实验报告#7 - 哈希表的应用

个人信息

姓名：席睿
学号：16340247
班级：软件工程教务三班

开发环境

- 编译器：gcc v6.3.0
- 编辑器：VS Code
- 平台：X86-64
- OS：win10 home

程序说明

运行逻辑

运行程序，弹出字符选单

- 输入s，查看全部学生
- 输入f，按姓名查找学生
- 输入i，往花名册增加学生
- 输入d，删除学生
- 输入q，退出程序

数据结构

```
struct node // 储存学生信息的节点
{
    string name, sex, id, phone;    //学生信息
    node *next;                     //指向下一个节点的指针
    node();                         //默认构造器
    node(string n, string s, string i, string p);     //构造器

};
class Address
{
    node *hashTable[50];            //花名册的哈希表
    void insert_h(string n, string s, string i, string p);  //插入函数的辅助函数
    bool find_h(string name);       //查找函数的辅助函数

  public:
    Address();                      //初始化
    void show();                    //查看全部学生
    void find();                    //按姓名查找学生
    void insert();                  //往花名册增加学生，使用拉链法处理冲突
    void delete_p();                //删除学生
};
```

算法设计

哈希函数

我使用姓名的全部ASACII码之和模50作为哈希函数。

```
int sum = 0;
for (int i = 0; i < n.length(); ++i)
   sum += (int)n[i];                //求ASCII码和
sum %= 50;                          //模50
handle(hashTable[sum]);             //……处理……
```

初始化/构造器

我在初始化时就利用 `insert()` 函数插入了一些学生。

```
for (int i = 0; i < 50; ++i)
      hashTable[i] = NULL;          //清空哈希表
for (int i = 0; i < 20; ++i)
      insert_h(name,sex,id,phone);  //插入一些数据
```

插入

插入算法需要解决两种情况的插入：有冲突的和无冲突的。

对于无冲突的情况来说，只需要将元素放入哈希表对应位置即可。在有冲突的情况下，则需要沿着链表向下插入。

```
void insert_h(string n, string s, string i, string p)
{
	node *new_node = new node(n, s, i, p);
	int sum = hash(n);			//取得哈希值
	if (hashTable[sum] == NULL)	//如果当前哈希值没有冲突，放入此处
		hashTable[sum] = new_node;
	else				//产生冲突，
	{
	node *curr = hashTable[sum];
	while (curr->next != NULL)	//沿单向链表向下查找，
		curr = curr->next;
	curr->next = new_node;		//加入末尾
	}
}
```

删除

删除算法与插入同理。

如果删除的是链表的表头元素，需要考虑表内有没有元素两种情况。

如果删除的是表内元素，则需要在表中查找。

```
int sum = hash(n);			//取得哈希值
if (hashTable[sum]->name == name)	//删除表头元素
{
	node *curr = hashTable[sum];
	if (curr->next == NULL)		//没有表内元素
		hashTable[sum] = NULL;	//表头置NULL
	else				//有表内元素
		hashTable[sum] = hashTable[sum]->next;//用下一个元素代替表头元素即可

}
else					//删除表内元素
{
	node *curr = hashTable[sum];
	while (curr->next != NULL) //查找表内
	{
		if (curr->next->name == name) //找到了
		{
			node *to_be_del = curr->next; //将前一个元素的next指向下一个元素
			curr->next = to_be_del->next;
			delete to_be_del;		//删除当前元素
			break;
		}
		curr = curr->next;		//没找到，继续寻找表的下一个元素
	}
}
```

查找

与删除类似，只是删除操作改成了输出操作。

```
int sum = hash(n);            //取得哈希值
if (hashTable[sum]->name == name) //查找表头
      print(hashTable[sum]);
else
{
      node *curr = hashTable[sum];
      while (curr->next != NULL) //查找表内
      {
            if (curr->next->name == name) //找到了
            {
                  print(curr->next);
                  break;
            }
            curr = curr->next;            //没找到，继续寻找表的下一个元素
      }
}
```

运行结果

```
//菜单
------ Address App -------
[s]how
[f]ind
[i]nsert
[d]elete
[q]uit
------ Address App -------
```

```
//显示全部人员
>>s
--------------- Address -------------------
name            sex             studentID               phone
test4           male            1633450123              131234567890
test5           female          16340123                131234567890
test16          male            16340123                131234567890*
test66          female          16340112                131234567890
deled           female          16340112                131234567890
test89          female          16340123                131234567890
tes181          male            16340123                131234567890
test0           female          16340123                131234567890
test1           male            16340123                131234567890
test12          male            16340123                131234567890*
test2           male            16367sdf83              131234567890
test13          female          16340112                131234567890*
test3           female          1634540123              131234567890
test14          male            1634234123              131234567890*
--------------- end of address -------------------
```

```
//查找成功/失败
>>f
--------- find --------------
name: test14
name            sex             studentID               phone
test14          male            1634234123              131234567890
--------- found --------------
>>f
--------- find --------------
name: null
--------- not found ----------
```

```
//插入成功/空用户名/重名
>>i
--------- insert --------------
name: notempty
sex: male
id: 16230453
phone: 13492247571
--------- insert success -------------
>>f
--------- find --------------
name: notempty
name            sex             studentID               phone
notempty        male            16230453                13492247571
--------- found --------------
>>i
--------- insert --------------
name:
name cannot be empty
--------- insert fail -------------
>>i
--------- insert --------------
name: test4
name            sex             studentID               phone
test4           male            1633450123              131234567890
name used
--------- insert fail -------------
```

```
//删除成功/失败
>>d
--------- delete --------------
name: test4
--------- delete success--------------
>>d
--------- delete --------------
name: test4
--------- delete fail--------------
>>s
--------------- Address -------------------
name            sex             studentID               phone
test5           female          16340123                131234567890
test16          male            16340123                131234567890*
test66          female          16340112                131234567890
deled           female          16340112                131234567890
test89          female          16340123                131234567890
tes181          male            16340123                131234567890
test0           female          16340123                131234567890
notempty        male            16230453                13492247571*
test1           male            16340123                131234567890
test12          male            16340123                131234567890*
test2           male            16367sdf83              131234567890
test13          female          16340112                131234567890*
test3           female          1634540123              131234567890
test14          male            1634234123              131234567890*
--------------- end of address -------------------
>>q
```