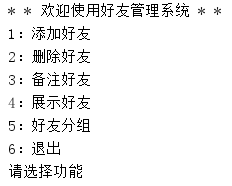
**好友管理系统**

如今的社交软件层出不穷，虽然功能千变万化，但都具有好友管理系统的基本功能，包括添加好友、删除好友、备注好友、展示好友、好友分组功能。下面是一个简单的好友管理系统的功能菜单，如图1所示。



1. 好友管理

图1中的好友管理系统中有5个功能，每个功能都对应一个序号，用户可根据提示“请输入您的选项”选择序号执行相应的操作，包括：

* 添加好友：用户根据提示“请输入要添加的好友：”输入要添加好友的姓名，添加后会提示“好友添加成功”。
* 删除好友：用户根据提示“请输入删除好友姓名：”输入要删除好友的姓名，删除后提示“删除成功”。
* 备注好友：用户根据提示“请输入要修改的好友姓名：”和“请输入修改后的好友姓名：”分别输入修改前和修改后的好友姓名，修改后会提示“备注成功”。
* 展示好友：展示用户功能分为展示所有好友和展示分组中的好友，如果用户选择展示所有好友，那么展示将好友列表中的所有好友进行展示；如果用户选择展示分组好友，那么根据用户选择的分组名展示此分组中的所有好友。
* 好友分组：好友分组功能用于将好友划分为不同的组，执行好友分组功能会提示用户是否创建新的分组。
* 退出：关闭好友管理系统。

本实例要求编写程序，实现如上所述功能的好友管理系统。

1. **实例目标**

通过完成本实例，读者应掌握以下知识或技能：

* 理解面向对象的思想
* 掌握类的定义以及对象的创建和使用
* 掌握如何在类中定义方法
* 熟练使用构造方法

1. **实例分析**

（1）好友管理系统中不仅需要保存好友名称，还需要保存分组以及分组中的好友，因此，我们需要明确以什么结构保存数据。根据功能的描述，我们可以将数据以['王五', '张三', '赵柳', {'家人': ['李四','刘备']},{'同事': ['李龙','张飞']}]结构保存。

（2）功能展示。功能展示用于向用户展示该程序中所包含的功能，以及接收用户输入的功能选项，实现用户人机交互的入口。

（3）添加好友。该功能逻辑相对简单，只需要将用户输入的好友名称添加到好友列表中即可。

（4）获取所有好友。该功能用于获取好友列表中的所有数据，包括分组中的数据，便于后期对好友列表中的数据操作。

（5）获取所有分组及其分组好友。该功能用于获取好友列表中分组的名称和分组的好友，不包括未分组的好友。

（6）获取分组名称。该功能仅仅用于获取所有的分组名称，用于后期判断用户输入的分组名是否合法。

（7）删除好友。此功能分为两种情况进行判断，一是：删除的好友未在分组内，二是：删除的好友在分组内。如果删除的好友未在分组内，那么直接在好友列表中删除；如果删除的好友在分组内，那么需要先获取所有分组和分组内的好友，根据用户输入的好友名称在对应的分组内进行删除。

（8）备注好友。备注好友功能与删除好友逻辑类似，需要分为两种情况修改好友的备注名称，若修改备注的好友未在分组内，则直接获取好友名称并对其重新赋值；若修改备注的好友在分组内，则需要找到该好友对应的分组并进行修改。

（9）展示好友。展示好友功能分为展示所有好友和展示指定分组的好友，展示所有好友，直接调用已经定义好的获取所有好友方法即可，展示指定分组的好友，需要根据用户输入的分组名进行展示分组内的好友名称。

（10）好友分组。好友分组功能是将好友划分为不同的组，在使用好友分组功能时，会询问用户是否创建新的分组，如果用户选择创建新的分组，那么由用户输入新的分组名并创建，新的分组会以{'家人': ['李四','刘备']}结构添加到好友列表中；如果用户选择使用已存在的分组，那么需要由用户输入具体的分组名和具体的好友名，在将好友名添加到分组前需要分别对分组名和好友名进行正确性校验，只有当分组名和好友名均存在时，才可以将好友名移动到指定的分组中。

（11）所有功能完成后，需要将定义的类实例化并调用入口函数。

1. **代码实现**

* 功能展示代码

class Friend:

def \_\_init\_\_(self):

self.friend\_li = []

def welcome(self):

print("\* \* 欢迎使用好友管理系统 \* \* ")

print("1：添加好友")

print("2：删除好友")

print("3：备注好友")

print("4：展示好友")

print("5：好友分组")

print("6：退出")

while True:

option = input("请选择功能\n")

# 添加好友

if option == '1':

self.add\_friend()

# 删除好友

elif option == '2':

self.del\_friend()

# 备注好友

elif option == '3':

self.modify\_friend()

# 展示好友

elif option == '4':

self.show\_friend()

# 分组好友

elif option == '5':

self.group\_friend()

elif option == '6':

break

* 添加好友具体代码如下：

def add\_friend(self):

add\_friend = input("请输入要添加的好友：")

self.friend\_li.append(add\_friend)

print('好友添加成功')

* 获取所有好友具体代码如下：

def get\_all\_friends(self):

new\_li = []

for friend\_li\_elem in self.friend\_li:

# 判断元素类型

if type(friend\_li\_elem) == dict:

# 遍历字典

[new\_li.append(dict\_elem\_name) for

dict\_elem in friend\_li\_elem.values()

for dict\_elem\_name in dict\_elem]

else:

new\_li.append(friend\_li\_elem)

return new\_li

* 获取所有分组及其好友具体代码如下：

def get\_all\_groups(self):

groups = []

for friend\_li\_elem in self.friend\_li:

if type(friend\_li\_elem) == dict:

groups.append(friend\_li\_elem)

return groups

* 获取所有分组名称具体代码如下：

def get\_all\_groups\_name(self):

groups\_name = []

for dict\_elem in self.get\_all\_groups():

for j in dict\_elem:

groups\_name.append(j)

return groups\_name

* 删除好友具体代码如下：

def del\_friend(self):

if len(self.friend\_li) !=0:

del\_name = input("请输入删除好友姓名：")

# 删除的好友未分组

if del\_name in self.friend\_li:

self.friend\_li.remove(del\_name)

print('删除成功')

else:

# 删除的好友在分组内

if del\_name in self.get\_all\_friends():

for group\_data in self.get\_all\_groups():

for group\_friend\_li in group\_data.values():

if del\_name in group\_friend\_li:

group\_friend\_li.remove(del\_name)

continue

print('删除成功')

else:

print('好友列表为空')

* 备注好友具体代码如下：

def modify\_friend(self):

friends = self.get\_all\_friends()

if len(friends)==0:

print('好友列表为空')

else:

before\_name = input("请输入要修改的好友姓名：")

after\_name = input("请输入修改后的好友姓名：")

if before\_name in self.friend\_li:

friend\_index = self.friend\_li.index(before\_name)

self.friend\_li[friend\_index] = after\_name

print("备注成功")

elif before\_name not in self.friend\_li:

for friend\_li\_elem in self.friend\_li:

if type(friend\_li\_elem) == dict:

for dict\_elem in friend\_li\_elem.values():

if before\_name in dict\_elem:

modify\_index = dict\_elem.index(before\_name)

dict\_elem[modify\_index] = after\_name

print('备注成功')

* 展示好友具体代码如下：

def show\_friend(self):

print("1.展示所有好友")

print("2.展示分组名称")

option\_show = input("请输入选项：")

groups = self.get\_all\_groups()

friends = self.get\_all\_friends()

if option\_show == '1':

# 展示所有好友

if len(friends)==0:

print("当前没有任何好友")

else:

print(friends)

elif option\_show == '2':

if len(friends)==0:

print("当前没有任何好友")

else:

if len(groups)==0:

print("当前没有任何分组")

else:

# 展示分组

for dict\_groups in groups:

for group\_name in dict\_groups:

print(group\_name)

is\_show\_group = input("是否展示组内好友：y/n\n")

if is\_show\_group=='y':

show\_group\_name = input("请输入查看的分组名称")

for i in groups:

if show\_group\_name in i:

show\_index = groups.index(i)

print(groups[show\_index][show\_group\_name])

* 分组好友具体代码如下：

def group\_friend(self):

create\_group = input("是否创建新的分组y/n\n")

friends = self.get\_all\_friends()

if create\_group == 'y':

if len(friends)==0:

print("当前没有任何好友")

else:

# 请创建分组

group\_name = input("请输入分组名称：\n")

group\_name\_li = list()

# 展示当前好友

print(friends)

# 移动联系人到哪个组

friend\_name = input("请输入好友名称：\n")

if friend\_name in friends:

all\_friend = []

for friend\_li\_elem in self.friend\_li:

if type(friend\_li\_elem) == dict:

[all\_friend.append(dict\_friends) for

dict\_elem in friend\_li\_elem.values()

for dict\_friends in dict\_elem]

else:

all\_friend.append(friend\_li\_elem)

if friend\_name in all\_friend:

group\_name\_li.append(friend\_name)

self.friend\_li.remove(friend\_name)

# 构建字典： {组名称：分组列表}

friend\_dict = dict()

friend\_dict[group\_name] = group\_name\_li

self.friend\_li.append(friend\_dict)

else:

print("请输入正确的名称")

else:

print('请输入正确好友名称')

elif create\_group =='n':

# 显示当前的分组，将用户添加到指定的组

current\_groups = self.get\_all\_groups()

print('当前分组:')

for current\_group in current\_groups:

for group\_name in current\_group:

print(group\_name)

add\_group = input('请选择添加的组:\n')

# 判断用户输入的组名是否在当前已存在的分组名中

if add\_group in self.get\_all\_groups\_name():

# 添加好友到指定的组

add\_name = input('请选择添加的好友名称：\n')

# 判断用户输入的好友是否存在好友列表中

if add\_name in self.friend\_li:

# 判断用户是否在其他组中

if add\_name not in current\_groups:

# 将好友添加到指定的组内

add\_group\_index = \

self.get\_all\_groups\_name().index(add\_group)

current\_groups[add\_group\_index][

add\_group].append(add\_name)

else:

print('该好友已在其他分组中')

else:

print('请输入正确的组名')

* 启动程序

if \_\_name\_\_ == '\_\_main\_\_':

friend = Friend()

friend.welcome()

1. **代码测试**

运行代码，执行添加好友功能，输出结果如下：

\* \* 欢迎使用好友管理系统 \* \*

1：添加好友

2：删除好友

3：备注好友

4：展示好友

5：好友分组

6：退出

请选择功能

1

请输入要添加的好友：李四

好友添加成功

运行代码，执行好友分组功能，输出结果如下：

请选择功能

5

是否创建新的分组y/n

y

请输入分组名称：

家人

['李四']

请输入好友名称：

李四

运行代码，执行展示好友功能，输出结果如下：

请选择功能

4

1.展示所有好友

2.展示分组名称

请输入选项：1

['李四']

请选择功能

4

1.展示所有好友

2.展示分组名称

请输入选项：2

家人

是否展示组内好友：y/n

y

请输入查看的分组名称家人

['李四']