

C++实现通讯录

- 一、需求分析
- 二、代码实现
 - 2.1 菜单
 - 2.2 主函数框架
 - 2.3 退出功能
 - 2.4 添加联系人
 - 2.5 显示联系人
 - 2.6 删除联系人
 - 2.7 查找联系人
 - 2.8 修改联系人
 - 2.9 清空通讯录
- 三、效果演示

一、需求分析

利用C++实现通讯录管理系统，功能如下：

- 添加联系人（姓名、性别、年龄、联系电话、住址）
- 显示联系人
- 删除联系人
- 查找联系人
- 修改联系人
- 清空通讯录
- 退出通讯录

二、代码实现

2.1 菜单

实现用户选择功能界面：

```
//显示菜单
void showMenu()
{
    cout << "*****" << endl;
    cout << "***** 1、添加联系人 *****" << endl;
    cout << "***** 2、显示联系人 *****" << endl;
    cout << "***** 3、删除联系人 *****" << endl;
    cout << "***** 4、查找联系人 *****" << endl;
    cout << "***** 5、修改联系人 *****" << endl;
    cout << "***** 6、清空通讯录 *****" << endl;
```

```

cout << "***** 0、退出通讯录 *****" << endl;
cout << "*****" << endl;
}

```

2.2 主函数框架

在实现上述7个功能之前，首先需要搭建一下基本框架，基本框架主函数如下：

```

int main()
{
    int select = 0; //创建用户选择变量

    while (true)
    {
        showMenu(); //调用菜单
        cin >> select; //获取用户输入

        switch (select)
        {
            case 1: //1、添加联系人
                break;
            case 2: //2、显示联系人
                break;
            case 3: //3、删除联系人
                break;
            case 4: //4、查找联系人
                break;
            case 5: //5、修改联系人
                break;
            case 6: //6、清空通讯录
                break;
            case 0: //0、退出通讯录
                break;
            default:
                break;
        }
    }

    system("pause");
    return 0;
}

```

2.3 退出功能

选择0则推出通讯录管理系统，case0中添加如下代码：

```

case 0: //0、退出通讯录
    cout << "系统已退出 ,欢迎下次使用" << endl;
    system("pause");
    return 0;
    break;

```

2.4 添加联系人

添加联系人的基本功能为：

- 构建联系人结构体
- 构建通讯录结构体
- 封装添加联系人函数

构建联系人结构体：

```
//联系人结构体
struct Person
{
    string m_Name;    //姓名
    int m_sex;        //性别
    int m_Age;        //年龄
    string m_Phone;   //电话
    string m_Addr;    //住址
};
```

构建通讯录结构体：

```
#define MAX 1000    //设置最大人数1000
//通讯录结构体
struct AddressBooks
{
    struct Person personArray[MAX]; //通讯录中保存的联系人数组
    int m_Size; //联系人个数
};
```

定义添加联系人功能：

```
//1、添加联系人
void addPerson(AddressBooks *abs)
{
    //判断通讯录是否超过1000,如果超过则不添加
    if (abs->m_Size == MAX)
    {
        cout << "通讯录已满,无法添加" << endl;
        return;
    }
    else
    {
        //姓名
        string name;
        cout << "请输入姓名:" << endl;
        cin >> name;
        abs->personArray[abs->m_Size].m_Name = name;
        //性别
        int sex=0;
        cout << "请输入性别:" << endl;
        cout << "1 —— 男" << endl;
        cout << "2 —— 女" << endl;
        while (true)
        {
```

```

        cin >> sex;
        if (sex == 1 || sex == 2)
        {
            abs->personArray[abs->m_Size].m_Sex = sex;
            break;
        }
        cout << "输入有误,请重新输入!" << endl;

    }
    //年龄
    int age=0;
    cout << "请输入年龄:" << endl;
    cin >> age;
    abs->personArray[abs->m_Size].m_Age = age;
    //电话
    string phone;
    cout << "请输入电话:" << endl;
    cin >> phone;
    abs->personArray[abs->m_Size].m_Phone = phone;
    //住址
    string address;
    cout << "请输入住址:" << endl;
    cin >> address;
    abs->personArray[abs->m_Size].m_Addr = address;

    //更新通讯录人数
    abs->m_Size++;

    cout << "添加成功" << endl;

    system("pause");//请按任意键继续
    system("cls");//清屏操作
}
}

```

主函数中实现：

```

//创建通讯录结构体变量
AddressBooks abs;
//初始化通讯录中当前人员个数
abs.m_Size = 0;

```

case1中修改代码为：

```

case 1:    //1、添加联系人
    addPerson(&abs);//地址传递,修改实参
    break;

```

2.5 显示联系人

定义显示联系人函数：

```

//2、显示联系人
void showPerson(AddressBooks *abs)
{

```

```

if (abs->m_Size == 0) {
    cout << "通讯录为空..." << endl;
}
else
{
    for (int i = 0; i < abs->m_Size; i++)
    {
        cout << "姓名:" << abs->personArray[i].m_Name << "\t";
        cout << "性别:" << (abs->personArray[i].m_Sex==1 ? "男":"女") << "\t"; //三目运算符
        cout << "年龄:" << abs->personArray[i].m_Age << "\t";
        cout << "电话:" << abs->personArray[i].m_Phone << "\t";
        cout << "住址:" << abs->personArray[i].m_Addr << endl;
    }
}
system("pause");//请按任意键继续
system("cls");//清屏操作
}

```

case2中修改代码为:

```

case 2:    //2、显示联系人
    showPerson(&abs);
    break;

```

2.6 删除联系人

定义检测联系人是否存在函数，若存在则返回联系人的下标编号，否则返回-1:

```

int isExist(AddressBooks *abs, string name)
{
    for (int i = 0; i < abs->m_Size; i++)
    {
        //若存在
        if (abs->personArray[i].m_Name == name)
        {
            return i;//返回下标编号
        }
    }
    return -1;//如果没有找到
}

```

定义删除联系人函数:

```

//3、删除联系人
void deletePerson(AddressBooks *abs)
{
    cout << "请输入要删除的联系人" << endl;
    string name;
    cin >> name;
    int ret = isExist(abs, name);//判断联系人是否存在
    if (ret != -1) {
        //查到了,则删除数据(利用后覆盖前实现)
        for (int i = ret; i < abs->m_Size; i++)
        {
            //数据迁移

```

```

        abs->personArray[i] = abs->personArray[i + 1];
    }
    abs->m_Size--; //更新通讯录中人员数
    cout << "删除成功" << endl;
}
else {
    //未查到
    cout << "未查到该联系人" << endl;
}
system("pause");//请按任意键继续
system("cls");//清屏操作
}

```

case3中修改代码为：

```

case 3:    //3、删除联系人
    deletePerson(&abs);
    break;

```

2.7 查找联系人

定义查找联系人函数：

```

//4、查找联系人
void findPerson(AddressBooks *abs)
{
    cout << "请输入要查找的联系人:" << endl;
    string name;
    cin >> name;
    int ret = isExist(abs, name); //判断联系人是否存在
    if (ret != -1) {
        //查到了,则显示数据
        cout << "姓名:" << abs->personArray[ret].m_Name << "\t";
        cout << "性别:" << (abs->personArray[ret].m_Sex == 1 ? "男" : "女") << "\t"; //三目运算符
        cout << "年龄:" << abs->personArray[ret].m_Age << "\t";
        cout << "电话:" << abs->personArray[ret].m_Phone << "\t";
        cout << "住址:" << abs->personArray[ret].m_Addr << endl;
    }
    else {
        //未查到
        cout << "未查到该联系人" << endl;
    }
    system("pause");//请按任意键继续
    system("cls");//清屏操作
}

```

case4中修改代码为：

```

case 4:    //4、查找联系人
    findPerson(&abs);
    break;

```

2.8 修改联系人

定义修改联系人函数：

```
//5、修改联系人
void modifyPerson(AddressBooks *abs)
{
    cout << "请输入要修改的联系人:" << endl;
    string name;
    cin >> name;
    int ret = isExist(abs, name); //判断联系人是否存在
    if (ret != -1) {
        //查到了,则修改数据
        //姓名
        string name;
        cout << "请输入姓名:" << endl;
        cin >> name;
        abs->personArray[ret].m_Name = name;
        //性别
        int sex = 0;
        cout << "请输入性别:" << endl;
        cout << "1 —— 男" << endl;
        cout << "2 —— 女" << endl;
        while (true)
        {
            cin >> sex;
            if (sex == 1 || sex == 2)
            {
                abs->personArray[ret].m_Sex = sex;
                break;
            }
            cout << "输入有误,请重新输入!" << endl;
        }
        //年龄
        int age = 0;
        cout << "请输入年龄:" << endl;
        cin >> age;
        abs->personArray[ret].m_Age = age;
        //电话
        string phone;
        cout << "请输入电话:" << endl;
        cin >> phone;
        abs->personArray[ret].m_Phone = phone;
        //住址
        string address;
        cout << "请输入住址:" << endl;
        cin >> address;
        abs->personArray[ret].m_Addr = address;
    }
    else {
        //未查到
        cout << "未查到该联系人" << endl;
    }
    system("pause");//请按任意键继续
    system("cls");//清屏操作
}
```

case5中修改代码为：

```
case 5:      //5、修改联系人
    modifyPerson(&abs);
    break;
```

2.9 清空通讯录

定义清空通讯录函数：

```
//6、清空通讯录
void clearPerson(AddressBooks *abs)
{
    abs->m_Size = 0; //将联系人数量置为0
    cout << "通讯录已清空..." << endl;

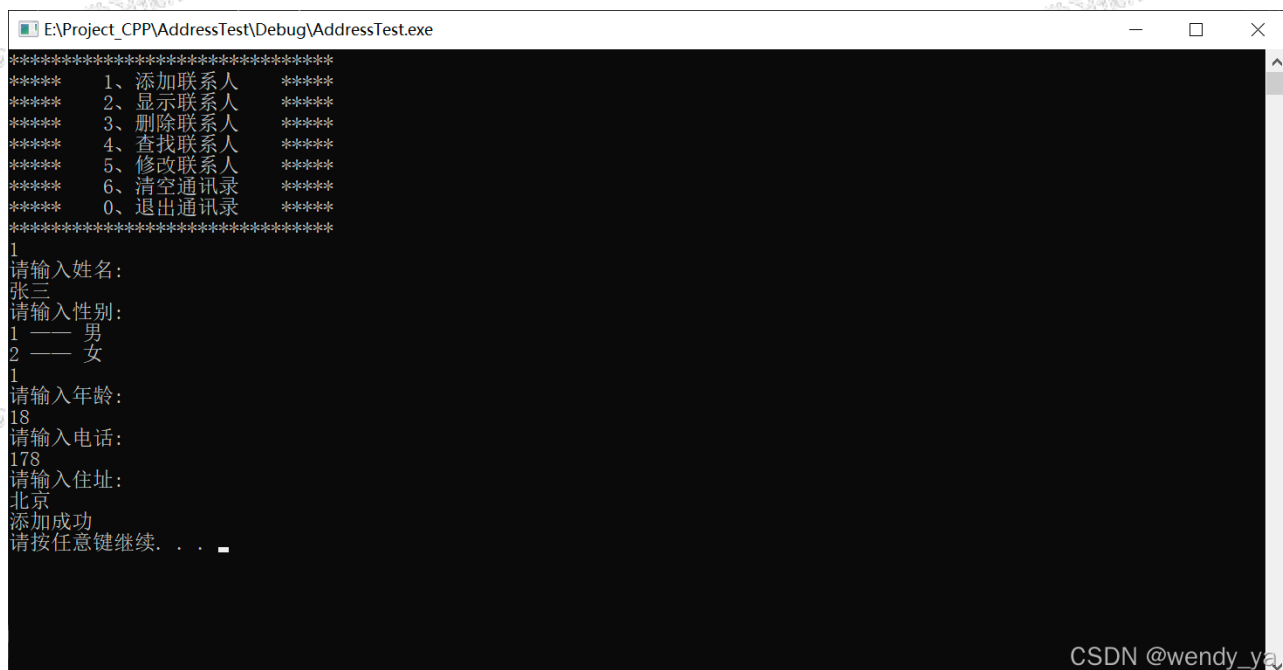
    system("pause"); //请按任意键继续
    system("cls"); //清屏操作
}
```

case6中修改代码为：

```
case 6:      //6、清空通讯录
    clearPerson(&abs);
    break;
```

三、效果演示

完整效果如下：



```
E:\Project_CPP\AddressTest\Debug\AddressTest.exe
*****
***** 1、添加联系人 *****
***** 2、显示联系人 *****
***** 3、删除联系人 *****
***** 4、查找联系人 *****
***** 5、修改联系人 *****
***** 6、清空通讯录 *****
***** 0、退出通讯录 *****
*****
1
请输入姓名:
张三
请输入性别:
1 —— 男
2 —— 女
1
请输入年龄:
18
请输入电话:
178
请输入住址:
北京
添加成功
请按任意键继续. . .
```