**C语言课程设计实习报告**



**学 院：**

**专 业：**

**班 级：**

**学 号：**

**姓 名：**

**指导老师：**

**目录**

1. **题目要求………………………………1**
2. **需求分析………………………………1**
3. **总体设计………………………………1**
4. **详细设计………………………………2**
5. **上机操作…………………………… 14**
6. **总 结…………………………… 21**
7. **题目要求**

**仓库管理系统**

该系统用于实现仓库管理，系统应支持入库、出库、库存查询以及打印。用户可自定库存警戒值，库存低于警戒值时以红字显示。仓库信息需要用文件存储。

1. **需求分析**

根据题目要求，由于货品信息存放在文件中，所以应该提供文件的读入、输出等操作；在程序中要浏览货品信息，应提供显示、查找、排序等操作；实现货物入库功能，要提供结构体的输入操作；实现统计功能，要提供相应的统计操作；实现修改功能，要提供修改操作；另外，还要提供键盘式菜单实现功能选择。

1. **总体设计**

根据上面的需求分析，可以将该系统的系统的设计分为大模块，即入库模块、出库模块、查询模块、排序模块、修改模块和统计模块等。

系统功能模块图

1. **详细设计**
2. **主函数**

结束

根据sele的值调用各功能模块函数

输入sele，判断sele

是否是0-7

开始

显示一系列功能选项

开始

显示一系列功能选项

N

输入sele，判断sele

是否是0-7

Y

根据sele的值调用各功能模块函数

结束

主函数流程图

【程序】

#include<stdio.h>

#include<windows.h>

#include<string.h>

#include<conio.h>

main()

{int sele;

Re\_file(); //读取货品信息

sele=1;

while(sele)

{system("cls");

printf("\n\n");

printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

printf("\* \*\n");

printf("\* 1.入库 2.出库 \*\n");

printf("\* \*\n");

printf("\* 3.查询 4.排序 \*\n");

printf("\* \*\n");

printf("\* 5.修改 6.统计 \*\n");

printf("\* \*\n");

printf("\* 7.退出 \*\n");

printf("\* \*\n");

printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

printf("请选择功能序号:");

scanf("%d",&sele);

switch(sele)

{ case 1:Stock\_in();Display();break;

case 2:Stock\_out();Display();break;

case 3:Query();break;

case 4:Sort();Display();break;

case 5:Modify();Display();break;

case 6:Statistics();break;

case 7:exit(0);sele=0;break;

}

printf("\n\n按任意键继续...\n");

getch();

}

Wr\_file();

}

1. **各功能模块设计**
2. 数据读入模块

【分析】单独看各组数据信息，货品名称是字符串，可以采用字符型数组；其他数据均可用整数存放。数据信息存放在文件中，一条记录对应一种货品的信息，既符合习惯也方便信息管理。把一种货品的相关信息作为结构体成员，如果要存放若干种货品的信息就要用结构体数组。

typedef struct

{

int num; //货品编号

char name[20]; //货品名称

int stock; //原始库存

int in; //入库数目

int out; //出库数目

int amount; //最终库存

int warning\_value; //警戒值

int state; //库存状态(是否低于警戒值)

}goods;

goods s[M]; //用于存放货品信息

goods r[M]; //用于存放入库货品信息

goods t[M]; //用于存放出库货品信息

【注意】s[M]、r[M]、t[M]的M为货品种类数，程序中采用宏定义的方式，可以随时在源程序宏定义中修改。本程序中宏定义为：#define M 50。

【程序】

void Re\_file() //读入原始库存文件

{

FILE\*fp;

N=0;

fp=fopen("goods.txt","r");

while(fscanf(fp,"%d%s%d%d%d%d%d",&s[N].num,&s[N].name,&s[N].stock,&s[N].in,&s[N].out,&s[N].amount,&s[N].warning\_value)!=EOF)N++;

fclose(fp);

P=N;

}

1. 入库模块

读入入库信息

找到相匹配的货品记录

将入库数目赋值

显示货品信息

入库模块流程图

【需求分析】该模块的功能是根据题目要求读入货品的入库数据。注意在主函数中已经读入了原始数据。

【程序】

void Stock\_in() //读入入库文件

{

FILE\*fp;

int i,j;

N=0;

fp=fopen("stockin.txt","r");

while(fscanf(fp,"%d%d",&r[N].num,&r[N].in)!=EOF)N++;

fclose(fp);

for(i=0;i<P;i++)

{for(j=0;j<N;j++)

{if(r[i].num==s[j].num)

s[j].in=r[j].in;

}

}

for(i=0;i<P;i++)

s[i].amount=s[i].stock+s[i].in;

}

1. 出库模块

【需求分析】该模块的功能是根据题目要求读入货品的出库数据。注意在主函数中已经读入了原始数据。

【程序】

void Stock\_out() //读入出库文件

{

FILE\*fp;

int i,j;

N=0;

fp=fopen("stockout.txt","r");

while(fscanf(fp,"%d%d",&t[N].num,&t[N].out)!=EOF)N++;

fclose(fp);

for(i=0;i<P;i++)

{for(j=0;j<N;j++)

{if(t[i].num==s[j].num)

s[j].out=t[j].out;

}

}

for(i=0;i<P;i++)

s[i].amount=s[i].stock+s[i].in-s[i].out;

}

1. 浏览模块

开始

判断货品库存总数是否小于警戒值

N

Y

白字输出

红字输出

结束

浏览模块流程图

【需求分析】该模块的功能是显示所有货品的记录信息。根据货品库存数目是否低于警戒值，分别按红字显示和白字显示。采用SetConsoleTextAttribute函数改变输出字体颜色。

【程序】

void Display() //显示库存情况

{

int i,j;

system("cls");

Estimate();

printf("货品编号 货品名称 原始库存 入库数目 出库数目 最终库存 警戒值\n");

for(i=0,j=1;i<P;i++,j++)

{

if(s[i].state==1) //库存值小于警戒值红字显示

{

HANDLE handle = GetStdHandle(STD\_OUTPUT\_HANDLE);

SetConsoleTextAttribute(handle, FOREGROUND\_INTENSITY |

FOREGROUND\_RED);

printf("%-9d %-10s%-10d%-10d%-10d%-10d%-10d\n",s[i].num,s[i].n

ame,s[i].stock,s[i].in,s[i].out,s[i].amount,s[i].warning

\_value);

if(j%10==0&&j!=P) //控制每次显示十行

{

printf("按任意键继续...");

getch();

puts("\n");

SetConsoleTextAttribute(handle, FOREGROUND\_INTENSITY |

FOREGROUND\_RED| FOREGROUND\_GREEN |

FOREGROUND\_BLUE);

printf("货品编号 货品名称 原始库存 入库数目 出库数目 最终

库存 警戒值\n");

}

}

else if(s[i].state==0) //库存值不小于警戒值白字显示

{

HANDLE handle = GetStdHandle(STD\_OUTPUT\_HANDLE);

SetConsoleTextAttribute(handle, FOREGROUND\_INTENSITY |

FOREGROUND\_RED| FOREGROUND\_GREEN |

FOREGROUND\_BLUE);

printf("%-9d %-10s%-10d%-10d%-10d%-10d%-10d\n",s[i].num,s[i].

name,s[i].stock,s[i].in,s[i].out,s[i].amount,s[i].warn

ing\_value);

if(j%10==0&&j<P)

{

HANDLE handle = GetStdHandle(STD\_OUTPUT\_HANDLE);

SetConsoleTextAttribute(handle, FOREGROUND\_INTENSITY |

FOREGROUND\_RED| FOREGROUND\_GREEN |

FOREGROUND\_BLUE);

printf("按任意键继续...");

getch();

puts("\n");

printf("货品编号 货品名称 原始库存 入库数目 出库数目 最

终库存 警戒值\n");

}

}

}

}

1. 判断模块

【需求分析】该模块用于判断货品库存数目是否小于警戒值并记录，多次在其他被模块调用。

【程序】

void Estimate() //判断库存值是否小于警戒值

{

int i;

for(i=0;i<P;i++)

{

if(s[i].amount>=s[i].warning\_value)

s[i].state=0;

else if(s[i].amount<s[i].warning\_value)

s[i].state=1;

}

}

1. 查询模块

输入货品信息

找到相应货品记录

输出货品信息

回主菜单

查询模块流程图

【需求分析】该模块的功能是根据输入的货品名称或货号查找对应的记录，找到以后，显示货品记录。同时，库存数目小于警戒值以红字显示。

【程序】

void Query() //查询货品

{

int k,m,i,j=-1;

char n[20];

system("cls");

printf("\n\n");

printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

printf("\* \*\n");

printf("\* \*\n");

printf("\* \*\n");

printf("\* 1.货品编号 2.货品名称 \*\n");

printf("\* \*\n");

printf("\* \*\n");

printf("\* \*\n");

printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

printf("\n\n请输入查询选项:");

scanf("%d",&k);

if(k==1)

{

printf("请输入货品编号:");

scanf("%d",&m);

for(i=0;i<P;i++)

{

if(m==i)

j=m-1;

}

}

else if(k=2)

{

printf("请输入货品名称:");

scanf("%s",&n);

for(i=0;i<P;i++)

{

if(strcmp(n,s[i].name)==0)

j=i;

}

}

Estimate();

if(j==-1)

{

printf("\n没有找到!\n");

Printf\_back();

}

else

{

if(s[j].state==1)

{

HANDLE handle = GetStdHandle(STD\_OUTPUT\_HANDLE);

SetConsoleTextAttribute(handle, FOREGROUND\_INTENSITY |

FOREGROUND\_RED);

printf("货品编号 货品名称 原始库存 入库数目 出库数目 最终

库存 警戒值\n");

printf("%-9d %-10s%-10d%-10d%-10d%-10d%-10d\n",s[j].num,s[j].n

ame,s[j].stock,s[j].in,s[j].out,s[j].amount,s[j].warnin

g\_value);

SetConsoleTextAttribute(handle, FOREGROUND\_INTENSITY |

FOREGROUND\_RED| FOREGROUND\_GREEN |

FOREGROUND\_BLUE);

Printf\_back();

}

else

{

HANDLE handle = GetStdHandle(STD\_OUTPUT\_HANDLE);

SetConsoleTextAttribute(handle, FOREGROUND\_INTENSITY |

FOREGROUND\_RED| FOREGROUND\_GREEN |

FOREGROUND\_BLUE);

printf("货品编号 货品名称 原始库存 入库数目 出库数目 最

终库存 警戒值\n");

printf("%-9d %-10s%-10d%-10d%-10d%-10d%-10d\n",s[j].num,s[j]

.name,s[j].stock,s[j].in,s[j].out,s[j].amount,s[j].war

ning\_value);

Printf\_back();

}

}

}

1. 排序模块

【需求分析】该模块的功能是将货品记录按照最终库存升序排列。这里按照冒泡法排序。排序完成后显示货品记录，显示完成后返回主菜单

【程序】

void Sort() //商品库存总量从小到大排序

{

int i,j;

goods t;

for(i=0;i<P-1;i++)

for(j=i+1;j<P;j++)

{

if(s[i].amount>s[j].amount)

{

t=s[i];

s[i]=s[j];

s[j]=t;

}

}

Display();

Printf\_back();

}

返回主菜单：

void Printf\_back() //返回主菜单

{

getch();

main();

}

1. 修改模块

输入货品信息

找到相应的货品记录

输入要修改的选项

修改相应数值

显示修改后的货品信息

修改模块流程图

【需求分析】该模块的功能是根据货品编号或货品名称找到要修改的记录，并提示用户修改该记录的哪部分信息（原始库存、入库数目、出库数目、警戒值），然后根据用户的选择修改相应的位置。最后显示货品信息供用户确认。

【程序】

void Modify() //修改货品信息

{

int k,m,i,j=-1,t,h;

char n[20];

system("cls");

printf("\n\n");

printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

printf("\* \*\n");

printf("\* \*\n");

printf("\* \*\n");

printf("\* 1.货品编号 2.货品名称 \*\n");

printf("\* \*\n");

printf("\* \*\n");

printf("\* \*\n");

printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

printf("\n\n请输入修改选项:");

scanf("%d",&k);

if(k==1)

{

printf("请输入货品编号:");

scanf("%d",&m);

for(i=0;i<P;i++)

{

if(m==i)

j=m-1;

}

}

else if(k==2)

{

printf("请输入货品名称:");

scanf("%s",&n);

for(i=0;i<P;i++)

{

if(strcmp(n,s[i].name)==0)

j=i;

}

}

if(j==-1)

{

printf("\n没有找到!\n");

Printf\_back();

}

else

{

printf("\n\n");

printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

printf("\* \*\n");

printf("\* 1.原始库存 2.入库数目 \*\n");

printf("\* \*\n");

printf("\* 3.出库数目 4.警戒值 \*\n");

printf("\* \*\n");

printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

printf("请选择修改选项:");

scanf("%d",&t);

if(t==1)

{

scanf("%d",&h);

s[j].stock=h;

}

else if(t==2)

{

scanf("%d",&h);

s[j].in=h;

}

else if(t==3)

{

scanf("%d",&h);

s[j].out=h;

}

else if(t==4)

{

scanf("%d",&h);

s[j].warning\_value=h;

}

s[j].amount=s[j].stock+s[j].in-s[j].out;

}

printf("请确认核对货品信息:\n");

Display();

}

1. 统计模块

选择统计功能

统计相应数据

输出相应数据

将货品信息存入文件

统计模块流程图

【需求分析】该模块的功能是根据用户的选择统计货品总数或最终库存低于警戒值的货品种数。统计结束后将所有货品信息以文件形式输出。

【程序】

void Statistics() //统计数目

{

int k,i,j,m=0,n=0;

system("cls");

printf("\n\n");

printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

printf("\* \*\n");

printf("\* \*\n");

printf("\* \*\n");

printf("\* 1.库存总数 2.库存状态 \*\n");

printf("\* \*\n");

printf("\* \*\n");

printf("\* \*\n");

printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

printf("\n\n请输入查询选项:");

scanf("%d",&k);

if(k==1)

{

for(i=0;i<P;i++)

m=m+s[i].amount;

printf("库存总数为：%d",m);

}

else if(k==2)

{

Estimate();

for(j=0;j<P;j++)

if(s[j].state==1)

n=n+1;

printf("低于库存警戒值的商品种数为:%d",n);

}

Wr\_file();

}

1. 数据输出模块

【需求分析】该模块的功能是将货品信息写入文件。

【程序】

int Wr\_file() //写输出文件

{

FILE\*fp;

int i;

fp=fopen("amount.txt","w");

for(i=0;i<P;i++)

fprintf(fp,"%-9d %-10s%-10d%-10d%-10d%-10d%-10d\n",s[i].num,s

[i].name,s[i].stock,s[i].in,s[i].out,s[i].amount,s[i].

warning\_value);

fclose(fp);

return 1;

}

1. **上机操作**
2. **数据源**

该程序的运行需要四个文件，三个用于读取货品信息，一个用于存储货品

息。

文件的名称和格式如下：

goods.txt

货品编号 货品名称 原始库存 入库数目 出库数目 最终库存 警戒值

1 A1 456 0 0 456 400

2 A2 23 0 0 23 100

3 A3 67 0 0 67 100

4 A4 104 0 0 104 100

5 A5 78 0 0 78 100

… … … … … … …

用于读取货品编号、货品名称、原始库存、警戒值等货品信息。

stockin.txt

货品编号 入库数目

1 57

3 78

4 80

5 23

6 10

… …

用于读取入库信息。

stockout.txt

货品信息 出库数目

1 10

2 5

3 23

4 12

5 3

… …

用于读取出库信息。

amount.txt

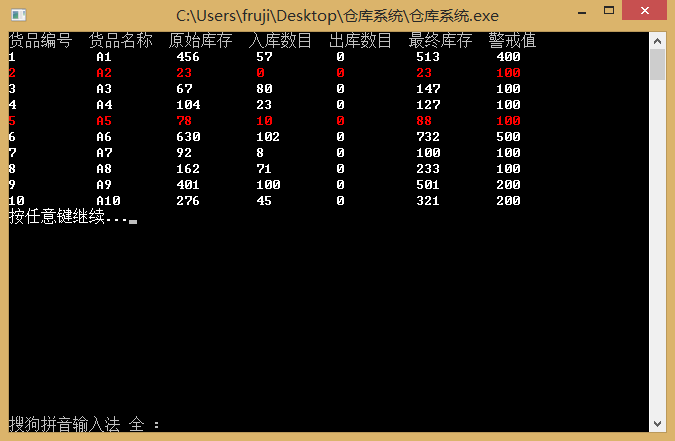
用于存储货品信息。

1. **编译、链接和运行**
2. **结果**
3. 主函数界面

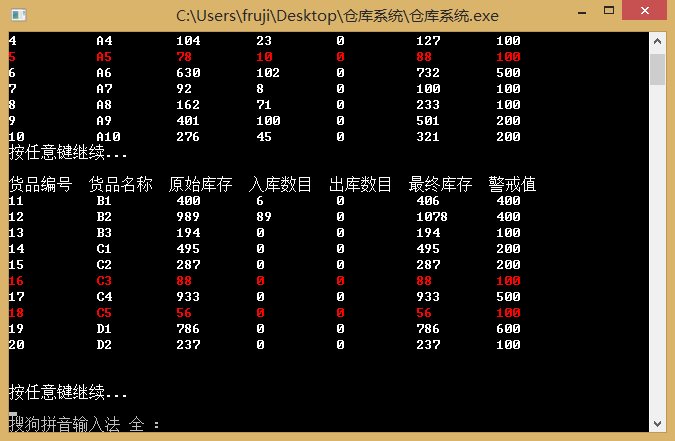


1. 入库模块

在主界面中输入“1”，然后按“Enter”键，进入入库模块。

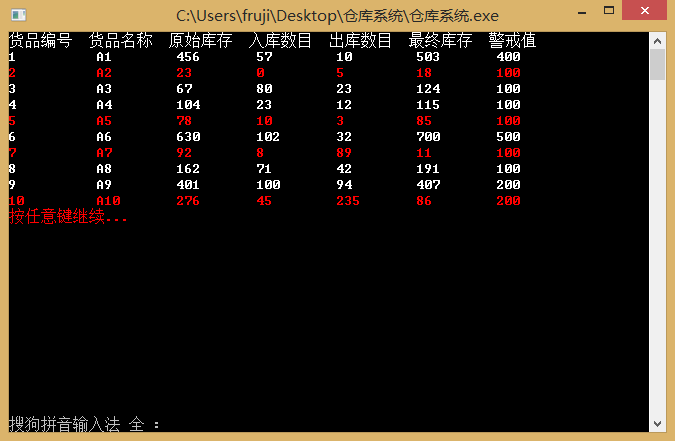


按任意键显示后十条记录:

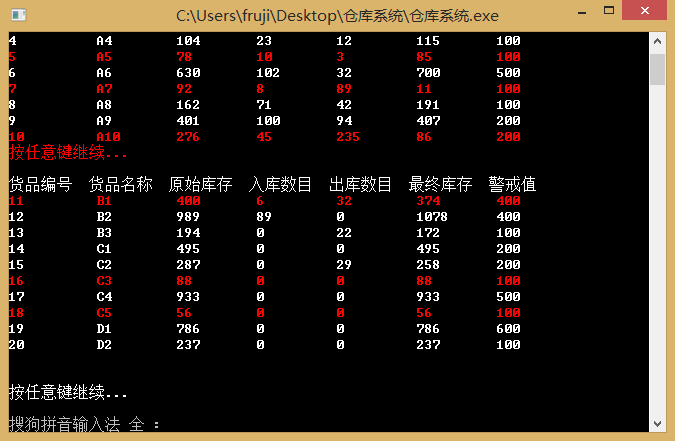


1. 出库模块

在主界面中输入“2”，然后按“Enter”键，进入出库模块。



按任意键显示后十条记录:



1. 查询模块

在主界面中输入“3”，然后按“Enter”键，进入查询模块。



1. 排序模块

在主界面中输入“4”，然后按“Enter”键，进入排序模块。可实现最终库存按升序排列。



按任意键显示后十条记录:



1. 修改模块

在主界面中输入“5”，然后按“Enter”键，进入修改模块。



1. 统计模块

在主界面中输入“6”，然后按“Enter”键，进入统计模块。



键入“1”查询库存总数：



键入“2”查询库存状态：



1. **总结**

通过这次课程设计，我进一步理解和运用了结构化程序设计的思想方法，初步掌握了开发小型实用系统的基本方法，学会了调试较长的程序，能够利用流程图描述算法，进一步掌握和提高利用C语言进行程序设计的能力。

在程序的编写中，我们首先要构架好整体结构，然后再将各个模块的函数写出来。在写函数的过程中，我们要选择性地利用到所学的知识，如排序法、while循环、if语句等，实现相应的功能。一个较大的程序主要调试方法有分模块调试和分步调试。文件的数据格式很重要，要与程序读取的数据类型保持一致。程序调试通了以后，在修改其中的细节问题。

学习好C语言这门课，能为我们以后开发出更多实用的程序打下坚实的基础。要学好C语言这门课，我们平时就要勤于实践，锻炼自己的逻辑思维能力与整体构架能力。