

高级语言程序设计

课程设计报告

|  |  |  |
| --- | --- | --- |
| 课程设计成绩（总分20） | | 分数 |
| 程序 | 程序设计完整，功能齐全（8-10分） |  |
| 程序设计基本完整，功能基本完成（0-7分） |
| 报告 | 结构清晰完整，报告格式正确，语言通顺（8-10分） |  |
| 格式不正确，封面或字体有误，语言欠通顺（0-7分） |
|  | 总分 |  |

|  |  |
| --- | --- |
| 课程名称 | 高级语言程序设计 |
| 课程编号 | 08304135 |
| 报告题目 | 宠物寄养管理系统 |
| 序 号 | 7 |
| 姓 名 | 吕斯路 |
| 学 号 | 21122787 |
| 所在学院 | 计算机工程与科学学院 |
| 专业名称 | 计算机科学与技术 |

2021年 11 月

目 录

[1. 数据格式与文件描述 3](#_Toc514061021)

[1.1 数据文件格式 3](#_Toc514061022)

[1.2 功能描述 3](#_Toc514061023)

[1.3 要求 4](#_Toc514061024)

[2. 问题分析 5](#_Toc514061025)

[3. 总体设计 5](#_Toc514061026)

[3.1 功能模块设计 5](#_Toc514061027)

[3.2 系统界面设计 6](#_Toc514061028)

[3.3 数据结构设计 6](#_Toc514061029)

[3.4 函数设计 7](#_Toc514061030)

[4. 详细设计 8](#_Toc514061031)

[4.1 各个函数的调用关系图 8](#_Toc514061032)

[4.2 各个函数的功能描述 8](#_Toc514061033)

[5. 功能测试 26](#_Toc514061034)

[5.1 初始化界面 26](#_Toc514061035)

[5.2 查询宠物信息 26](#_Toc514061036)

[5.3 删除宠物信息 28](#_Toc514061037)

[5.4 修改宠物信息 29](#_Toc514061038)

[5.5 增加宠物信息 31](#_Toc514061039)

[5.6 查询主人信息 32](#_Toc514061040)

[5.7 主人消费排序 35](#_Toc514061041)

[5.8 退出系统 37](#_Toc514061042)

[6. 总 结 37](#_Toc514061046)

宠物寄养管理系统

宠物寄养管理系统旨在给广大忙碌的上班族提供一个寄养宠物的平台，主要针对宠物店中宠物信息的增删改查，宠物主人信息的管理。

本实验要求设计一个简单的家宠物寄养管理系统，涉及函数、结构体、链表、文件等方面的知识，学习利用链表处理数据的方法，熟练掌握文件操作，构建综合程序设计的思路及框架，提高综合设计软件系统的能力。

1. 数据格式与文件描述
   1. 数据文件格式

数据文件分为2个，宠物店中店宠物信息与宠物主人的登记信息。宠物信息文件包括了此宠物的编号、名字、品种、主人编号，寄养天数，每天价格。格式如图1-1所示。主人文件包括主人编号、主人姓名、联系方式、寄养宠物的数量，具体格式如图1-2所示。

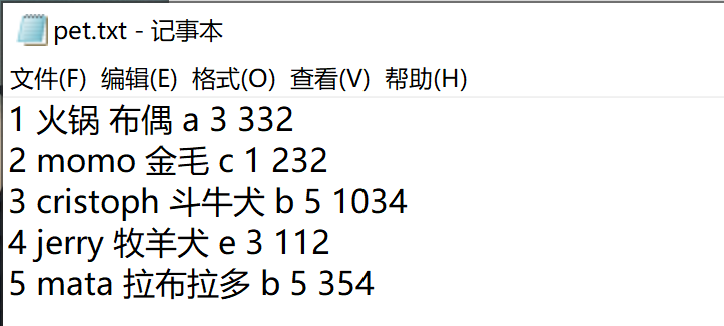


图1-1 宠物信息

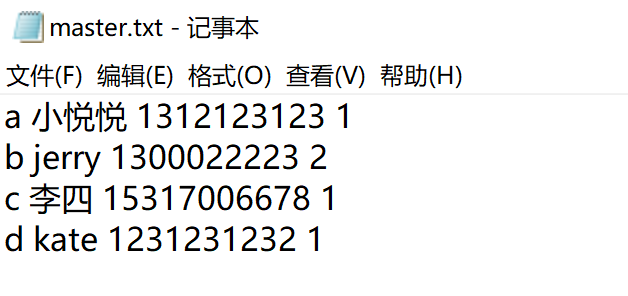


图1-1 主人信息

* 1. 功能描述

设计一个宠物寄养管理系统，利用单链表来处理宠物信息与主人 信息，要求实现如下系统功能。

1. 初始化菜单

程序开始运行后显示菜单，要求菜单能完全展示本系统的功能，尽力做到设计美观。

(2)选择待操作的文件

考虑到本系统有多个文件，需要提示用户输入来选择需要操作的文件是宠物信息还是主人信息。

(3)查询宠物信息与主人信息

选择要查询的宠物编号或者名字，输出此宠物的个狗/猫信息与主人信息。

可输出此全部宠物信息，若当前宠物店没有宠物，输出提示信息。

可输出全部主人信息，若当前没有没有主人，输出提示信息。

(4) 删除宠物信息

按指定的宠物编号、名字删除宠物信息中对应记录，删除时应删除此宠物的所有信息，此外如果此宠物的主人没有其他宠物寄养在本宠物店，删除此主人信息并提示删除成功。若此宠物不存在，输出提示信息。

(5) 修改宠物信息

修改某宠物的信息。输入待修改宠物的编号，若该宠物编号不存在，输出提示信息。若宠物存在，提示修改宠物寄养天数或者宠物寄养价格，

(6) 增加宠物信息与主人信息

通过控制台增加宠物信息，提示用户输入待增加的宠物编号，如果编号存在，则输出提示信息；如果不存在，新增宠物信息。如果主人文件中没有此宠物主人信息，逐项录入主人信息，否则输出提示，主人信息已经存在。

(7) 花费排序

按照主人的花销，从高到低输出主人的信息、拥有宠物名称及品种、总花销。

(8) 保存宠物信息与主人信息

将当前宠物信息与主人信息分别保存到txt文件中。

(9)退出系统

用户可通过输入特定指令来退出程序，在用户输入退出指令之前，程序不得自行退出。

* 1. 要求

(1) 源程序编写要求

根据系统功能描述，采用模块化程序设计方法进行程序设计，要求程序结构清晰。上述各个功能模块要求分别用函数实现，在主函数中通过调用这些函数，完成系统功能的要求。代码书写规范，有简要的注释，给出数据和函数说明。

(2) 设计报告撰写要求

设计报告内容包括题目内容和要求、总体设计、详细设计、源代码、调试过程中的问题、总结等。

总体设计：对程序的整体设计思路进行描述，画出图书管理系统的总体功能模块图，说明系统使用的主要数据结构，给出包含上述功能的系统功能模块图。

详细设计：分析实现各函数功能的算法，描述函数的功能。

调试过程中的问题：记录程序编写和调试过程中遇到的各种问题，以及解决这些问题的途径和方法。

总结：回顾整个综合程序设计的过程，对学习到的设计方法和思路进行总结，写出个人体会。

1. 问题分析

宠物和主人信息的数据为文本格式，由多个数据项组成。在程序设计中，可以用结构体来存储，由于系统功能涉及宠物和主人信息插入和删除，使用链表结构存储信息更有利于插入和删除处理。因此，本实验将宠物和主人信息分别存储在两个单链表的结点中，由数据域和指针域构成。宠物链表数据部分包括：编号、名字、品种、主人编号，寄养天数，每天价格共六个成员；主人链表数据部分包括：编号、主人姓名、联系方式、寄养宠物的数量。指针域存储其直接后继结点的地址。

宠物和主人信息的查找、删除、修改、插入都需要先查找，找到目标结点后再进行相应操作。对主人消费的排序，可采用选择排序、冒泡排序、插入排序等，针对单链表结构的特点，宠物寄养管理系统可采用选择排序的方法实现按不同消费的排序。保存信息后，修改后的信息被写入数据文件。

1. 总体设计
   1. 功能模块设计

根据系统功能描述和问题分析，可将系统功能划分为若干模块，如图3-1所示。

图3-1 系统功能模块图

* 1. 系统界面设计

进入宠物寄养管理系统后，首先显示系统主菜单，菜单列出3个选项：宠物信息、主人信息和退出系统。选项编号为1～2，0为退出程序。

宠物信息菜单有5个选项：查询信息、删除信息、修改信息、增加信息和返回。选项编号为1～4，0为返回主菜单。

主人信息菜单有3个选项：查询信息、主任消费排序和返回。选项编号为1～2，0为返回主菜单。

用户根据主菜单中显示的功能模块及其相应编号，选择相应编号，执行相应的功能。所选模块执行完毕后，返回系统主菜单。

* 1. 数据结构设计

实验内容涉及到多条记录的处理，每条记录含有不同类型的数据项，使用单链表来存储较为方便，链表结点设计如下：

struct master

{

char mnum[10];

char mname[200];

int phone;

int sum;

master \*next;

float expense;

};

struct pet

{

int pnum;

string pname;

string type;

char owner[10];

int time;

float dayprice;

pet \*next;

};

单链表结构中，每个节点的next作为指针域，用于存储其后续节点的地址，其他部分为数据域，存储结点中的数据。

* 1. 函数设计

函数设计的主要内容包括函数的名称，函数的功能与函数中参数类型的说明。

|  |  |  |
| --- | --- | --- |
| 函数名称 | 函数功能 | 参数说明 |
| pet \*CreatePetList() | 创建宠物链表 |  |
| master \*CreateHostList() | 创建宠物链表 |  |
| pet \*idGetPet(pet \*&Pet, int id) | 用编号获取宠物链表结点 | Pet表示宠物链表头指针，  id表示宠物编号 |
| pet \*nameGetPet(pet \*&Pet, string name) | 用名字获取宠物链表结点 | Pet表示宠物链表头指针，  name表示宠物名字 |
| master \*idGetMaster(master \*&Master, char id[]); | 用编号获取主人链表结点 | Master表示主人链表头指针，id表示主人编号 |
| Master \*nameGetMaster(master \*&Master, char name[]); | 用名字获取主人链表结点 | Master表示主人链表头指针，name表示主人名字 |
| void SaveFile(pet \*&Pet) | 保存数据至宠物文件 | Pet表示宠物链表头指针 |
| void SaveFile(master \*&Master) | 保存数据至主人文件 | Master表示主人链表头指针 |
| void Search() | 查询宠物信息 |  |
| void pSearch() | 查询单个宠物信息 |  |
| void mSearch() | 查询单个客户信息 |  |
| void Search2() | 查询客户信息 |  |
| void MainMenu() | 初始化界面 |  |
| bool idDelData(int id) | 依据宠物编号删除信息 | id表示宠物编号 |
| bool nameDelData( string name); | 依据名字删除信息 | name表示名字 |
| void Del() | 删除操作 |  |
| bool idDelData( char num[]); | 依据主人编号删除信息 | num表示主人编号 |
| void Change() | 更改数据 |  |
| void Add() | 增加数据 |  |
| void ToSort() | 消费选择排序 |  |
| void Sort(); | 排序模块 |  |

1. 详细设计

详细设计包括的源代码的框架及其功能描述。

* 1. 各个函数的调用关系图

如图4-1所示。

图4-1

* 1. 各个函数的功能描述

4.2.1主函数及初始化界面

int main()

{

Master = CreateHostList();

Pet = CreatePetList();

int choice = -1;

while(1)

{

system("cls");

MainMenu();

cin >> choice;

if( choice == 1 )

{

PetMenu();

cin >> choice;

switch(choice)

{

case 1:

Search();

break;

case 2:

Del();

break;

case 3:

Change();

break;

case 4:

Add();

break;

case 0:

continue;

}

}

else if( choice == 2 )

{

MasterMenu();

cin >> choice;

switch(choice)

{

case 1:

Search2();

break;

case 2:

Sort();

break;

case 0:

continue;

}

}

else{

break;

}

}

return 0;

}

//打印菜单

void MainMenu(){

SetConsoleTextAttribute(GetStdHandle(STD\_OUTPUT\_HANDLE), FOREGROUND\_INTENSITY | FOREGROUND\_BLUE|FOREGROUND\_GREEN);

cout<<" "<<endl;

cout<<" : ..r "<<endl;

cout<<" rb1JJr. rBQ SB: DQQi BB vBE .iuX5jvr "<<endl;

cout<<" .. .iLsXZJ::. BB Q is. .BQQP rJ b2 EB iLr.:ZSL7::irU: "<<endl;

cout<<" :PLjri::rY :JSr :. ..iBPvi:5PDBBDqBBBBBP.EM.LLY:7B.JQQBBBgqQBBbKL:72B2 . .i .d7r. i:rrvrv. . "<<endl;

cout<<" r.r.rrii11 . i: Z. .dZBdJBBgQg2BBBgBE M: .r2BBM M.7BBEv: R. 7BBgBBPqBgBBKSQMPs S :. r IIriii.P77 "<<endl;

cout<<" LPr ...:: i :vv j7 . Bq 2BB. YJQP Q5 B:g MBg Bv . .2. iq7 . Qv:."<<endl;

cout<<" ir7g LIIUiiiB Br .i rBvPrBD .: Q5 UMvir2Xj .. 7qi7i"<<endl;

cout<<".rviYSi.....:r7i7ivs.: 7 . .B Qi vQq Bb PZ i. qL:ii::rrvvv77i.:i."<<endl;

cout<<" .irrr:. ..:rv . rDJri7d . B7 7S:ivKE . "<<endl;

SetConsoleTextAttribute(GetStdHandle(STD\_OUTPUT\_HANDLE),FOREGROUND\_INTENSITY|FOREGROUND\_RED|FOREGROUND\_GREEN|FOREGROUND\_BLUE);

cout<<" ........ :r "<<endl;

cout<<" .YYL7ririrrvL7: Q:2: "<<endl;

cout<<" v5KI.. .:ru:. rZ. u1 "<<endl;

cout<<" i5ur5 . . . . . SSX: qiL :P: "<<endl;

cout<<" .q1ri75 ............ I7rSL qi5P vU1Jrirrri7Q. "<<endl;

cout<<" XuiiiJv ... ... 2riij2: PrEr ... .qE "<<endl;

cout<<" Luii:5.:Ud7 . ... . : sriii7b5 i7KS 7: Us5q "<<endl;

cout<<" .SiirI EbQS .. ZDP5Y:ii7X. 7g7: ZBBQ 2KY "<<endl;

cout<<" 57iU: :. ..: :5I:Jiii1i :Y :.7BBL X. BB7J "<<endl;

cout<<" IK5K :: . KBBBB. : 71iiK d 7. BMPZv "<<endl;

cout<<" ii1 5PPY . i:YI1SS :X 5 "<<endl;

cout<<" ivr.....:. . d2. :i. Z. .P "<<endl;

cout<<" 1r.71i. .i21:ib d b. "<<endl;

cout<<" Yr :s ..:.... 1: .q Xv "<<endl;

cout<<" .K rr. . iY 7L i2s. "<<endl;

cout<<" iv :i.. .... .U Li .Uj "<<endl;

cout<<" :r : .. ....... .j J: :b: "<<endl;

cout<<" Ks j: .... . Y 7r 5 Z. 77LY: "<<endl;

cout<<" si1 s. .... .X J :2 .2 :. .E :Z: .PM "<<endl;

cout<<" L..u u . :K .g d B7 rs P g5dJ XY "<<endl;

cout<<" .j S. 7v . 2 XP E Xru .: X q :b q "<<endl;

cout<<" j rr .1 . .v vq. Ji q b UJ v g7 :P. K: "<<endl;

cout<<" dX.Ur .Z:::.I: SJK :5 .2 q DLJLI .gv.rYvvv1u7 .5r "<<endl;

cout<<" iiiiIY2: 7Sr7P7Y E:P: D.Si rg. ..ruj .7i.... .:uJ. "<<endl;

cout<<" . . ..... ...... "<<endl;

SetConsoleTextAttribute(GetStdHandle(STD\_OUTPUT\_HANDLE), FOREGROUND\_INTENSITY | FOREGROUND\_BLUE|FOREGROUND\_GREEN);

cout << " " << endl;

cout << " \*##\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*";

SetConsoleTextAttribute(GetStdHandle(STD\_OUTPUT\_HANDLE),FOREGROUND\_INTENSITY|FOREGROUND\_RED|FOREGROUND\_GREEN|FOREGROUND\_BLUE);

cout << "【宠物寄养管理系统】";

SetConsoleTextAttribute(GetStdHandle(STD\_OUTPUT\_HANDLE), FOREGROUND\_INTENSITY | FOREGROUND\_BLUE|FOREGROUND\_GREEN);

cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*##\*" << endl;

cout << " \* \*" << endl;

SetConsoleTextAttribute(GetStdHandle(STD\_OUTPUT\_HANDLE), FOREGROUND\_INTENSITY | FOREGROUND\_RED|FOREGROUND\_GREEN);

cout << " \* 请选择要处理的信息： \*"<< endl;

cout << " \* [1] 宠物信息 \*" << endl;

cout << " \* [2] 主人信息 \*" << endl;

cout << " \* [0] 退出系统 \*" << endl;

SetConsoleTextAttribute(GetStdHandle(STD\_OUTPUT\_HANDLE), FOREGROUND\_INTENSITY | FOREGROUND\_BLUE|FOREGROUND\_GREEN);

cout << " \* \*" << endl;

cout << " \*##\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*##\*" << endl;

}

//打印菜单

void MasterMenu(){

SetConsoleTextAttribute(GetStdHandle(STD\_OUTPUT\_HANDLE),FOREGROUND\_INTENSITY|FOREGROUND\_RED|FOREGROUND\_GREEN|FOREGROUND\_BLUE);

cout << " [1] 查询信息"<<endl;

cout << " [2] 主人消费排序" << endl;

cout << " [0] 返回" << endl<<endl;

cout << " 请选择0~2:" << endl;

}

//打印菜单

void PetMenu(){

SetConsoleTextAttribute(GetStdHandle(STD\_OUTPUT\_HANDLE),FOREGROUND\_INTENSITY|FOREGROUND\_RED|FOREGROUND\_GREEN|FOREGROUND\_BLUE);

cout << " [1] 查询信息" << endl;

cout << " [2] 删除信息" << endl;

cout << " [3] 修改信息" << endl;

cout << " [4] 增加信息" << endl;

cout << " [0] 返回" << endl << endl;

cout << " 请选择0~4:" << endl;

}

4.2.2 具体功能实现

//把文件数据读入链表

pet \*CreatePetList()

{

int n;

pet \*head = NULL, \*p, \*q;

ifstream f1;

f1.open("pet.txt", ios::in);

f1 >> n;

if(n == 0)

{

cout<<"文件为空！"<<endl;

return NULL;

}

else{

p = new pet;

p -> pnum = n;

f1 >> p -> pname;

f1 >> p -> type;

f1 >> p -> owner;

f1 >> p -> time;

f1 >> p -> dayprice;

if(head == NULL) head = p;

else {

q -> next = p;

}

q = p;

}

while(!f1.eof())

{

p = new pet;

f1 >> p -> pnum;

f1 >> p -> pname;

f1 >> p -> type;

f1 >> p -> owner;

f1 >> p -> time;

f1 >> p -> dayprice;

if(head == NULL) head = p;

else {

q -> next = p;

}

q = p;

}

if(head != NULL) q->next = NULL;

f1.close();

return head;

}

//把文件数据读入链表

master \*CreateHostList()

{

master \*head = NULL, \*p, \*q;

FILE \*fp1;

char ch, a[10];

fp1=fopen("master.txt","r");

if(fp1 == NULL)

{

cout<<"打开文件失败！"<<endl;

return NULL ;

}

ch = fgetc(fp1);

if(ch == EOF){

cout << "文件为空！ ";

return NULL;

}

else{

p = new master;

a[0] = ch;

strcpy(p->mnum, a);

fscanf(fp1,"%s %d %d", p->mname, &p->phone, &p->sum);

p->expense = 0;

if(head == NULL)

{head = p;q=p;}

else {

q->next = p;

q = p;

}

}

while(!feof(fp1))

{

p = new master;

fscanf(fp1,"%s %s %d %d",p->mnum,p->mname,&p->phone,&p->sum);

p->expense = 0;

if(head == NULL)

{head = p;q=p;}

else {

q->next = p;

q = p;

}

}

if(head != NULL) q -> next = NULL;

fclose(fp1);

return head;

}

//查询单个宠物信息

void pSearch(){

system("cls");

cout << "[1] 编号" << endl;

cout << "[2] 名字" << endl;

cout << "[0] 返回" << endl;

cout << "\*\*\*\*\*\*请选择0~2：" << endl;

int choice;

cin >> choice;

if(choice == 1){

cout << "请输入编号：" << endl;

int num;

cin >> num;

pet \*k= idGetPet(Pet, num);

if(k == NULL) cout << "请检查" << num <<"是否存在！";

else{

cout << "编号：" << k->pnum << endl;

cout << "名字：" << k->pname << endl;

cout << "品种：" << k->type << endl;

cout << "主人编号：" << k->owner << endl;

cout << "寄养天数" << k->time << endl;

cout << "每日价格" << k->dayprice << endl << endl;

master \*k2 = idGetMaster(Master, k->owner);

if(k2 !=NULL){

cout << "主人信息:" <<endl;

cout << " 主人编号：" << k2->mnum <<endl;

cout << " 主人名字：" << k2->mname << endl;

cout << " 联系方式：" << k2->phone << endl;

cout << " 寄养宠物数量：" << k2->sum << endl << endl;

}

else{

cout << "缺少主人信息！" << endl;

}

}

system("pause");

}

else if(choice == 2){

cout << "请输入名字：" << endl;

string name;

cin >> name;

pet \*k = NULL;

k = nameGetPet(Pet, name);

if(k == NULL) cout << "请检查" << name <<"是否存在！";

else{

cout << "编号：" << k->pnum << endl;

cout << "名字：" << k->pname << endl;

cout << "品种：" << k->type << endl;

cout << "主人编号：" << k->owner << endl;

cout << "寄养天数:" << k->time << endl;

cout << "每日价格:" << k->dayprice << endl << endl;

master \*k2 = idGetMaster(Master, k->owner);

cout << "主人信息:" <<endl;

cout << " 主人编号：" << k2->mnum <<endl;

cout << " 主人名字：" << k2->mname << endl;

cout << " 联系方式：" << k2->phone << endl;

cout << " 寄养宠物数量：" << k2->sum << endl << endl;

}

system("pause");

}

else{

return;

}

}

//查询宠物信息

void Search(){

system("cls");

if(Pet == NULL){

cout << "没有宠物信息！" << endl;

system("pause");

return;

}

cout << "[1] 查询单个宠物信息" << endl;

cout << "[2] 查询所有宠物信息" << endl;

cout << "[0] 返回" << endl;

cout << "\*\*\*\*\*\*请选择0~2：" << endl;

int choice;

cin >> choice;

if(choice == 1){

pSearch();

}

else if(choice == 2){

pet \*p = Pet;

while(p != NULL){

cout << "编号：" << p->pnum << endl;

cout << "名字：" << p->pname << endl;

cout << "品种：" << p->type << endl;

cout << "主人编号：" << p->owner << endl;

cout << "寄养天数:" << p->time << endl;

cout << "每日价格:" << p->dayprice << endl << endl;

master \*k2 = idGetMaster(Master, p->owner);

if(k2 != NULL){

cout << "主人信息:" <<endl;

cout << " 主人编号：" << k2->mnum <<endl;

cout << " 主人名字：" << k2->mname << endl;

cout << " 联系方式：" << k2->phone << endl;

cout << " 寄养宠物数量：" << k2->sum << endl << endl;

}

else{

cout << "缺少主人信息！" << endl << endl;

}

p = p->next;

}

system("pause");

}

else{

return;

}

}

// 查询单个客户信息

void mSearch(){

system("cls");

cout << "[1] 编号" << endl;

cout << "[2] 名字" << endl;

cout << "[0] 返回" << endl;

cout << "\*\*\*\*\*\*请选择0~2：" << endl;

int choice;

cin >> choice;

if(choice == 1){

cout << "请输入编号：" << endl;

char num[10];

cin >> num;

master \*k = NULL;

k = idGetMaster(Master, num);

if(k == NULL) cout << "请检查" << num <<"是否存在！";

else{

cout << "编号：" << k->mnum << endl;

cout << "名字：" << k->mname << endl;

cout << "联系方式:" << k->phone << endl;

cout << "寄养宠物数量:" << k->sum << endl;

}

system("pause");

}

else if(choice == 2){

cout << "请输入名字：" << endl;

char name[100];

cin >> name;

master \*k = NULL;

k = nameGetMaster(Master, name);

if(k == NULL) cout << "请检查" << name <<"是否存在！";

else{

cout << "编号：" << k->mnum << endl;

cout << "名字：" << k->mname << endl;

cout << "联系方式:" << k->phone << endl;

cout << "寄养宠物数量:" << k->sum << endl;

}

system("pause");

}

else{

return;

}

}

//查询客户信息

void Search2(){

system("cls");

cout << "[1] 查询单个客户信息" << endl;

cout << "[2] 查询所有客户信息" << endl;

cout << "[0] 返回" << endl;

cout << "\*\*\*\*\*\*请选择0~2：" << endl;

int choice;

cin >> choice;

if(choice == 1){

mSearch();

}

else if(choice == 2){

master \*p = Master;

if(p == NULL){

cout << "没有客户信息！" << endl;

system("pause");

}

while(p != NULL){

cout << "编号：" << p->mnum << endl;

cout << "名字：" << p->mname << endl;

cout << "联系方式:" << p->phone << endl;

cout << "寄养宠物数量:" << p->sum << endl;

p = p->next;

}

system("pause");

}

else{

return;

}

}

//删除处理

void Del()

{

system("cls");

cout << "[1] 按编号删除" << endl;

cout << "[2] 按名字删除" << endl;

cout << "[0] 返回" << endl;

cout << "\*\*\*\*\*\*请选择0~2：" << endl;

int choice;

cin >> choice;

while(true){

if(choice == 1){

cout << "请输入编号：" << endl;

int num;

cin >> num;

pet \*k = idGetPet(Pet, num);

if(k == NULL){

cout << "请检查" << num <<"是否存在！" << endl;

system("pause");

continue;

}

else{

cout << "编号：" << k->pnum << endl;

cout << "名字：" << k->pname << endl;

cout << "品种：" << k->type << endl;

cout << "主人编号：" << k->owner << endl;

cout << "寄养天数:" << k->time << endl;

cout << "每日价格:" << k->dayprice << endl;

char mnum[10];

strcpy(mnum,k->owner);

master \*t = idGetMaster(Master, mnum);

if(t->sum == 1){

cout << "已近找到编号为" << num << "的数据，是否删除全部数据和其主人信息？" << endl;

if(MessageBox(NULL,"【确定要进行删除操作吗？】","提示",MB\_OKCANCEL) == 1){

if(idDelData(num)&&idDelData(mnum)){

SaveFile(Pet);

SaveFile(Master);

cout << "操作成功！"<<endl;

system("pause");

break;

}

else{

cout<<"操作失败！"<<endl;

system("pause");

continue;

}

}

else{

return;

}

}

else

{

cout << "已近找到编号为" << num << "的数据，是否全部删除？" << endl;

if(MessageBox(NULL,"【确定要进行删除操作吗？】","提示",MB\_OKCANCEL) == 1){

if(idDelData(num)){

t->sum--;

SaveFile(Master);

SaveFile(Pet);

cout << "操作成功！"<<endl;

system("pause");

break;

}

else{

cout<<"操作失败！"<<endl;

system("pause");

continue;

}

}

else{

return;

}

}

system("pause");

}

}

else if(choice == 2){

cout << "请输入名字：" << endl;

string name;

cin >> name;

pet \*k = NULL;

k = nameGetPet(Pet, name);

if(k == NULL) cout << "请检查" << name <<"是否存在！" << endl;

else{

cout << "编号：" << k->pnum << endl;

cout << "名字：" << k->pname << endl;

cout << "品种：" << k->type << endl;

cout << "主人编号：" << k->owner << endl;

cout << "寄养天数:" << k->time << endl;

cout << "每日价格:" << k->dayprice << endl;

char mnum[10];

strcpy(mnum,k->owner);

master \*t = idGetMaster(Master, mnum);

if(t->sum == 1){

cout << "已近找到名字为" << name << "的数据，是否删除全部数据和其主人信息？" << endl;

if(MessageBox(NULL,"【确定要进行删除操作吗？】","提示",MB\_OKCANCEL) == 1){

if(nameDelData(name)&&idDelData(mnum)){

SaveFile(Pet);

SaveFile(Master);

cout << "操作成功！"<<endl;

system("pause");

break;

}

else{

cout<<"操作失败！"<<endl;

system("pause");

continue;

}

}

else{

return;

}

}

else{

cout << "已近找到名字为" << name << "的数据，是否全部删除？" << endl;

if(MessageBox(NULL,"【确定要进行删除操作吗？】","提示",MB\_OKCANCEL) == 1){

if(nameDelData(name)){

t->sum--;

SaveFile(Master);

SaveFile(Pet);

cout << "操作成功！"<<endl;

system("pause");

break;

}

else{

cout<<"操作失败！"<<endl;

system("pause");

continue;

}

}

else{

return;

}

}

system("pause");

}

}

else{

return;

}

}

}

//更改数据

void Change(){

system("cls");

cout << "\*\*\*\*\*\*请输入宠物编号：" << endl;

int num;

int day;

float price;

cin >> num;

pet \*k = NULL;

k = idGetPet(Pet, num);

if(k == NULL) cout << "请检查" << num <<"是否存在！";

else{

cout << "编号：" << k->pnum << endl;

cout << "名字：" << k->pname << endl;

cout << "品种：" << k->type << endl;

cout << "主人编号：" << k->owner << endl;

cout << "寄养天数:" << k->time << endl;

cout << "每日价格:" << k->dayprice << endl;

cout << "[1] 修改寄养天数" << endl;

cout << "[2] 修改寄养价格" << endl;

cout << "\*\*\*\*\*\*请选择1~2：" << endl;

int choice ;

cin >> choice;

if(choice == 1){

cout << "请输入寄养天数：" << endl;

cin >> day;

k->time = day;

SaveFile(Pet);

cout << "操作成功！"<<endl;

}

else if(choice == 2){

cout << "请输入寄养价格：" << endl;

cin >> price;

k->dayprice = price;

SaveFile(Pet);

cout << "操作成功！"<<endl;

}

else{

cout << "无此选项！" << endl;

}

}

system("pause");

}

//增加数据

void Add(){

int id;

string name;

string ty;

int day, price;

char mnum1[10];

struct pet \*p, \*q = NULL, \*r = NULL;

while(true){

system("cls");

cout << "\*\*\*\*\*\*请输入要增加的宠物编号：" << endl;

cin >> id;

if(idGetPet(Pet,id) != NULL){

cout << "编号 "<<id<<" 已经存在!请重新输入！"<<endl;

system("pause");

continue;

}

else{

p = new pet;

p->pnum = id;

p->next = NULL;

cout << "请输入 宠物名字 " ;

cout << " 宠物品种 " ;

cout << " 主人编号 " ;

cout << " 寄养天数 " ;

cout << " 寄养每日价格 :" <<endl;

cout << "输入格式：一行数据&各数据之间有一空格。例如 ：momo 萨摩耶 b 1 600 " << endl <<"请输入：";

cin >> name;

cin >> ty;

cin >> mnum1;

cin >> day;

cin >> price;

p->pname = name;

p->type = ty;

strcpy(p->owner, mnum1);

p->time = day;

p->dayprice = price;

if(NULL == Pet || p->pnum <= Pet->pnum){

p->next = Pet;

Pet = p;

SaveFile(Pet);

cout << "操作成功！"<<endl;

//break;

}

else{

r = Pet;

q = Pet->next;

while(q != NULL){

if(p->pnum > q->pnum){

r = q;

q = q->next;

}

else {

break;

}

}

r->next = p;

p->next = q;

master \*t = idGetMaster(Master, p->owner);

if(t != NULL){

t->sum++;

SaveFile(Pet);

SaveFile(Master);

cout<< "操作成功！"<<endl;

break;

}

else{

cout << "未查到此顾客，需新增主人信息！" << endl;

master \*p2, \*q2, \*r2;

char name2[10];

int phone2, sum2;

p2 = new master;

p2->next = NULL;

strcpy(p2->mnum, mnum1);

cout << "请输入 主人名字 ";

cout << " 联系方式 " ;

cout << " 寄养宠物数量：" << endl;

cout << "输入格式：一行数据&各数据之间有一空格。"<< endl << " 请输入：";

cin >> name2;

cin >> phone2;

cin >> sum2;

strcpy(p2->mname, name2);

p2->phone = phone2;

p2->sum = sum2;

r2 = Master;

q2 = Master->next;

r2->next = p2;

p2->next = q2;

SaveFile(Pet);

SaveFile(Master);

cout<< "操作成功！"<<endl;

system("pause");

break;

}

}

}

}

}

//客户消费排序

void ToSort(){

if(Master == NULL || Pet == NULL){

return;

}

float m,n;

master \*p, \*q, \*bigger, \*a, \*b;

for(p = Master; p != NULL; p = p->next){

bigger = p;

q=p->next;

while( q != NULL){

m=bigger->expense;

n=q->expense;

if(m < n){

bigger = q;

}

q=q->next;

}

if(bigger != p){

swap(bigger->mnum, p->mnum);

swap(bigger->mname, p->mname);

swap(bigger->phone , p->phone);

swap(bigger->sum , p->sum);

swap(bigger->expense , p->expense);

}

}

}

//排序模块

void Sort(){

pet \*p = Pet;

master \*p1;

while(p != NULL){

p1 = idGetMaster(Master, p->owner);

p1->expense += (p->time) \* (p->dayprice);

p = p->next;

}

ToSort();

master \*o = Master;

while(o != NULL){

p = Pet;

SetConsoleTextAttribute(GetStdHandle(STD\_OUTPUT\_HANDLE), FOREGROUND\_INTENSITY | FOREGROUND\_RED | FOREGROUND\_GREEN | FOREGROUND\_BLUE);

cout << "主人编号：" << o->mnum << endl;

cout << "主人名字：" << o->mname << endl;

cout << "联系方式：" << o->phone << endl;

cout << "寄养宠物个数：" << o->sum << endl;

cout << "总花费:" << o->expense << endl;

SetConsoleTextAttribute(GetStdHandle(STD\_OUTPUT\_HANDLE), FOREGROUND\_INTENSITY | FOREGROUND\_RED | FOREGROUND\_GREEN);

cout << "拥有宠物："<<endl;

while(p!=NULL){

if(strcmp(p->owner, o->mnum)==0){

cout << " 宠物名字：" << p->pname << endl;

cout << " 品种：" << p->type << endl << endl;

}

p=p->next;

}

cout << endl;

o = o->next;

}

system("pause");

}

4.2.3 文件读取、删除获取结点等工具函数

//保存数据至文件

void SaveFile(pet \*&Pet){

ofstream fout;

fout.open("pet.txt", ios\_base::binary);

pet \*p = Pet;

while(p != NULL){

if(p->next != NULL){

fout << p->pnum << " " << p->pname << " " << p->type << " " << p->owner << " " << p->time << " " << p->dayprice << endl;

p = p->next;

}

else{

fout << p->pnum << " " << p->pname << " " << p->type << " " << p->owner << " " << p->time << " " << p->dayprice ;

p = p->next;

}

}

fout.close();

return;

}

//保存数据至文件

void SaveFile(master \*&Master){

ofstream fout;

fout.open("master.txt", ios\_base::binary);

master \*p = Master;

while(p != NULL){

if(p->next != NULL){

fout << p->mnum << " " << p->mname << " " << p->phone << " " << p->sum << endl;

p = p->next;

}

else{

fout << p->mnum << " " << p->mname << " " << p->phone << " " << p->sum ;

p = p->next;

}

}

fout.close();

return;

}

//依据id删除链表结点

bool idDelData(int id){

pet \*p = Pet, \*q = NULL;

if(id == Pet -> pnum){

Pet = Pet->next;

delete p;

return true;

}

q = p->next;

while(NULL != q && id != q -> pnum){

p = q;

q = q -> next;

}

if(NULL != q){

p->next = q->next;

delete q;

return true;

}

return false;

}

//依据name删除链表结点

bool nameDelData( string name){

pet \*p = Pet, \*q = NULL;

if(name == Pet -> pname){

Pet = p->next;

delete p;

return true;

}

q = p->next;

while(NULL != q && name != q -> pname){

p = q;

q = q -> next;

}

if(NULL != q){

p->next = q->next;

delete q;

return true;

}

return false;

}

//依据id删除链表结点

bool idDelData( char num[]){

master \*p = Master, \*q = NULL;

if(strcmp(num, Master->mnum)==0){

Master = p->next;

delete p;

return true;

}

q = p->next;

while(NULL != q && strcmp(num, q -> mnum) != 0){

p = q;

q = q -> next;

}

if(NULL != q){

p->next = q->next;

delete q;

return true;

}

return false;

}

//用id获取链表结点

pet \*idGetPet(pet \*&Pet, int id)

{

pet \*q = Pet;

if(NULL == Pet) return NULL;

if(id == Pet -> pnum){

return Pet;

}

while(NULL != q && id != q -> pnum){

q = q -> next;

}

return q;

}

//用name获取链表结点

pet \*nameGetPet(pet \*&Pet, string name)

{

pet \*q = Pet;

if(NULL == Pet) return NULL;

if(name == Pet -> pname){

return Pet;

}

while(NULL != q && name != q -> pname){

q = q -> next;

}

return q;

}

//用id获取链表结点

master \*idGetMaster(master \*&Master, char id[]){

master \*q = Master;

if(NULL == Master) return NULL;

if(strcmp(id,Master -> mnum)==0){

return Master;

}

while(NULL != q && strcmp(id, q -> mnum)!=0){

q = q -> next;

}

return q;

}

//用name获取链表结点

master \*nameGetMaster(master \*&Master, char name[])

{

master \*q = Master;

if(NULL == Master) return NULL;

if(name == Master->mname){

return Master;

}

while(NULL != q && name != q->mname){

q = q -> next;

}

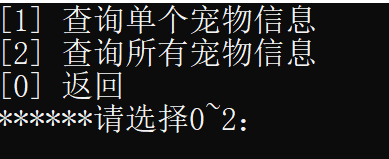
return q;

}

1. 功能测试
   1. 初始化界面

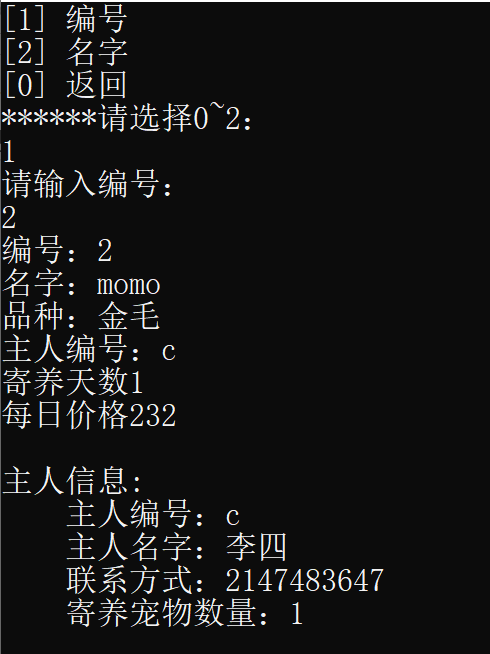


* 1. 查询宠物信息

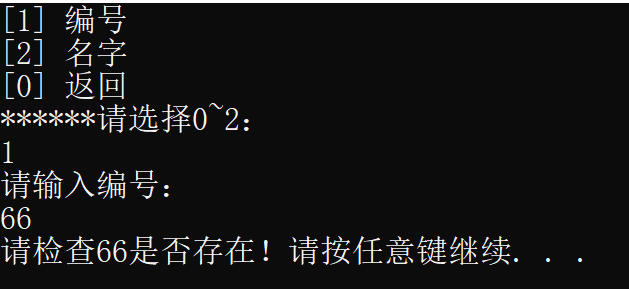


5.2.1 根据编号查询信息

宠物存在，输出宠物及主人信息

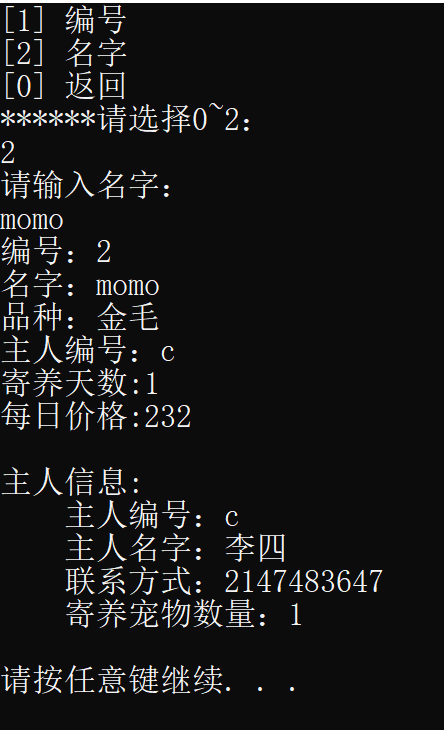


宠物不存在，输出提示信息

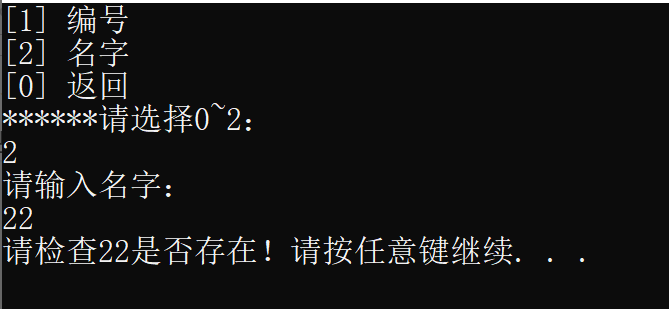


5.2.2 根据名字查询

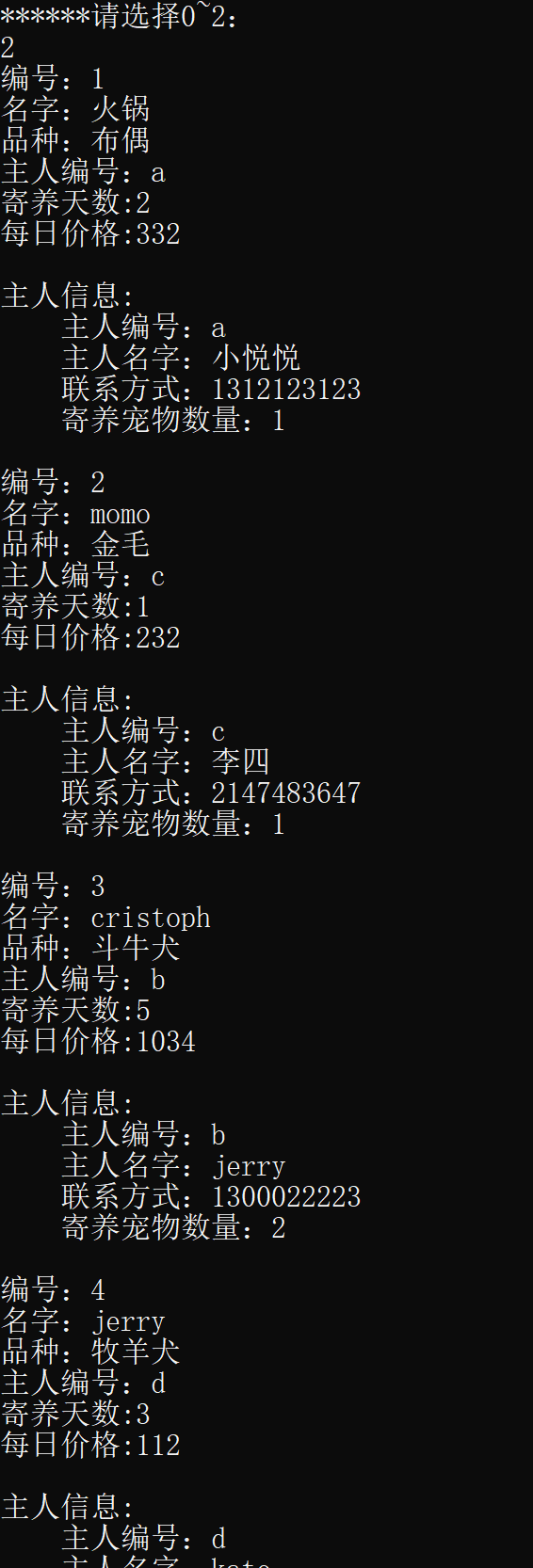
宠物存在，输出宠物及主人信息



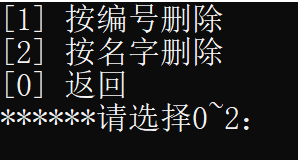
宠物不存在，输出提示信息



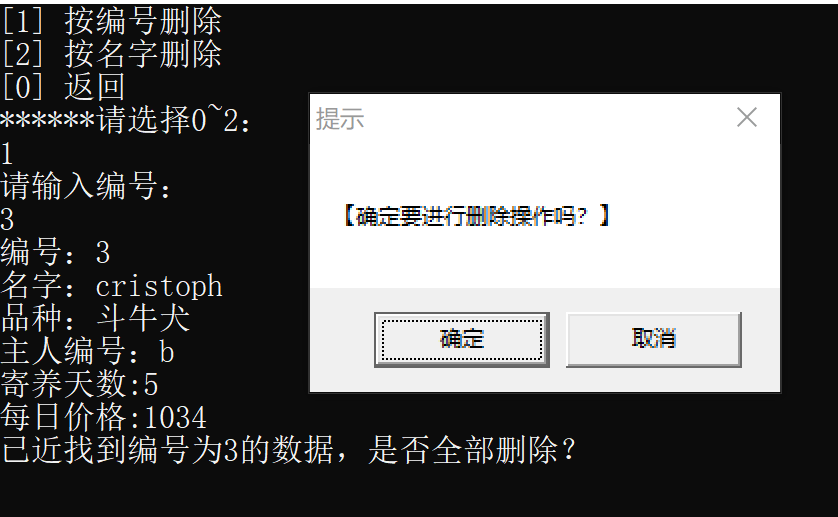
5.2.3 查询所有宠物信息

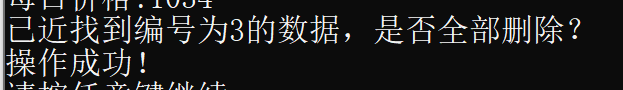


* 1. 删除宠物信息

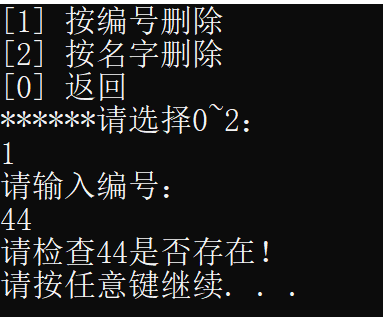


宠物存在，删除信息。



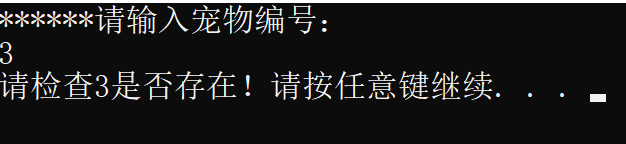


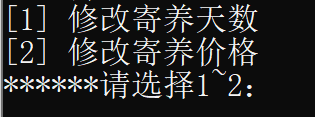
宠物不存在，输出提示信息。



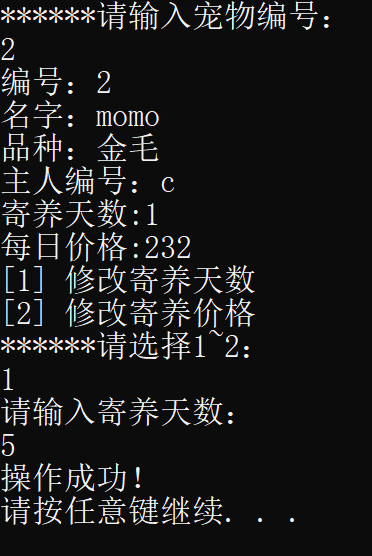
* 1. 修改宠物信息

不存在宠物信息，输出提示

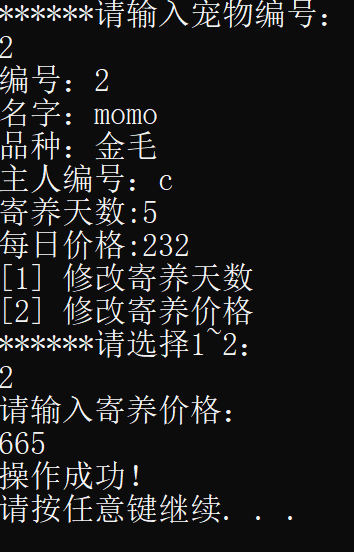




5.4.2 修改天数

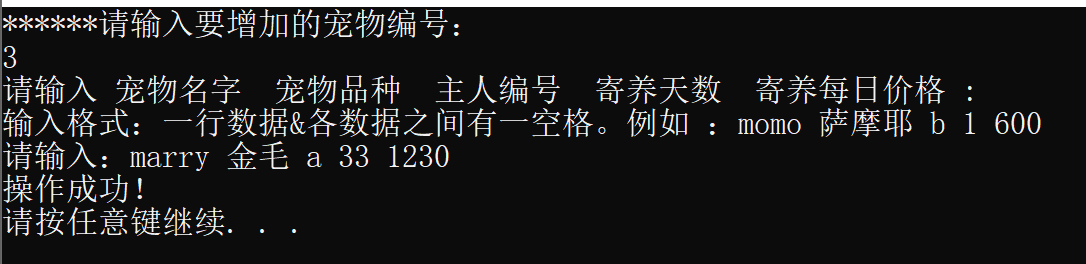


5.4.3 修改价格

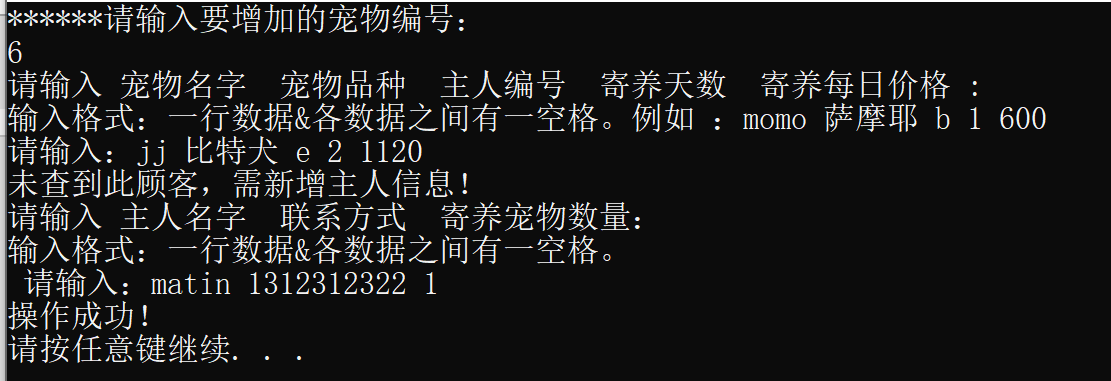


* 1. 插入宠物信息

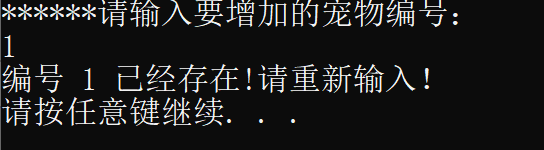
id不存在，增加宠物。



缺少主人信息，输出提示。

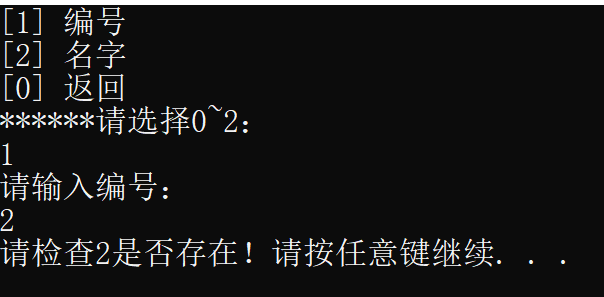


id存在，输出提示信息。

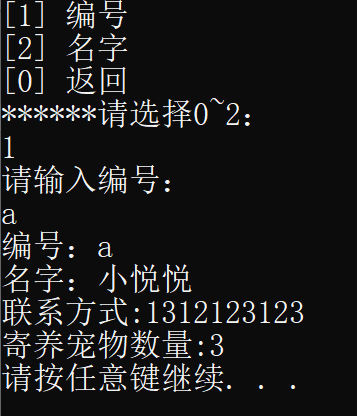


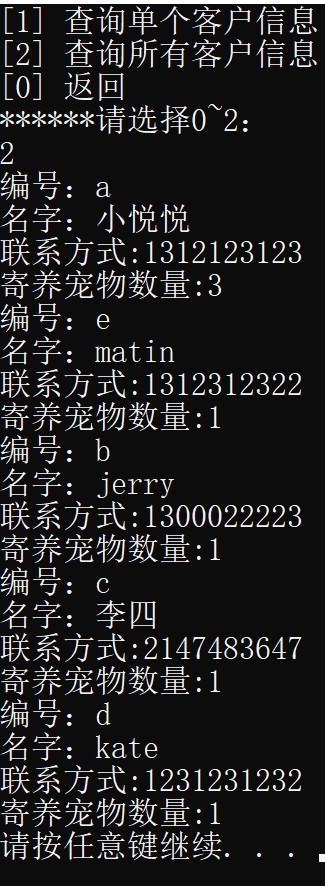
* 1. 查询主人信息

若不存在，输出提示。



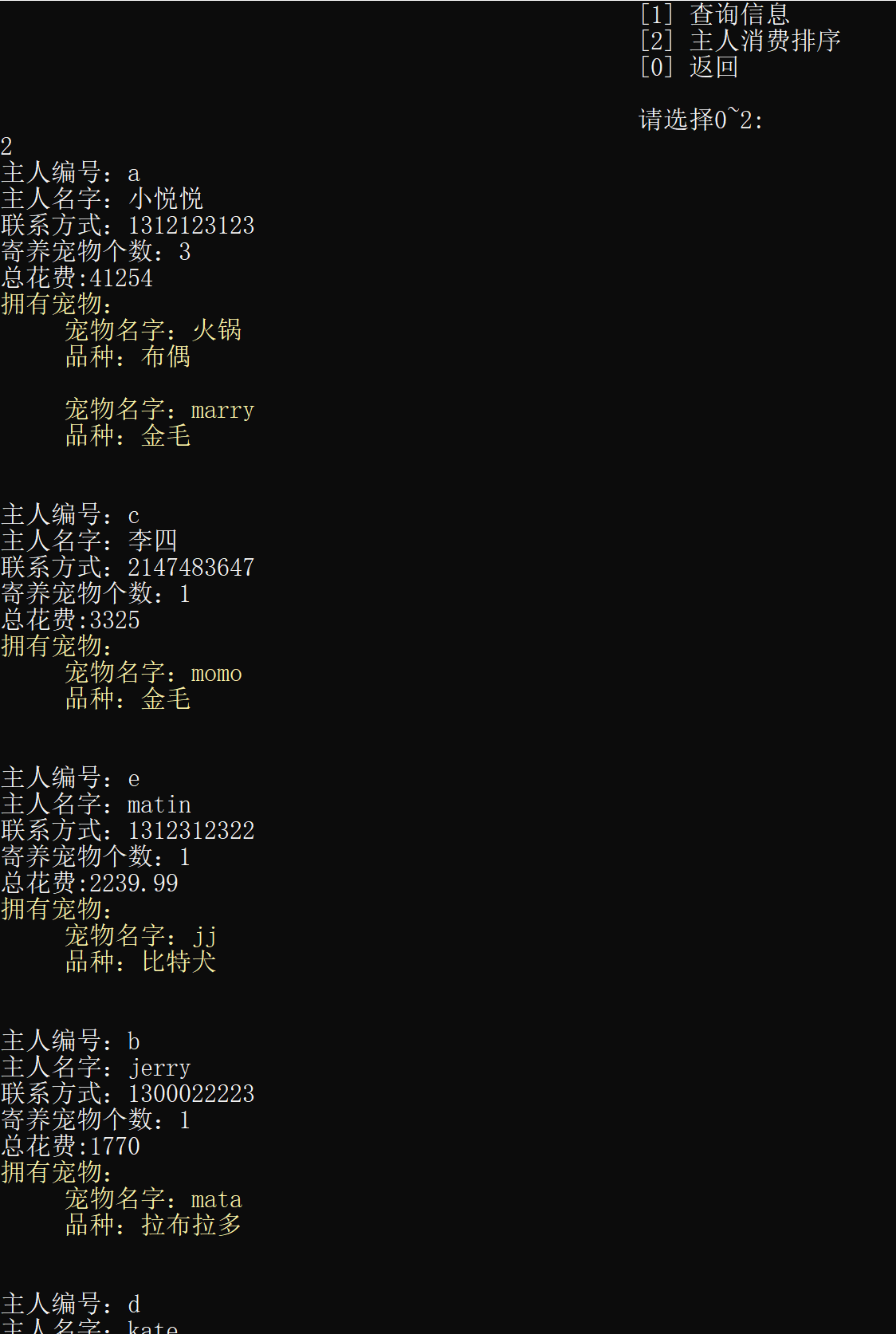
存在，则输出信息。



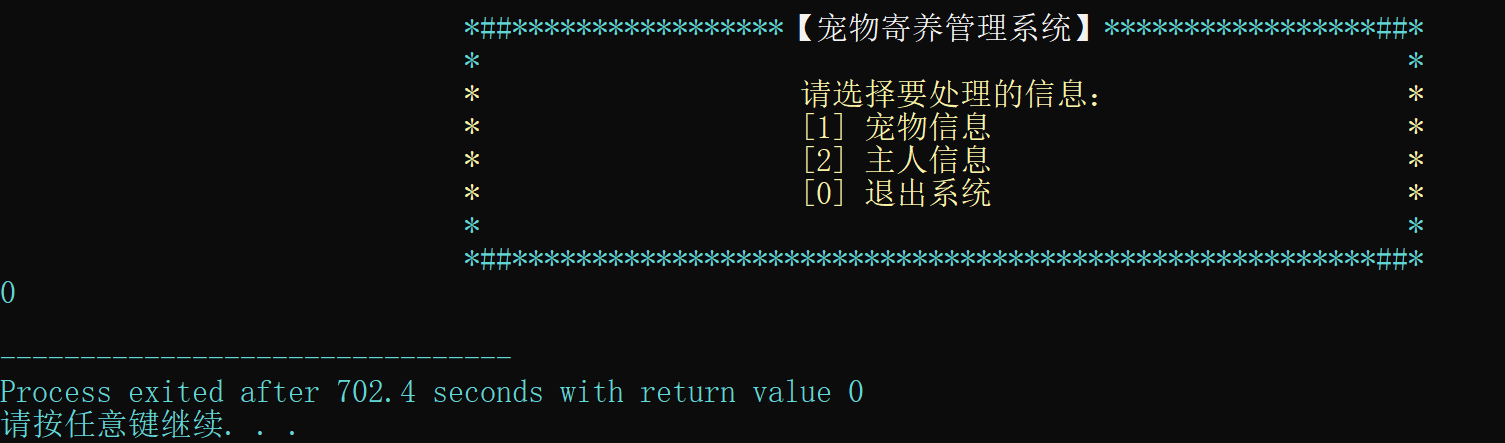


* 1. 消费排序

从高到低输出客户信息及其宠物信息。



* 1. 退出系统



1. 总 结

1、函数之间进行值传递，导致输出的文件产生乱码，故改用全局变量，不必引用，可直接用void处理。

2、char不能直接用“=”传递和比较，需要strcpy和strcmp。

3、同样功能的函数应对不同文件时，不需要各自命名，可以用函数重载，便于记忆。

4、保存文件时每次最后一行都会出现乱码，经过长时间的检测才发现是文件最后有回车，导致输入产生问题。因此在保存文件函数内添加了一个判断条件，如果p->next是NULL，则不输出回车。

优化方案：

1、增加宠物突发情况记录，如生病等，并计入主人的总消费。

2、可加入自动判断宠物寄养价格的功能，只需输入品种，自动记录价格，无需手动输入，导致产生错误。

3、界面简陋，需要进一步美化窗口设计。