

实 验 报 告

## 实验一《动态链表的设计与应用》

**学生姓名 李广滨**

**学号 176003246**

**课程名称 数据结构与算法**

**实验学期 2018 至 2019 学年 第 二 学期**

**所在学院 信息科学与工程学院**

**专业班级**

**任课教师**

**实验成绩**

**一、实验题目**

动态链表的设计与应用

**二、实验目的：**

1、掌握单链表的基本操作：插入、删除、查找等运算。

2、掌握单向链表的存储特点及其实现。

**三、实验设备与环境**

微型计算机、Windows 系列操作系统 、Visual Studio系列软件

**四、实验内容**

利用单向循环链表实现一个班级学生信息管理（数据录入、插入、删除、排序、查找等）。

**五、概要设计（思路、算法、步骤等）**

将学生信息保存在一个txt文件中，软件可以通过导入此txt文件导入学生信息及其照片文件地址。读入txt文件时将学生信息以链表方式储存。同时用户可以在软件界面对链表信息进行添，删，改，查等操作。修改完毕后，可以将信息再以txt文件格式保存。

代码已打包成exe，建议在与image文件夹同目录下打开exe文件，不然可能会出现文件显示不出的情况。

**六、详细设计（核心代码、算法流程图等）**

**数据结构（链表）部分代码：**

class Student:

def \_\_init\_\_(self,ID,name,sex,age,image):

self.ID=ID

self.name=name

self.sex=sex

self.age=age

self.image=image

class Node(object):

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

self.data=data

self.next=None

class LinkedList(object):

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

self.head=None

def clear(self):

self.head=None

def len(self):

ans=0

cur\_node=self.head

while cur\_node is not None:

ans+=1

cur\_node=cur\_node.next

return ans

def init\_list(self,list):

self.head=Node(list[0])

cur\_node=self.head

for i in range (1,len(list)):

nex\_node=Node(list[i])

cur\_node.next=nex\_node

cur\_node=cur\_node.next

def add(self,student):

add\_node=Node(student)

if self.head==None:

self.head=add\_node

else:

iterator=self.head

while iterator.next!=None:

iterator=iterator.next

iterator.next=add\_node

def remove(self,key):

cur\_node=self.head

pre\_node=self.head

while cur\_node!=None:

if key==cur\_node.data.ID or key==cur\_node.data.name:

pre\_node.next=cur\_node.next

return

else:

pre\_node=cur\_node

cur\_node=cur\_node.next

def change(self,student):

iterator=self.head

while iterator!=None:

if student.ID==iterator.data.ID or student.name==iterator.data.name:

iterator.data.ID=student.ID

iterator.data.name=student.name

iterator.data.sex=student.sex

iterator.data.age=student.age

iterator.data.image=student.image

return True

else:

iterator=iterator.next

return False

def traverse(self):

iter\_list=[]

iterator=self.head

while iterator is not None:

iter\_list.append(iterator.data)

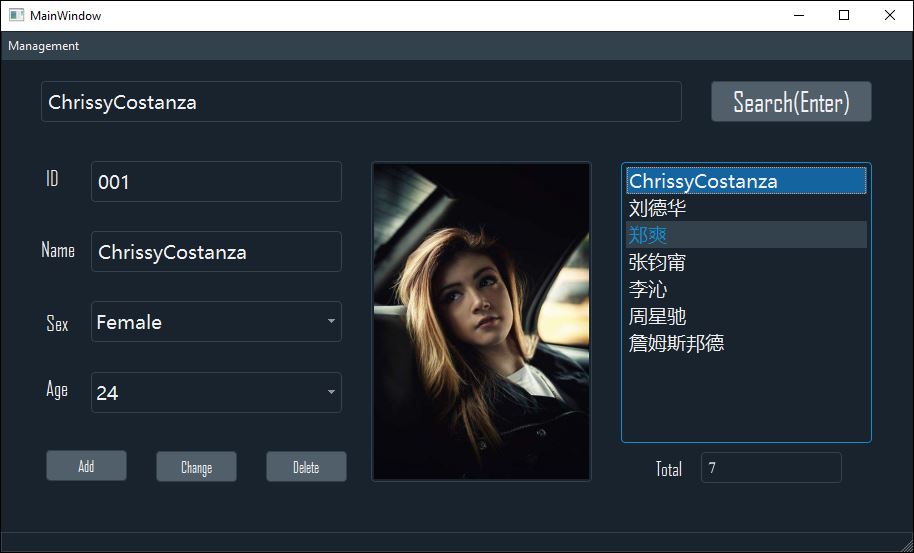
iterator=iterator.next

return iter\_list

**七、测试结果及分析**

## import(导入功能)

### Management中import功能支持导入txt文件



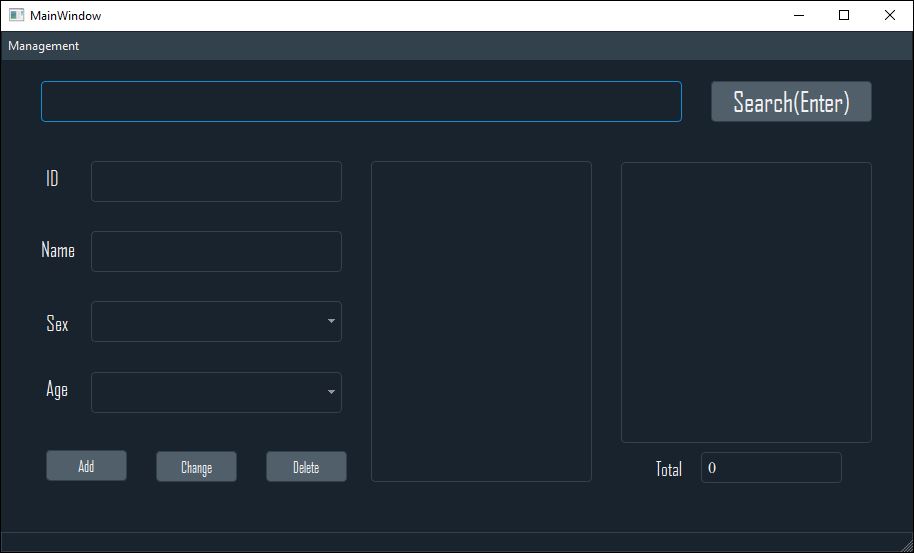
## export(导出功能)

### Management中export功能支持导出txt文件，修改完或创建完信息后可以再以txt文件形式保存

**样例:略**

## Shutdown(关闭当前数据库)

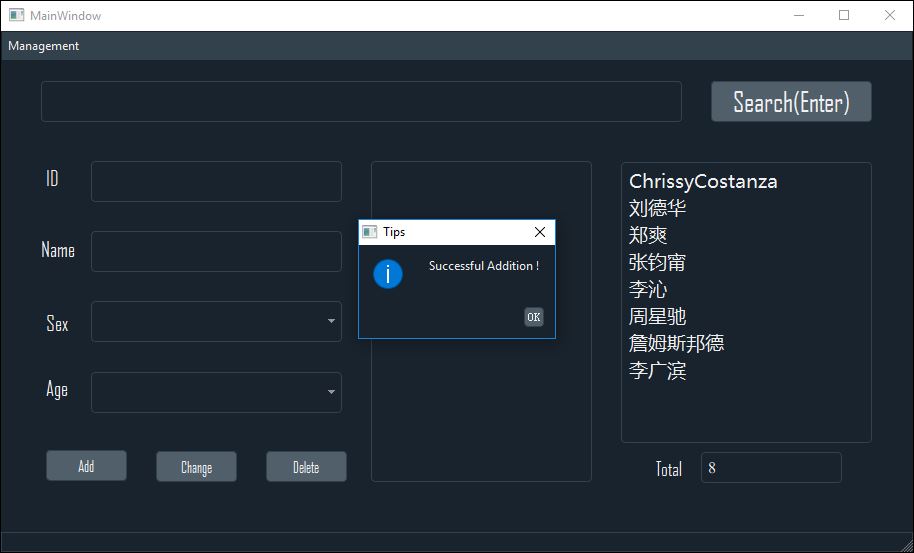
### Management中Shutdown功能支持清空当前软件中的信息，即初始化



## Quit(退出功能)

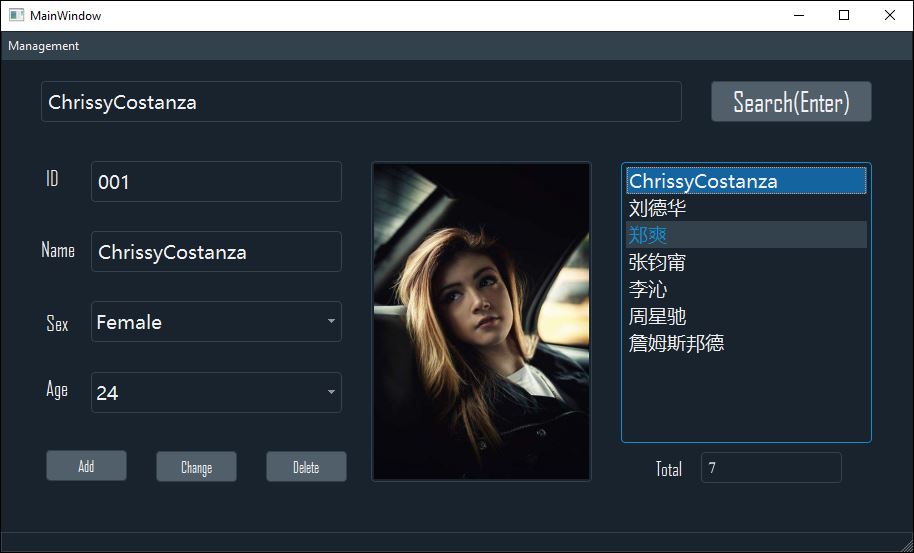
### Management中Shutdown功能可以退出软件

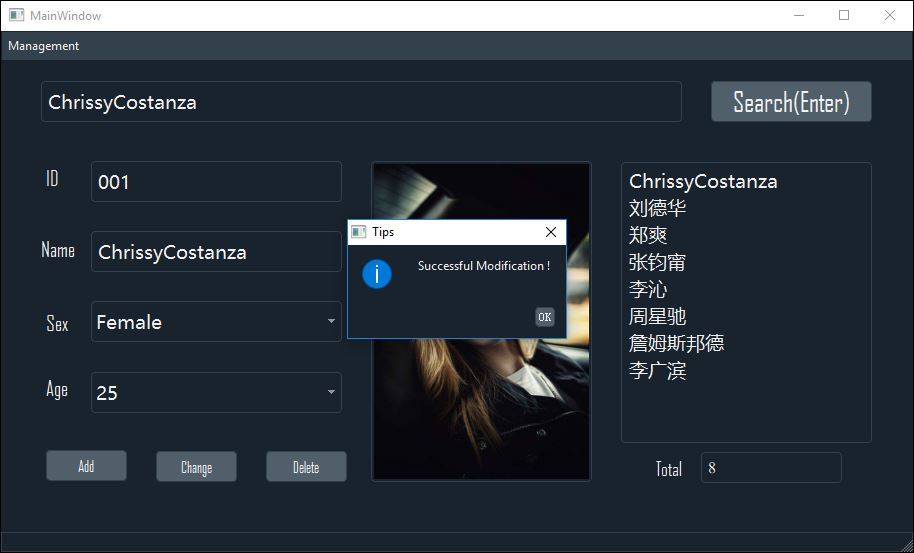
## Add(添加功能)



## Change(修改功能)

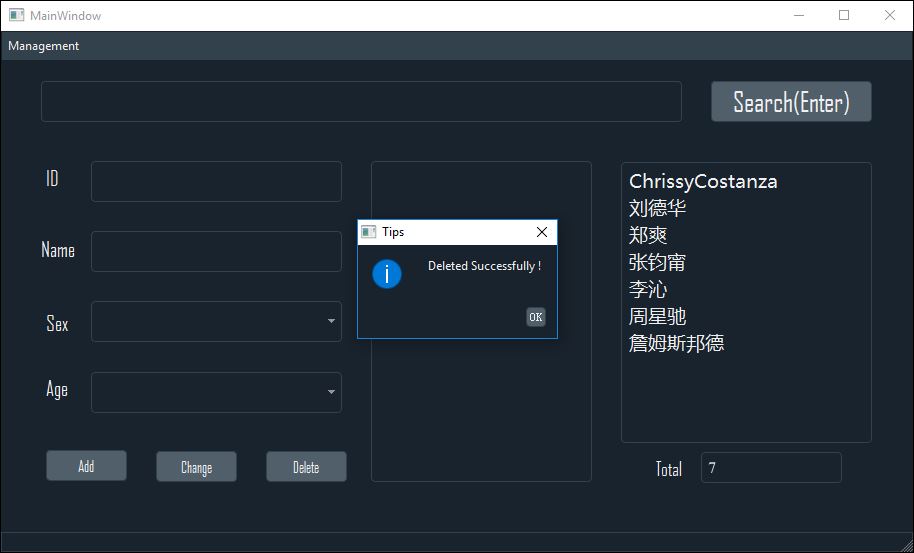
### 修改年龄 24->25





## Delete(删除功能)

### 删除李广滨同学的信息



## Search(查询功能)

### 支持姓名和ID查询，支持回车快捷键



## Total(实时统计人数)

### 软件右下角

照片导入:将照片命名为Name+”.jpg”放入image文件夹即可。

**八、总结**

利用PyQt5完成此次实验，并用Pyinstaller打包成exe文件。在学习数据结构的同时学会了PC端应用的开发流程。深刻意识到在写代码前先规划好代码结构的重要性。同时理解了链表这个数据结构在插入，删除操作上性能的优势。