. 教师评分（扩展附加功能）

 请输入序号并按回车以选择: 3

```

1      离散
2      线代
请输入所评课程序号: 1
任课老师如下:
90002 李老师
请输入所评老师工号: 90002
请输入分数: 100

```

17. 修改奖惩信息

```

(0)退出
-----
请输入序号并按回车以选择: 8

1      10001 张三      M
2      10002 李四      F

请输入学生序号(第一列): 1
专业奖学金 奖
国家奖学金 奖
请输入奖惩序号(第一列): 1
请输入名称: 专业大奖学金
请输入性质(奖->T, 惩->F): T

```

修改后: 变成“专业大奖学金”

```

请输入序号并按回车以选择: 4

请输入所查询学生学号: 10001
10001 张三      M
1      离散 99
2      线代 95
专业大奖学金 奖
国家奖学金 奖

```

18. 修改学生基本信息


```

请输入序号并按回车以选择：5

1      10001 张三      M
2      10002 李四      F

请输入要修改的学生序号(第一列)：2
请输入学号：10005
请输入姓名：李四
请输入性别(男性->M, 女性->F)：F
-----菜单-----

```

修改后：李四的学号 10002->10005

```

-----菜单-----
请输入序号并按回车以选择：1

1      10001 张三      M
2      10005 李四      F

```

19. 删除奖惩信息

```

-----菜单-----
请输入序号并按回车以选择：7

1      10001 张三      M
2      10005 李四      F

请输入学生序号(第一列)：1
专业大奖学金 奖
国家奖学金 奖
请输入奖惩序号(第一列)：1
-----菜单-----

```

删除了专业大奖学金

```

-----菜单-----
请输入序号并按回车以选择：4

请输入所查询学生学号：10001
10001 张三      M
1      离散 99
2      线代 95
国家奖学金 奖
-----菜单-----

```

20. 删除离散课程

```

-----菜单-----
请输入序号并按回车以选择：3

1      离散
2      线代
请输入课程序号(第一列)：1
-----菜单-----

```

删除后张三的课表里也没有离散了

```

请输入序号并按回车以选择: 4
请输入所查询学生学号: 10001
10001 张三 M
1 线代 95
国家奖学金 奖

```

21. 删除学生

```

请输入序号并按回车以选择: 3
1 10001 张三 M
2 10005 李四 F
请输入要删除的学生序号(第一列): 2
-----菜单-----

```

二、仓库信息管理系统

(1) 基本思想及设计思路

基本结构与教务系统类似，包含四个大类：库存、入库信息、出库信息、工人信息，扩展附加功能为“工人信息管理”。

先看库存类，库存类存储货物个体数组以及货物种数，包含入库、出库、改变出入价、显示货物的功能函数。而其中的货物个体类有货物的基本信息，以及一些控制/改变货物基本信息的函数。

入库出库以及工人信息都包含信息（Info）类对象数组，且每个类都设有增删改查四类函数，涵盖了各项功能。

值得一提的是，库存类的入库函数会与入库信息类的增加函数联动，也就是说，入库的同时会自动增添一条入库信息来记录此次入库，出库时也与之类似。

(2) 代码说明

1. 库存类

```

26 class Inventory
27 {
28     public:
29         int num;
30         Good g[400];
31         void add_(int id_,int num_,string name_,double InPrice_,double OutPrice_)
32         {
33             // ...
34         }
35         void out_(int id_,int num_)
36         {
37             // ...
38         }
39         void change_(int id_,double InPrice_,double OutPrice_)
40         {
41             // ...
42         }
43         void display_all()
44         {
45             // ...
46         }
47     };

```

2. 库存个体类

```

3  class Good
4  {
5      public:
6          int id;
7          int num;
8          string name;
9          double InPrice;
10         double OutPrice;
11         Good(){num=0;}
12         void SetId(int id_){id=id_;}
13         void SetName(string name_){name=name_;}
14         void SetInPrice(double InPrice_){InPrice=InPrice_;}
15         void SetOutPrice(double OutPrice_){OutPrice=OutPrice_;}
16         void import_(int add_){num+=add_;}
17         void export_(int sub_)
18         {
19             if((num-sub_)<0) cout<<"error!"<<endl;
20             else
21             {
22                 num-=sub_;
23             }
24         }
25     };
26     class Inventory

```

这描述了一种货物的基本信息，并设置相关函数。

3. 信息类（基础类）

```

95  class Info
96  {
97      public:
98          int id;
99          int num;
100         int hours;
101         int mins;
102         string name;
103         void SetId(int id_){id=id_;}
104         void SetNum(int num_){num=num_;}
105         void SetHours(int hours_){hours=hours_;}
106         void SetMins(int mins_){mins=mins_;}
107         void SetName(string name_){
108             name=name_;
109         }
110     };

```

包含各种用以描述货物/工人的信息，以及一些功能函数（主要是设置函数）。其中 hours 和 mins 描述了入库/出库时间，id 既可以描述货物编码，又可以描述工人工号。

4. 入库信息类（由信息类作为基础类）

```

152 class ImportInfo
153 {
154     public:
155         int num;
156         Info a[400];
157         ImportInfo(){
158             void add_(int id_,int num_,int hours_,int mins_)
159             {
160             void delete_(int n)
161             {
162             void change_(int n,int num_,int hours_,int mins_)
163             {
164             void display_all()
165             {
166         };

```

其中 add 添加函数只能由入库时自动调用，而不能手动凭空添加一条入库信息，保证了信息的准确性。

5. 出库信息类

```

195 class ExportInfo
196 {
197     public:
198         int num;
199         Info a[400];
200         ExportInfo(){
201             void add_(int id_,int num_,int hours_,int mins_)
202             {
203             void delete_(int n)
204             {
205             void change_(int n,int num_,int hours_,int mins_)
206             {
207             void display_all()
208             {
209         };

```

与上类似。

6. 工人信息类（扩展附加功能）

```

111 class WorkerInfo
112 {
113     public:
114         int num;
115         Info a[100];
116         WorkerInfo(){
117             void add_(int id_,string name_)
118             {
119             void delete_(int n)
120             {
121             void display_all()
122             {
123         };

```

与上类似，但是工人信息可以手动调用 add 函数。

7. 主函数

```
290 int main()
291 {
292     Inventory a;
293     ImportInfo b;
294     ExportInfo c;
295     WorkerInfo d;
296     int tag;
297     bool f=1;
298     while(f)
299     {
300         display_mainmenu();
301         cin>>tag;
302         cout<<endl;
303         switch(tag)
304         {
305             // ...
306         }
307     }
308     return 0;
309 }
```

同样是用 while-Switch 结构，需要强调的是在入库时的联动：

```
315 case 2:{
316     int id_,num_,hours_,mins_;
317     string name_;
318     double InPrice_,OutPrice_;
319     cout<<"请输入货号: ";
320     cin>>id_;
321     cout<<"请输入名称: ";
322     cin>>name_;
323     cout<<"请输入进货量: ";
324     cin>>num_;
325     cout<<"请输入进价: ";
326     cin>>InPrice_;
327     cout<<"请输入售价: ";
328     cin>>OutPrice_;
329     cout<<"请输入进货时间（格式为 时钟+空格+分钟）: ";
330     cin>>hours_>>mins_;
331     a.add_(id_,num_,name_,InPrice_,OutPrice_);
332     b.add_(id_,num_,hours_,mins_);
333     break;
334 }
```

入库时输入所有信息，在调用库存类的 add 函数后，还要调用库存信息类的 add 函数，从而实现入库的同时，同时增加一条入库信息。

(3) 运行结果

1. 入库

```

-----物资/库存管理菜单-----
(1)显示所有项
(2)进库
(3)出库
(4)修改进出价
(0)退出
-----
请输入序号并按回车以选择：2

请输入货号：10001
请输入名称：薯片
请输入进货量：50
请输入进价：15
请输入售价：35
请输入进货时间（格式为 时钟+空格+分钟）：19 56

```

2. 显示所有库存

```

请输入序号并按回车以选择：1

货号   名称   进价   出价   数量
10001   薯片   15     35     50
10002   钢笔   40     80     100

```

3. 修改进出价

```

-----
请输入序号并按回车以选择：4

请输入货号：10001
请输入进价：20
请输入售价：48

```

修改后：

```

请输入序号并按回车以选择：1

货号   名称   进价   出价   数量
10001   薯片   20     48     50
10002   钢笔   40     80     100

```

4. 显示入库信息

```

-----入库管理菜单-----
(1)显示所有项
(2)删除条目
(3)修改条目
(0)退出
-----
请输入序号并按回车以选择：1

序号   货号   数量   入库时间
0       10001   50      19:56
1       10002   100     19:59

```

5. 修改入库信息

```

请输入序号并按回车以选择: 3
序号    货号    数量    入库时间
0       10001    50      19:56
1       10002    100     19:59
请输入要修改的条目序号: 1
请输入修改后的数量: 100
请输入修改后的时间(格式为 时钟+空格+分钟): 20 15
-----菜单-----

```

修改后:

```

请输入序号并按回车以选择: 1
序号    货号    数量    入库时间
0       10001    50      19:56
1       10002    100     20:15
-----菜单-----

```

6. 删除入库信息

```

请输入序号并按回车以选择: 2
序号    货号    数量    入库时间
0       10001    50      19:56
1       10002    100     20:15
请输入要删除的条目序号: 1
-----菜单-----

```

7. 出库

```

-----菜单-----
请输入序号并按回车以选择: 3
请输入货号: 10001
请输入出货量: 30
请输入出货时间(格式为 时钟+空格+分钟): 21 15
-----菜单-----

```

出库后: 薯片数量变为 20

```

-----菜单-----
请输入序号并按回车以选择: 1
货号    名称    进价    出价    数量
10001    薯片    20      48      20
10002    钢笔    40      80      100
-----菜单-----

```

如果出货量大于原货量, 则显示 error!

```

请输入序号并按回车以选择: 3
请输入货号: 10001
请输入出货量: 21
请输入出货时间(格式为 时钟+空格+分钟): 15 15
error!
-----菜单-----

```

如果全出, 则删除此货品信息

```

-----
请输入序号并按回车以选择：3

请输入货号：10001
请输入出货量：20
请输入出货时间（格式为 时钟+空格+分钟）：21 20
-----
菜单

```

```

-----
请输入序号并按回车以选择：1

货号  名称  进价  出价  数量
10002  钢笔  40    80    100
-----
菜单

```

8. 显示出库信息

```

-----
请输入序号并按回车以选择：1

序号  货号  数量  出库时间
0      10001  30     21:15
1      10001  21     15:15
2      10001  20     21:20
-----
菜单

```

9. 删除出库信息

```

-----出库管理菜单-----
(1)显示所有项
(2)删除条目
(3)修改条目
(0)退出
-----

请输入序号并按回车以选择：2

序号  货号  数量  出库时间
0      10001  30     21:15
1      10001  21     15:15
2      10001  20     21:20
请输入要删除的条目序号：1
-----

```

10. 修改出库信息

```

-----
请输入序号并按回车以选择：3

序号  货号  数量  出库时间
0      10001  30     21:15
1      10001  20     21:20
请输入要修改的条目序号：1
请输入修改后的数量：20
请输入修改后的时间(格式为 时钟+空格+分钟)：21 50
-----
菜单

```

```

-----
请输入序号并按回车以选择：1

序号  货号  数量  出库时间
0      10001  30     21:15
1      10001  20     21:50
-----
菜单

```

11. 附加功能：工人信息管理 添加工人


```

-----工人管理菜单-----
(1)显示所有项
(2)添加工人
(3)删除工人
(0)退出

请输入序号并按回车以选择： 2

请输入工号： 80001
请输入姓名： 张哥
菜单

```

12. 显示所有工人

```

-----
请输入序号并按回车以选择： 1

序号    工号    姓名
0       1111    张三
1       2222    李四
2       3333    王五
3       80001   张哥
菜单

```

13. 删除工人

```

-----
请输入序号并按回车以选择： 3

序号    工号    姓名
0       1111    张三
1       2222    李四
2       3333    王五
3       80001   张哥
请输入要删除的序号： 0
菜单

```

删除后：张三被删除

```

-----工人管理菜单-----
(1)显示所有项
(2)添加工人
(3)删除工人
(0)退出

请输入序号并按回车以选择： 1

序号    工号    姓名
0       2222    李四
1       3333    王五
2       80001   张哥
菜单

```

14. 查找货物

```
-----物资/库存管理菜单-----
(1)显示所有项
(2)进库
(3)出库
(4)修改进出价
(5)查找货物
(0)退出
-----
请输入序号并按回车以选择: 5
请输入要查找的货号: 10005
货号  名称  进价  出价  数量
10005  钢笔  56   99   15
-----菜单-----
```

如果找不到，则显示 error

```
(0)退出
-----
请输入序号并按回车以选择: 5
请输入要查找的货号: 10000
未查询到!
-----菜单-----
```

三、 注意事项

程序开始时没有初始化数据，各类数据需要自己根据需要逐个添加。