

通讯录管理系统

目 录

一、任务书	1
二、总体设计	2
三、各功能模块实现	13
3.1. 增加联系人	13
3.2. 删除联系人	15
3.3. 修改联系人	17
3.4. 查找联系人	19
3.5. 浏览联系人	20
3.6. 转存联系人	22
四、总结	25

一、任务书

手机通讯录中的联系人的信息既可以存储在手机中，也可以存储在手机卡中，也可以同时存储在两个位置上（假设每个位置上的存储容量为 1000，即手机卡中或手机上最多只能存储 1000 个联系人）。存储在手机中的联系人的信息只包含姓名和电话号码两项信息。存储在电话卡上的联系人的信息除了上面提到的两项信息外，还包含籍贯，QQ 号等信息。

根据通用的手机通讯录的使用方式，采用OOP(Object Oriented Programming, 面向对象编程)方法编写一个手机通讯录管理。

要求：

创建文本文件，记录联系人的信息（需要创建两个文本文件，分别存储手机和手机卡上的联系人的信息）。

以菜单方式工作(字符界面即可)

存储在手机上的联系人的信息包括：姓名和电话号码；存储在手机卡上的联系人的信息包括姓名，籍贯，电话号码，QQ 号等信息

管理系统的功能包括：

新建联系人：添加新的联系人（添加时确定是添加到手机上还是手机卡中, 并且要检查目标位置上是否存在相同的电话号码）

删除：删除一个联系人（输入电话号码，删除该联系人。说明，如果两个存储位置上都存在该联系人的话，需要在两个存储位置上都要进行删除操作）

修改：修改某个联系人的信息（输入电话号码，查询到联系人之后进行信息的修改。说明，如果两个存储位置上都存在该联系人的话，需要在两个存储位置上都要进行修改操作）

查询：根据名字查询联系人信息（查询结果可能不唯一）

浏览：显示所有联系人的信息

将联系人的信息从手机转存到手机卡上（转存过程中, 要避免重复数据(姓名和电话号码相同)的存在 ;并且在转存是要检查容量是否受限 。下同。)

将联系人的信息从手机卡上转存到手机上（同时要避免重复数据的存在）

要支持继承、多态、重载(运算符重载、函数重载)等面向对象的基本特点。

二、总体设计

通讯录管理系统当中定义如下的七个类。

1. 菜单类

Menu {

Menu 类的成员函数包括：

static void user()

前置条件：无用户登录

输入：无

功能：显示用户登录通讯录系统以欢迎用户

输出：系统名称

后置条件：某用户已登录系统

static void operation()

前置条件：某用户已登录系统

输入：无

功能：提示用户选择要执行的操作

输出：8 个操作提示

后置条件：用户输入操作序号

static void addoption()

前置条件：某用户选择进行添加操作

输入：无

功能：提醒用户选择添加联系人的位置

输出： 可选择两个添加位置

后置条件：可开始输入被添加联系人的信息

static void display ()

前置条件：通讯录中已有联系人信息

输入：无

功能：提醒用户选择查看的联系人的存放位置

输出：选择显示手机还是手机卡里的联系人

后置条件：跳转出联系人信息

static void copy ()

前置条件：某联系人已被添加

输入：无

功能：提醒用户选择修改联系人的操作

输出：复制还是剪切联系人信息

后置条件：操作成功提示

}

2. 手机联系人类：

`Mpcontact{`

`Mpcontact` 类的数据成员包括：

`name` 表示手机联系人姓名；

`phonenum` 表示手机联系人电话号

`Mpcontact` 类的成员函数包括：

`Mpcontact(string n="\0", string p="\0"):name(n), phonenum(p)`

功能：有参构造函数

`string getname()`

输入：无

功能：用来返回类内私有属性`name`

`void setname(string name)`

功能：用来设置`name` 的属性值

`string getphonenum()`

输入：无

功能：用来返回类内私有属性 `phonenum`

`void setphonenum(string phonenum)`

功能：用来设置 `phonenum` 的属性值

`void modify (string name,string phonenumbe)`

功能：修改手机联系人输入

`friend istream& operator>>(istream& in,Mpcontact& m)`

前置条件：已输入联系人

输入：联系人姓名和电话号码

功能：重载运算符>>，将手机联系人信息输入

后置条件：存入联系人信息

```
friend ostream& operator<<(ostream& out, Mpcontact& m)
```

输入：无

功能：重载运算符<<，输出联系人信息

输出：联系人姓名和电话号

```
}
```

3. 手机卡联系人类：

```
Mccontact:public Mpcontact
```

```
{
```

数据成员包括：

address 联系人地址

qq 联系人 QQ 号

成员函数包括：

```
void getaddress
```

输入：无

功能：用来返回类内私有属性address

```
string setaddress
```

功能：用来设置 address 的属性值

```
void getqq
```

输入：无

功能：用来返回类内私有属性 qq

```
string setqq
```

功能：用来设置 qq 的属性值

```
void modify
```

功能：修改手机卡联系人输入

```
friend istream& operator>>(istream& in, Mccontact& m)
```

输入：联系人姓名、电话号码、地址和 QQ

功能：重载运算符>>，将手机卡联系人信息输入

后置条件：存入联系人信息

```
friend ostream& operator<<(ostream& out, Mccontact& m)
```

输入：无

功能：重载运算符<<，输出联系人信息

输出：联系人姓名、电话号码、地址和 QQ

```
};
```

4. 通讯簿类：

```
Commbook
```

```
{
```

Commbook 类的成员函数包括：

```
virtual void addb()=0;
```

功能：增加联系人

```
virtual void deleteb(string phonenumber)=0;
```

功能：删除联系人

```
virtual void displayb()=0;
```

功能：显示所有联系人

```
virtual void revisionb(string name, string phonenumber, string
```

```
address, string qq, int i)=0;
```

功能：修改某一联系人

```
virtual void queryb(string name)=0;
```

功能：查找某个联系人

```
};
```

5. 手机通讯簿类：

```
Mpcommbook:public Commbook
```

```
{
```

数据成员包括：

quantity 手机联系人的数量

Mpcontact mpcontact[max] 手机联系人对象数组

成员函数包括：

`Mpcommbook()`

前置条件：打开文本文件

输入：联系人姓名和电话号码

功能：无参构造函数，读取文本文件中的数据，并根据文件内容创建联系人对象数组

输出：无

后置条件：创建了联系人对象数组

`~Mpcommbook()`

前置条件：已有文本文档

输入：无

功能：析构函数：将对象数组中的内容写入到文本文件中

输出：无

后置条件：文本文档中写入联系人信息

`int getquantity()`

输入：无

功能：用来返回类内私有属性 `quantity`

`void setquantity(int i)`

功能：用来设置 `quantity` 的属性值

`Mpcontact getmpcontact(int i)`

输入：无

功能：用来返回类内私有对象数组

`void setmpcontact(string name, string phonenumber)`

功能：用来设置对象数组的值

`void addb()`

前置条件：手机号不存在，无重复信息

输入：联系人数组

功能：添加联系人信息

输出：添加成功提醒

后置条件：成功添加联系人数组

`void deleteb(string phonenumber)`

前置条件：联系人信息不为空

输入：无

功能：删除联系人

输出：无

后置条件：删除一条联系人信息

`void displayb()`

前置条件：联系人信息不为空

输入：无

功能：显示所有联系人信息

输出：已保存的联系人信息

后置条件：成功显示通讯簿

`void revisionb(string name, string phonenumber, string address, string qq, int i)`

前置条件：联系人信息不为空

输入：修改信息

功能：修改联系人信息

输出：修改成功提醒

后置条件：修改成功

`void queryb(string name)`

前置条件：有该name 的联系人

输入：姓名

功能：查找某个联系人

输出：联系人信息

后置条件：查找到的信息

`int judge(string phonenumber)`

输入：电话号码

功能：返回所输入电话号码的数组位置，以辅助其他成员函数

输出：无

`};`

6. 手机卡通讯簿类：

`Mccommbook:public Commbook`

`{`

数据成员包括：

`int quantity=0` 手机卡联系人的数量

`Mccontact mccontact[max]` 手机卡联系人对象数组

成员函数包括:

`Mccommbook()`

前置条件: 打开文本文件

输入: 联系人姓名、电话号码、地址和 QQ 号

功能: 无参构造函数, 读取文本文件中的数据, 并根据文件内容创建联系人对象数组

输出: 无

后置条件: 创建了联系人对象数组

`~Mccommbook()`

前置条件: 已有文本文件

输入: 无

功能: 析构函数: 将对象数组中的内容写入到文本文件中

输出: 无

后置条件: 文本文件中写入联系人信息

`int getquantity()`

输入: 无

功能: 用来返回类内私有属性 `quantity`

`Mccontact getmccontact(int i)`

输入: 无

功能: 用来返回类内私有对象数组

`void setmccontact(string name, string phonenumber, string address="\0", string qq="\0")`

功能: 用来设置对象数组的值

`void addb()`

前置条件: 手机号不存在, 无重复信息

输入: 联系人数组

功能: 添加联系人信息

输出: 添加成功提醒

后置条件: 成功添加联系人数组

`void deleteb(string phonenumber)`

前置条件：联系人信息不为空

输入：无

功能：删除联系人

输出：无

后置条件：删除一条联系人信息

`void displayb()`

前置条件：联系人信息不为空

输入：无

功能：显示所有联系人信息

输出：已保存的联系人信息

后置条件：成功显示通讯簿

`void revisionb(string name, string phonenumber, string address, string qq, int i)`

前置条件：联系人信息不为空

输入：修改信息

功能：修改联系人信息

输出：修改成功提醒

后置条件：修改成功

`void queryb(string name)`

前置条件：有该name 的联系人

输入：姓名

功能：查找某个联系人

输出：联系人信息

后置条件：查找到的信息

输入：电话号码

功能：返回所输入电话号码的数组位置，以辅助其他成员函数

输出：无

`int judge(string phonenumber)`前置条件：

输入：电话号码

功能：返回所输入电话号码的数组位置，以辅助其他成员函数

输出：无

`};`

7. 用户类:

User

{

数据成员包括:

Mpcommbook mpcommbook 手机通讯簿对象

Mccommbook mccommbook 手机卡通讯簿对象

Commbook* b

成员函数包括:

User() {}

功能: 无参构造类同名函数

void addb()

前置条件: 用户选择添加按钮

输入: 1 或 2

功能: 添加联系人到通讯簿

输出: 添加成功提醒

后置条件: 添加一条联系人信息

void deleteb()

前置条件: 通讯簿中有要删除的联系人

输入: 要删除联系人的电话号码

功能: 删除一条联系人信息

输出: 删除成功提醒

后置条件: 被删除的联系人信息不存在

void displayb()

前置条件: 通讯簿有联系人

输入: 1 或 2

功能: 显示所有联系人信息

输出: 电话或电话卡的联系人信息

后置条件: 显示所有信息

void revisionb()

前置条件: 用电话号码找到要被修改的联系人电话

输入: 修改后的信息

功能: 修改某个联系人信息

输出：修改成功提醒

后置条件：联系人信息被修改

`void queryb()`

前置条件：存在该联系人

输入：要查询的联系人姓名

功能：查询某个联系人的信息

输出：一条联系人信息

`void transter1()`

前置条件：存在要转存的信息

输入：1 或 2

功能：手机转存到手机卡

输出：转存成功提醒

后置条件：手机卡中存入信息

`void transter2()`

前置条件：存在要转存的信息

输入：1 或 2

功能：手机卡转存到手机

输出：转存成功提醒

后置条件：手机中存入信息

`void move1()`

前置条件：在 `transter1` 中选择 1, 并且手机卡通讯簿数据未溢出

功能：手机通讯簿移动到手机卡通讯簿

输出：移动完成提醒

`void replicate1()`

前置条件：在 `transter1` 中选择 2, 手机卡通讯簿数据未溢出

功能：手机通讯簿 复制到手机卡 通讯簿

输出：复制完成提醒

`void move2()`

前置条件：在 `transter2` 中选择 1, 手机通讯簿数据未溢出

功能：手机卡通讯簿移动到手机通讯簿

输出：移动完成提醒

`void replicate2()`

前置条件：在 transter2 中选择 2,手机通讯簿数据未溢出

功能：手机卡通讯簿复制到手机通讯簿

输出：复制完成提醒

} ;

三、各功能模块实现

3.1. 增加联系人

在主函数中定义用户类对象，调用addb 函数并且在用户类的成员函数中实现多态。基类指针分别指向手机通讯簿对象和手机卡通讯簿对象，进而调用派生类的虚函数，实现不同位置的增加联系人。

流程图见图3.1

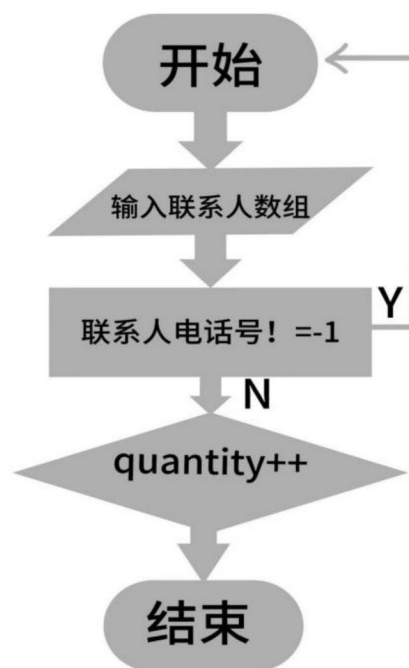


图 3.1 增加联系人流程图

运行效果见图 3.2 、3.3

```
*****通讯录管理系统*****
*****请输入您的操作:*****
***** 1. 添加联系人 *****
***** 2. 删除联系人 *****
***** 3. 修改联系人 *****
***** 4. 查找联系人 *****
***** 5. 浏览联系人 *****
***** 6. 将联系人的信息从手机转存到手机卡上 *****
***** 7. 将联系人的信息从手机卡上转存到手机上 *****
***** 8. 保存并退出 *****
1
请选择您的操作: 1. 将联系人添加到手机中
2. 将联系人添加到手机卡中
1
请输入添加联系人信息:
name:狮子王
phonenumber:10086
添加成功
```

图 3.2 增加联系人到手机运行效果图

```
*****请输入您的操作:*****
***** 1. 添加联系人 *****
***** 2. 删除联系人 *****
***** 3. 修改联系人 *****
***** 4. 查找联系人 *****
***** 5. 浏览联系人 *****
***** 6. 将联系人的信息从手机转存到手机卡上 *****
***** 7. 将联系人的信息从手机卡上转存到手机上 *****
***** 8. 保存并退出 *****
1
请选择您的操作: 1. 将联系人添加到手机中
2. 将联系人添加到手机卡中
2
请输入联系人信息
name:山楂丸
phonenumber:10010
address:sdau
qq:2817525555
添加成功
```

图 3.3 增加联系人到手机卡运行效果图

3.2. 删除联系人

在主函数中定义用户类对象，调用deleteb 函数实现删除联系人功能，删除前要判断联系人是否存在。

流程图见图3.4

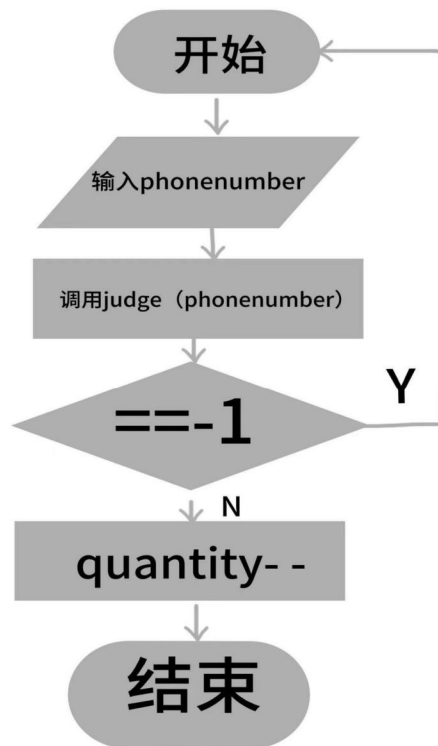


图 3.4 删除联系人流程图

运行效果见图 3.5

```
D:\程序设计b作业\2022215987_史志玮_课程设计源代码\x64\Debug\2022215987_史志玮_课程设计源代码.exe
添加成功
*****请输入您的操作: *****
***** 1. 添加联系人 *****
***** 2. 删除联系人 *****
***** 3. 修改联系人 *****
***** 4. 查找联系人 *****
***** 5. 浏览联系人 *****
***** 6. 将联系人的信息从手机转存到手机卡上 *****
***** 7. 将联系人的信息从手机卡上转存到手机上 *****
***** 8. 保存并退出 *****
2
请输入要删除联系人的手机号:
10010
删除成功
```

图 3.5 删除联系人流程图

3.3. 修改联系人

在主函数中定义用户类对象，调用revisionb 函数并且判断联系人是否存在来修改联系人。

流程图见图3.6

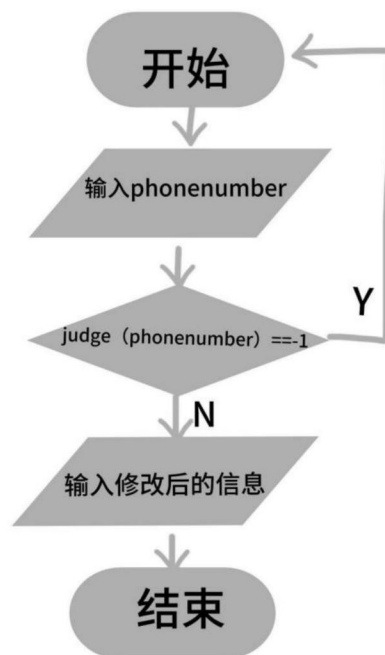


图 3.6 修改联系人流程图

运行效果见图 3.7

D:\程序设计b作业\2022215987_史志玮_课程设计源代码\64\Debug\2022215987_史志玮_课程设计源代码.exe

```

删除成功
*****请输入您的操作：*****
***** 1. 添加联系人 *****
***** 2. 删除联系人 *****
***** 3. 修改联系人 *****
***** 4. 查找联系人 *****
***** 5. 浏览联系人 *****
***** 6. 将联系人的信息从手机转存到手机卡上 *****
***** 7. 将联系人的信息从手机卡上转存到手机上 *****
***** 8. 保存并退出 *****
}
请输入要修改联系人的手机号
10086
请输入修改后的姓名
山楂丸
请输入修改后的住址
sdau
请输入修改后的QQ
2817525555
修改成功
  
```

图 3.7 修改联系人运行效果图

3. 4. 查找联系人

在主函数中定义用户类对象，调用queryb 函数实现查找某一联系人。
流程图见图3. 8



图 3. 8 查找联系人流程图

运行效果见图 3. 9

```
D:\程序设计b作业\2022215987_史志玮_课程设计源代码\x64\Debug\2022215987_史志玮_课程设计源代码.exe
修改成功
*****请输入您的操作：*****
***** 1. 添加联系人 *****
***** 2. 删除联系人 *****
***** 3. 修改联系人 *****
***** 4. 查找联系人 *****
***** 5. 浏览联系人 *****
***** 6. 将联系人的信息从手机转存到手机卡上 *****
***** 7. 将联系人的信息从手机卡上转存到手机上 *****
***** 8. 保存并退出 *****
4
请输入要查询联系人的姓名
山楂丸
手机中该联系人的信息为:
name:山楂丸      phonenumber:10086
手机卡中未找到该联系人信息
```

图 3. 9 查找联系人运行效果图

3. 5. 浏览联系人

在主函数中定义用户类对象，调用 displayb 函数并且在用户类的成员函数中实现多态。基类指针分别指向手机通讯簿对象和手机卡通讯簿对象，进而调用派生类的虚函数，实现浏览不同位置的联系人。

流程图见图 3. 10

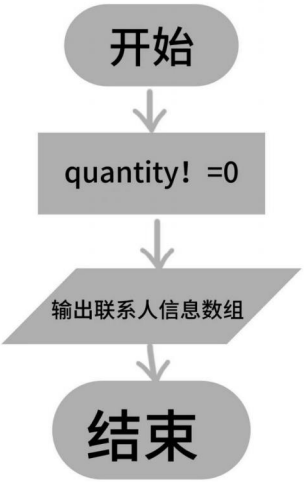


图 3. 10 浏览联系人流程图

运行效果见图 3. 11、3. 12

```
D:\程序设计b作业\2022215987_史志玮_课程设计源代码\x64\Debug\2022215987_史志玮_课程设计源代码.exe
***** 1. 添加联系人 *****
***** 2. 删除联系人 *****
***** 3. 修改联系人 *****
***** 4. 查找联系人 *****
***** 5. 浏览联系人 *****
***** 6. 将联系人的信息从手机转存到手机卡上 *****
***** 7. 将联系人的信息从手机卡上转存到手机上 *****
***** 8. 保存并退出 *****
5
请选择您的操作：1. 显示手机联系人
2. 显示手机卡联系人
1
手机中联系人的信息为：
name:山楂丸      phonenumber:10086
```

图 3. 11 浏览手机联系人运行效果图

```
D:\程序设计b作业\2022215987_史志玮_课程设计源代码\x64\Debug\2022215987_史志玮_课程设计源代码.exe

*****请输入您的操作：*****
***** 1. 添加联系人 *****
***** 2. 删除联系人 *****
***** 3. 修改联系人 *****
***** 4. 查找联系人 *****
***** 5. 浏览联系人 *****
***** 6. 将联系人的信息从手机转存到手机卡上 *****
***** 7. 将联系人的信息从手机卡上转存到手机上 *****
***** 8. 保存并退出 *****
5
请选择您的操作：1. 显示手机联系人
2. 显示手机卡联系人
2
手机卡中无联系人的信息
```

图 3.12 浏览手机卡联系人运行效果图

3.6. 转存联系人

在主函数中定义用户类对象，调用 transter1 函数将联系人信息从手机转存到手机卡，transter2 函数实现联系人从手机卡到手机的转存，转存前选择是复制还是移动。

流程图见图 3.13

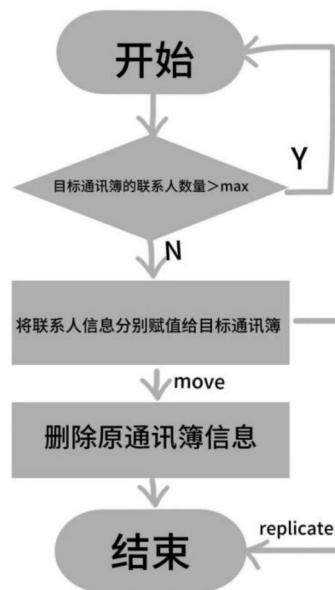


图 3.6 转存联系人流程图

运行效果见图 3.14、3.15、3.16、3.17

```

D:\程序设计b作业\2022215987_史志玮_课程设计源代码\x64\Debug\2022215987_史志玮_课程设计源代码.exe
*****请输入您的操作:*****
***** 1. 添加联系人 *****
***** 2. 删除联系人 *****
***** 3. 修改联系人 *****
***** 4. 查找联系人 *****
***** 5. 浏览联系人 *****
***** 6. 将联系人的信息从手机转存到手机卡上 *****
***** 7. 将联系人的信息从手机卡上转存到手机上 *****
***** 8. 保存并退出 *****
6
请输入转存方式: 1. 移动
2. 复制
1
移动完成
  
```

图 3.14 手机联系人移动到手机卡运行效果图

```
D:\程序设计b作业\2022215987_史志玮_课程设计源代码\x64\Debug\2022215987_史志玮_课程设计源代码.exe

*****请输入您的操作: *****
***** 1. 添加联系人 *****
***** 2. 删除联系人 *****
***** 3. 修改联系人 *****
***** 4. 查找联系人 *****
***** 5. 浏览联系人 *****
***** 6. 将联系人的信息从手机转存到手机卡上 *****
***** 7. 将联系人的信息从手机卡上转存到手机上 *****
***** 8. 保存并退出 *****
5
请输入转存方式: 1. 移动
2. 复制
2
复制完成
```

图 3.15 手机联系人复制到手机卡运行效果图

```
D:\程序设计b作业\2022215987_史志玮_课程设计源代码\x64\Debug\2022215987_史志玮_课程设计源代码.exe

*****请输入您的操作: *****
***** 1. 添加联系人 *****
***** 2. 删除联系人 *****
***** 3. 修改联系人 *****
***** 4. 查找联系人 *****
***** 5. 浏览联系人 *****
***** 6. 将联系人的信息从手机转存到手机卡上 *****
***** 7. 将联系人的信息从手机卡上转存到手机上 *****
***** 8. 保存并退出 *****
7
请输入转存方式: 1. 移动
2. 复制
1
移动完成
```

图 3.16 手机卡联系人移动到手机运行效果图

```
D:\程序设计b作业\2022215987_史志玮_课程设计源代码\x64\Debug\2022215987_史志玮_课程设计源代码.exe

*****请输入您的操作: *****
***** 1. 添加联系人 *****
***** 2. 删除联系人 *****
***** 3. 修改联系人 *****
***** 4. 查找联系人 *****
***** 5. 浏览联系人 *****
***** 6. 将联系人的信息从手机转存到手机卡上 *****
***** 7. 将联系人的信息从手机卡上转存到手机上 *****
***** 8. 保存并退出 *****
7
请输入转存方式: 1. 移动
2. 复制
2
复制完成
```

图 3.17 手机卡联系人复制到手机运行效果图

四、总结

虽然过程十分考验人的耐心，尤其是遇到代码出错时，很难找出自己的问题，但是在这期间我收获了很多，比如：

1. 在写代码前要从整体出发，想明白所需要实现的功能以及各个功能中应具备的操作，先把大体框架写出来，再填充内容；
2. 编写过程中，要注意规范性，各变量名称要具有逻辑，不然到后期出现错误自己都很难理解自己的代码，避免浪费不必要的时间；
3. 编写程序要考虑实际情况，从而使系统更加符合生活；
4. 注意代码分段以及具有逻辑的缩进，这样可以节省检查花费的时间。