

## 1. 代码实践

```
#include <iostream>
#include <string>
#define MAX 1000
using namespace std;

//联系人结构体
struct Person
{
    string m_name; //姓名
    int m_Sex; //性别 1: 男 2: 女
    int m_Age; //年龄
    string m_phone; //电话
    string m_Addr; //住址
};

//通讯录结构体
struct Addressbooks
{
    struct Person personArray[MAX];
    int m_size; //通讯录中当前的人数
};

void addPerson(Addressbooks *abs)
{
    //先判断通讯录是否已满，如果满了就不再添加
    if(abs->m_size==MAX)
        cout << "通讯录已满，请勿继续添加" << endl;
    else
    {
        //添加具体联系人
        string name;
        cout << "请输入姓名:  " << endl;
        cin >> name;
        abs->personArray[abs->m_size].m_name=name;

        int sex; //性别 1: 男 2: 女
        while(true)
        {
            cout << "请输入性别:  " << endl;
            cin >> sex;
            if(sex==1||sex==2)
            {
```

```
        abs->personArray[abs->m_size].m_Sex=sex;
        break;
    }
    else
        cout << "输入有误，请重新输入" << endl;
};

int age; //年龄
cout << "请输入年龄： " << endl;
cin >> age;
abs->personArray[abs->m_size].m_Age=age;

string phone; //电话
cout << "请输入电话： " << endl;
cin >> phone;
abs->personArray[abs->m_size].m_phone=phone;

string addr; //住址
cout << "请输入住址： " << endl;
cin >> addr;
abs->personArray[abs->m_size].m_Addr=addr;

abs->m_size++;
cout << "添加成功" << endl;

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

void showPerson(Addressbooks *abs)
{
    if(abs->m_size==0)
    {
        cout << "当前记录为空" << 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" ;
            cout << "年龄： " << abs->personArray[i].m_Age << "\t" ;
            cout << "电话： " << abs->personArray[i].m_phone << "\t" ;
```

```
        cout << "地址:  " << abs->personArray[i].m_Addr << "\t" ;
        cout << endl;
    }
    system("pause"); //请按任意键继续
    system("cls"); //清屏
}

int isExist(Addressbooks *abs,string name)
{
    for(int i=0;i<abs->m_size;i++)
    {
        if(abs->personArray[i].m_name==name)
        {
            return i;    //如果找到的话, 返回下标
        }
    }
    return -1;
}

void delectPerson(Addressbooks *abs)
{
    cout << "请输入删除联系人的姓名:  " << endl;
    string name;
    cin >> name;
    if(isExist(abs,name)==-1)
    {
        cout << "查无此人" << endl;
    }
    else
    {
        for(int i=isExist(abs,name);i<abs->m_size;i++)
        {
            abs->personArray[i]=abs->personArray[i+1];
        }
        abs->m_size--;
        cout << "删除成功" << endl;
    }
    system("pause"); //请按任意键继续
    system("cls"); //清屏
}

void searchPerson(Addressbooks *abs)
{
    cout << "请输入您要查找的联系人:  " << endl;
```

```
string name;
cin >> name;
int ret=isExist(abs,name);
if(ret!=-1)
{
    cout << "姓名:  " << abs->personArray[ret].m_name << "\t" ;
    cout << "性别:  " << (abs->personArray[ret].m_Sex==1?"男":"女") << "\t" ;
    cout << "年龄:  " << abs->personArray[ret].m_Age << "\t" ;
    cout << "电话:  " << abs->personArray[ret].m_phone << "\t" ;
    cout << "地址:  " << abs->personArray[ret].m_Addr << "\t" ;
    cout << endl;
}
else
{
    cout << "该联系人不存在" << endl;
}
system("pause"); //请按任意键继续
system("cls"); //清屏
}
```

备注：昨天只联系到前三个功能实现，今日补充了剩下的全部功能。

## 2. 计算机基础知识整理

### 2.1 JDK、JRE、JVM

- JDK（开发工具包）
- JRE（运行环境）
- JVM（JAVA 虚拟机）

JDK：程序开发者必须安装 JDK，提供了编译、运行 Java 程序所需的各种工具和资源，包括 Java 编译器、Java 运行环境 JRE，以及常用的 Java 基础类库等，是整个 JAVA 的核心。

JRE：Java 运行环境，并不是一个开发环境，所以没有包含任何开发工具（如编译器和调试器），只是针对于使用 Java 程序的用户。

### 2.2 红黑树

R-B Tree，一种特殊的二叉查找树。红黑树的每个结点上都有存储位表示节点的颜色，可以是红或黑。

红黑树的特性：

- ① 每个结点或者为黑，或者为红。
- ② 根节点为黑色。
- ③ 每个叶子结点是黑色。（这里是指为 NULL 的叶子结点）
- ④ 如果一个结点是红色，则它的子节点必须是黑色。
- ⑤ 从一个结点到该结点的子孙结点的所有路径上包含相同数目的黑结点。

红黑树的应用：

主要用它来存储有序的数据，时间复杂度是  $O(\log n)$ ，效率非常高。

### 3. 当日工作总结

- ① 熟悉了一些 Git 相关操作
- ② 上传学习笔记至小组仓库。