

# 人工智能导论

## 实验报告

实验题目： 班级管理系统

姓 名： 李玮瀚

学 号： 2020212265

日 期： 2020.12.6

自我评分： A+

自我评分说明：A+，A，B+，B，B-，C，D，分别对应分数 95、90、85、80、75、70、60

## 诚信声明

本人郑重承诺：本实验程序和实验报告均是本人独立学习和工作所获得的成果。尽我所知，实验报告中除特别标注的地方外，不包含其他同学已经发表或撰写过的成果；实验程序中对代码工作的任何帮助者所作的贡献均做了明确的说明，并表达了谢意。

如有抄袭，本人愿意承担因此而造成的任何后果。

特此声明。

签名：\_\_\_\_\_李玮瀚\_\_\_\_\_

日期：\_\_\_\_\_2020.12.6\_\_\_\_\_

## 程序引用说明

序号	引用项	来源	相同代码行数
1	查找函数	《书名》	
2	构造有序链表函数	互联网网址	
3			
小计			

总代码行数\_\_\_\_\_；引用占比\_\_\_\_\_

## 1、实验简介

【实验内容的简要说明，具体说明实验完成的功能和性能要求】

编写班级管理系统实现要求功能，读写文件，查询学生信息等功能。

## 2、程序框架

【实验程序共包含哪些函数，一一列举函数名和函数功能】

(以下函数均包含在头文件中)

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <time.h>
#include <ctype.h>

struct birthday { ... };
struct student { ... };
int all() { ... };
struct monitor { ... };
void read(struct monitor* s) { ... };
void coutinf(struct monitor* s, int i) { ... };
void hello() { ... };
void sort(struct monitor* s, int len) { ... };
void write(struct monitor* s) { ... };
void writes(struct monitor* s) { ... };
string judtell(int num) { ... };
void add(struct monitor* s) { ... };

int searchname(struct monitor* s, string name) { ... };
int searchnum(struct monitor* s, string num) { ... };
void findhomeinf(struct monitor* s, string home) { ... };
void findhome(struct monitor* s, string name) { ... };
void findhomen(struct monitor* s, string num) { ... };
void findroom(struct monitor* s, string name) { ... };
void findroom(struct monitor* s, string num) { ... };
void findtell(struct monitor* s, string name) { ... };
void findtell(struct monitor* s, string num) { ... };
void changeclass(struct monitor* s, string num) { ... };
void changeclass(struct monitor* s, string name) { ... };
```

• 结构体生日嵌套在结构体学生里面，在monitor结构体中指针指向创建动态结构体student数组。

• All函数通过统计txt文件行数来统计班级总人数  
• Read函数完成读取txt中信息放入内存的结构体中，以便访问和操作数据。

• Write和writes函数负责将内存中某一条学生信息写入文件中，其中writes函数一开始不进行换行（write函数一开始总进行换行操作），只有在写入第一个学生信息时进行调用。

• Add函数用以添加学生信息，其中调用write和read函数，时刻保持内存中数据与txt文件一致

• Change class (n) 函数用于更改学生班级，用以实现转入转出功能。

• Searchname (num) 函数用于查询学生信息

• Hello用于打印开始菜单  
• Sort用于将学生按生日排序  
• Coutinf用于打印制定学生信息

• Judtell函数用以判断星座  
• findtell和findtelln函数用以查询学生星座。  
(findxxx (n) 一系列函数功能类似不做赘述 home为家乡 room为宿舍)

### 3、关键代码实现

【哪些函数是你认为最能体现自己工作成果的函数，说明函数实现基本思想（可用文字或图表示），以及具体的实验步骤（用伪代码或带注释代码）】

（本程序由于使用指针，所以每次移动指针都要对指针进行还原，关键代码中不予叙述该过程，一般还原方式为使用计数器，每当指针++计数器也++，在函数结尾减去计数器即可）

#### 3.1 all函数（统计班级人数）

代码如下

```
int all() //班级总人数实现成功
{
    ifstream inf;
    string sline;
    string s1, s11, s12;
    int temp = 0;
    inf.open("data1.txt", ios::in);
    while (getline(inf, s1))
    {
        istringstream sin(sline);
        temp++;
        sin >> s11 >> s12;
    }
    //cout << "班级总人数为: " << temp - 1 << endl;
    return temp;
}
```

使用fstream库

定义读取流（inf）

每次getline（获取txt文档的一整行）都会让temp++（计数器统计行数）  
统计行数来统计学生总数

#### 3.2 Read读取信息函数

（从左到右）

```
void read(struct monitor* s) //用于更新存储于结构体内的数据，成功
{
    string s0;
    int temp = all();
    ifstream inf;
    string sline;
    string a[1000][11];
    int b[1000][11];
    inf.open("data1.txt", ios::in);
    if (!inf)
    {
        cerr << "Open Error!" << endl;
        exit(1);
    }
    //cout << "i see name" << endl; //debug
    for (int i = 0; i < all(); i++)
    {
        for (int j = 0; j < 11; j++)
        {
            if (j != 5)
                inf >> a[i][j];
            else
                inf >> b[i][j];
        }
    }
}
```

```
for (int i1 = 0; i1 < all(); i1++)
{
    s->ptr->class1 = a[i1][0];
    //cout << "i see name" << endl; //debug
    s->ptr->num = a[i1][1];
    s->ptr->name = a[i1][2];
    s->ptr->classnum = a[i1][3];
    s->ptr->bth.year = b[i1][5] / 10000;
    s->ptr->bth.month = b[i1][5] % 10000 / 100;
    s->ptr->bth.date = b[i1][5] % 10000 % 100;
    s->ptr->sex = a[i1][4];
    s->ptr->home = a[i1][6];
    s->ptr->room = a[i1][7];
    s->ptr->course = a[i1][8];
    s->ptr->hobby = a[i1][9];
    s->ptr->organ = a[i1][10];
    s->ptr++;
}

inf.close();
for (int temp1 = 0; temp1 < all(); temp1++)
```

原理为创建数组，先用ifstream讲txt中文件读入二维数组中，再将数组中数据写入结构体数组中。

（不完整部分为指针还原过程，此处指针还原比较特殊，需要减去人数再+1，因为人数增加了一个）

### 3.3 Write系列函数

```
void write(struct monitor* s)
{
    ofstream out;
    out.open("data1.txt", ios::app);
    out << endl << s->ptr->class1 << " " << s->ptr->num << " " << s->ptr->name << " " << s->ptr->classnum << " "
    << s->ptr->sex << " " << s->ptr->bth.date + s->ptr->bth.month * 100 + s->ptr->bth.year * 10000 <<
    " " << s->ptr->home << " " << s->ptr->room << " " << s->ptr->course << " " << s->ptr->hobby << " " << s->ptr->organ;
    out.close();
}

void writes(struct monitor* s)
{
    ofstream out;
    out.open("data1.txt", ios::app);
    out << s->ptr->class1 << " " << s->ptr->num << " " << s->ptr->name << " " << s->ptr->classnum << " "
    << s->ptr->sex << " " << s->ptr->bth.date + s->ptr->bth.month * 100 + s->ptr->bth.year * 10000 <<
    " " << s->ptr->home << " " << s->ptr->room << " " << s->ptr->course << " " << s->ptr->hobby << " " << s->ptr->organ;
    out.close();
}
```

同理运用fstream讲结构体的数据写入txt中。

### 3.4 Add添加学生信息并写入txt

代码如下

```
void add(struct monitor *s)
{
    //cout << "如果想在输出过程中取消添加学生信息请在任意信息中输入取消" << endl;
    s->ptr=>ptr+1;
    string num; //学号
    cout << "请输入学号:" << endl;
    cin >> num;
    s->ptr->num = num;
    string sex; //性别 男 女 0
    while (true)
    {
        cout << "请输入性别:" << endl;
        cin >> sex;
        if (sex == "男" || sex == "女")
        {
            s->ptr->sex = sex;
            break;
        }
        else
        {
            cout << "输入有误!" << endl;
        }
    }

    string name; //姓名
    cout << "请输入姓名:" << endl;
    cin >> name;
    s->ptr->name = name;
    string class1; //班级
    cout << "请输入班级:" << endl;
    cin >> class1;
    s->ptr->class1 = class1;
    string classnum; //校内序号
    cout << "请输入校内序号:" << endl;
    cin >> classnum;
    s->ptr->classnum = classnum;
    string hobby; //爱好
    cout << "请输入爱好:" << endl;
    cin >> hobby;
    s->ptr->hobby = hobby;
    string organ; //组织 学生组织
    cout << "请输入学生组织:" << endl;
    cin >> organ;
    s->ptr->organ = organ;
    struct bthday bth; //生日
    int year, month, date;
    cout << "请输入生日, (年份)" << endl;
    cin >> year;
    s->ptr->bth.year = year;
    while (true)
    {
        cout << "请输入生日, (月份)" << endl;
        cin >> month;
        if (month >= 1 && month <= 12)
        {
            break;
        }
        else
        {
            cout << "输入月份非法! 请重新输入!" << endl;
        }
    }
    s->ptr->bth.month = month;
    while (true)
    {
        cout << "请输入生日, (日期)" << endl;
        cin >> date;
        if (date >= 1 && date <= 31)
        {
            break;
        }
        else
        {
            cout << "输入日期非法! 请重新输入!" << endl;
        }
    }
    s->ptr->bth.date = date;
    if ((s->ptr->bth.date + s->ptr->bth.month * 100) >= 1210)
    {
        s->ptr->age = -(s->ptr->bth.year - 2020 + 1);
    }
    else
    {
        s->ptr->age = -(s->ptr->bth.year - 2020);
    }
    cout << "该学生的年龄为" << s->ptr->age << endl;
    string home; //家乡
    cout << "请输入家乡:" << endl;
    cin >> home;
    s->ptr->home = home;
    string course; //课程
    cout << "请输入课程:" << endl;
    cin >> course;
    s->ptr->course = course;
    string room; //宿舍
    cout << "请输入宿舍:" << endl;
    cin >> room;
    s->ptr->room = room;
    s->ptr->tell = justify(s->ptr->bth.date + s->ptr->bth.month * 100);
    write(s);
    s->ptr = s->ptr + all() + 1;
    read(s);
}
```

如图所示

利用假的死循环，判断用户输入信息是否合法，如若不合法，则要求用户重新输入，并调用write函数，讲数据写入txt中，最后还原指针。

### 3.5 Sort年龄排序函数

```
void sort(struct monitor* s, int len) //年龄由大到小
{
    for (int i = 0; i < len - 1; i++)
    {
        for (int j = 0; j < len - i - 1; j++)
        {
            if (s->ptr->bth.year > s->ptr->bth.year)
            {
                struct student temp = s->ptr[j];
                s->ptr[j] = s->ptr[j+1];
                s->ptr[j+1] = temp;
            }
            else if (s->ptr[j].bth.year == s->ptr[j+1].bth.year)
            {
                if (s->ptr[j].bth.month > s->ptr[j+1].bth.month)
                {
                    struct student temp1 = s->ptr[j];
                    s->ptr[j] = s->ptr[j+1];
                    s->ptr[j+1] = temp1;
                }
                else if (s->ptr[j].bth.month == s->ptr[j+1].bth.month)
                {
                    if (s->ptr[j].bth.date > s->ptr[j+1].bth.date)
                    {
                        struct student temp2 = s->ptr[j];
                        s->ptr[j] = s->ptr[j+1];
                        s->ptr[j+1] = temp2;
                    }
                }
            }
        }
    }
}
```

如图所示代码

利用student结构体的bth嵌套结构体的函数比较学生年龄大小

利用冒泡排序算法进行排序，主函数中再调用coutinf函数输出所有学生信息

### 3.6 主函数

利用假的死循环防止用户输入规定以外的选项导致程序崩溃

只有输入正确才能跳出死循环

代码如下

额外功能

- 更改字体颜色

```
SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE),  
FOREGROUND_INTENSITY | FOREGROUND_GREEN | FOREGROUND_BLUE);
```

代码如下

- 输入错误后或者调用功能结束后进行清屏

代码如下

```
system("cls");
```

### 3.7 Find系列函数（展示一个做代表，其余函数与之原理相同，不予赘述）

```
void findhome(monitor* s, string name)  
{  
    string home;  
    string sname;  
    int temp2 = 0;  
    int temp3 = 0;  
    int temp1 = 0;  
    while(s->ptr->name != name)  
    {  
        s->ptr++;  
        temp1++;  
        if (temp1 > all())  
        {  
            cout << "查无此人!" << endl;  
            break;  
        }  
    }  
    if (s->ptr->name == name)  
    {  
        home = s->ptr->home;  
        sname=s->ptr->name;  
        s->ptr -= temp1;  
        while (temp3<=all())  
        {  
            temp3++;  
            if (s->ptr->home == home&& s->ptr->name!=sname)  
            {  
                cout << "ta与您来自同一地区, 是您的老乡!" << endl;  
                cout << s->ptr->name<<endl;  
                temp2++;  
            }  
            s->ptr++;  
        }  
        if(temp2==0)  
            cout<<"很遗憾! 您没有老乡"<<endl;  
        s->ptr=temp3;  
    }  
    else  
    {  
        s->ptr -= temp1;  
    }  
}
```

```
temp3++;  
if (s->ptr->home == home&& s->ptr->name!=sname)  
{  
    cout << "ta与您来自同一地区, 是您的老乡!" << endl;  
    cout << s->ptr->name<<endl;  
    temp2++;  
}  
s->ptr++;  
if(temp2==0)  
    cout<<"很遗憾! 您没有老乡"<<endl;  
s->ptr=temp3;  
else  
{  
    s->ptr -= temp1;  
}
```

## 4、不足

对指针的运用还不是很熟练，操作完指针总是忘记还原指针指向。

## 5、心得体会

通过此次作业，提升了我的综合运用能力，运用所学知识编写一个较为完整的程序，很有成就感，虽说中途也有困难，但通过csdn社区的帮助和自己的思考都能够比较顺利的解决。