# 基本介绍

本程序使用python可视化图形库Tkinter并配合json文件来储存学生基本信息，并可以通过桌面程序点击按钮输入等，对学生的信息进行，补充，显示，查询，修改等功能。实用性较强。因为时间原因，所以数据中的元素较少，有4个元素。以下为一些文件的介绍

## Mainpage.py

用来显示开始的主页面，通过pack与pack\_forget函数来进行页面的刷新更替功能。

## views.py

主要是各个分页面的显示，相当于是图形库的前端，并且同时跟后端bd.py发生联系，不断传输数据来进行功能的实现

## db.py

主要是数据的写入和读取等，相当于网页的后端，与json文件数据库进行不断读取和写入，同时和页面中传入的数据进行查找或者计算，删除。

# stus.json

存储学生数据,开始自带存储了两组数据

# 各个模块代码

## Mainpage.py

|  |
| --- |
| import tkinter as tk  from views import aboutframe  from views import changeframe  from views import insertframe  from views import searchframe  from views import deletframe  class mainpage:  def \_\_init\_\_(self,master:tk.Tk):  self.root=master  self.root.title("小辉的学生管理系统")  self.root.geometry('600x400')  self.creat\_page()  def creat\_page(self):  self.insert\_frame = insertframe(self.root)  self.search\_frame = searchframe(self.root)  self.delete\_frame = deletframe(self.root)  self.change\_frame = changeframe(self.root)  self.about\_frame=aboutframe(self.root)  menubar=tk.Menu(self.root)  menubar.add\_command(label="录入",command=self.show\_insert)  menubar.add\_command(label="查询",command=self.show\_seacher)  menubar.add\_command(label="删除",command=self.show\_delete)  menubar.add\_command(label="修改",command=self.show\_change)  menubar.add\_command(label="关于",command=self.show\_about)  self.root['menu']=menubar  def show\_insert(self):#插入  self.insert\_frame.pack()  self.search\_frame.pack\_forget()  self.delete\_frame.pack\_forget()  self.change\_frame.pack\_forget()  self.about\_frame.pack\_forget()  def show\_seacher(self):#展示  self.insert\_frame.pack\_forget()  self.search\_frame.pack()  self.delete\_frame.pack\_forget()  self.change\_frame.pack\_forget()  self.about\_frame.pack\_forget()  def show\_delete(self):#删除  self.insert\_frame.pack\_forget()  self.search\_frame.pack\_forget()  self.delete\_frame.pack()  self.change\_frame.pack\_forget()  self.about\_frame.pack\_forget()  def show\_change(self):#修改  self.insert\_frame.pack\_forget()  self.search\_frame.pack\_forget()  self.delete\_frame.pack\_forget()  self.change\_frame.pack()  self.about\_frame.pack\_forget()  def show\_about(self):#关于  self.insert\_frame.pack\_forget()  self.search\_frame.pack\_forget()  self.delete\_frame.pack\_forget()  self.change\_frame.pack\_forget()  self.about\_frame.pack()  if \_\_name\_\_ == '\_\_main\_\_':  root=tk.Tk()  mainpage(root)  root.mainloop() |

## View.py

|  |
| --- |
| import tkinter as tk from tkinter import ttk from db import db class aboutframe(tk.Frame):#关于页面  def \_\_init\_\_(self,root):  super().\_\_init\_\_(root)  tk.Label(self, text='本程序由nwpu小辉制作').pack()  tk.Label(self, text='因为刚开始学图形库，所以有些错误，请老师多多包涵').pack()  class changeframe(tk.Frame):#修改功能页面  def \_\_init\_\_(self,root):#初始化  super().\_\_init\_\_(root)  self.name = tk.StringVar()  self.sex = tk.StringVar()  self.age = tk.StringVar()  self.id= tk.StringVar()  self.status = tk.StringVar()  self.creat\_page()   def creat\_page(self):#显示修改功能时的图形  tk.Label(self).grid(row=0, pady=10)  tk.Label(self, text="姓名：").grid(row=1, column=1, pady=10)  tk.Entry(self, textvariable=self.name).grid(row=1, column=2, pady=10)   tk.Label(self, text="性别：").grid(row=2, column=1, pady=10)  tk.Entry(self, textvariable=self.sex).grid(row=2, column=2, pady=10)   tk.Label(self, text="年龄：").grid(row=3, column=1, pady=10)  tk.Entry(self, textvariable=self.age).grid(row=3, column=2, pady=10)   tk.Label(self, text="学号：").grid(row=4, column=1, pady=10)  tk.Entry(self, textvariable=self.id).grid(row=4, column=2, pady=10)  tk.Button(self, text="修改", command=self.change\_user).grid(row=5, column=1, pady=10)  tk.Button(self, text="查询", command=self.seacher\_user).grid(row=5, column=2, pady=10)  tk.Label(self, textvariable=self.status).grid(row=6, column=2, pady=10, sticky=tk.E)  def seacher\_user(self):#查找函数，通过姓名   flag,info=db.seacher\_by\_username(self.name.get())  if flag:  self.name.set(info['name'])  self.sex.set(info['sex'])  self.age.set(info['age'])  self.id.set(info['id'])  self.status.set("数据查询成功")  else:  self.status.set(info)  pass  def change\_user(self):#修改数据功能  stu = {"name": self.name.get(), "sex": self.sex.get(),  "age": self.age.get(), "id": self.id.get()}  print(stu)  self.name.set('')  self.sex.set('')  self.age.set('')  self.id.set('')  print(stu)  db.update(stu)  self.status.set("修改数据成功")  pass  class insertframe(tk.Frame):#插入数据功能  def \_\_init\_\_(self,root):  super().\_\_init\_\_(root)  # tk.Label(self, text='插入页面').pack()  self.name=tk.StringVar()  self.sex = tk.StringVar()  self.age = tk.StringVar()  self.id= tk.StringVar()  self.status = tk.StringVar()  self.creat\_page()  def creat\_page(self):#插入数据页面展示  tk.Label(self).grid(row=0,pady=10)  tk.Label(self,text="姓名：").grid(row=1,column=1, pady=10)  tk.Entry(self,textvariable=self.name).grid(row=1,column=2, pady=10)   tk.Label(self, text="性别：").grid(row=2, column=1, pady=10)  tk.Entry(self, textvariable=self.sex).grid(row=2, column=2, pady=10)   tk.Label(self, text="年龄：").grid(row=3, column=1, pady=10)  tk.Entry(self, textvariable=self.age).grid(row=3, column=2, pady=10)   tk.Label(self, text="学号：").grid(row=4, column=1, pady=10)  tk.Entry(self, textvariable=self.id).grid(row=4, column=2, pady=10)  tk.Button(self,text="录入",command=self.recode\_info).grid(row=5,column=2,pady=10)  tk.Label(self,textvariable=self.status).grid(row=6,column=2,pady=10,sticky=tk.E)  def recode\_info(self):  stu={"name":self.name.get(),"sex":self.sex.get(),  "age":self.age.get(),"id":self.id.get()}  print(stu)  self.name.set('')  self.sex.set('')  self.age.set('')  self.id.set('')   db.insert(stu)  self.status.set("插入数据成功") class searchframe(tk.Frame):#查找功能  def \_\_init\_\_(self,root):  super().\_\_init\_\_(root)   self.table\_view =tk.Frame()  self.table\_view.pack()  self.create\_page()  def create\_page(self):#查找页面显示  column=("name","sex","age","id")  columns\_values=("姓名","性别","年龄","学号")  self.tree\_view = ttk.Treeview(self, show='headings' , columns=column)  self.tree\_view.column('name', width=80, anchor='center')  self.tree\_view.column('sex', width=80, anchor='center')  self.tree\_view.column('age', width=80, anchor='center')  self.tree\_view.column('id', width=80, anchor='center')  self.tree\_view.heading('name',text='姓名')  self.tree\_view.heading('sex', text='性别')  self.tree\_view.heading('age', text='年龄')  self.tree\_view.heading('id', text='学号')  self.tree\_view.pack(fill=tk.BOTH,expand=True)  self.show\_date\_frame()   tk.Button(self,text='刷新数据',command=self.show\_date\_frame).pack(anchor=tk.E,pady=5)  def show\_date\_frame(self):  #删除旧的阶段  for \_ in map(self.tree\_view.delete,self.tree\_view.get\_children('')):  pass  students=db.all()  index=0  for stu in students:  print(stu)  self.tree\_view.insert('',index+1,values=(stu['name'],stu["sex"],stu['age'],stu['id'])) class deletframe(tk.Frame):#删除数据页面  def \_\_init\_\_(self,root):  super().\_\_init\_\_(root)  self.username=tk.StringVar()  self.status = tk.StringVar()  tk.Label(self, text='根据名称删除数据').pack()  tk.Entry(self, textvariable=self.username).pack()  tk.Button(self, text='删除',command=self.delete).pack()  tk.Label(self, textvariable=self.status).pack()  def delete(self):#删除数据函数  username=self.username.get()  flag,message= db.delete\_by\_username(username)  self.status.set(message)  pass |

## db.py

|  |
| --- |
| import json  class MysqlDatebases:#数据  def \_\_init\_\_(self):  with open('user.json',mode='r',encoding='utf-8') as f:  text=f.read()  self.users=json.loads(text)  self.students=json.loads(open('stus.json',mode='r',encoding='utf-8').read())   def check\_login(self,username,password):   for user in self.users:  if username==user['username']:  if password ==user['password']:  return True,'登录成功'  else:  return False,"密码错误"  return False,"登录失败，用户不存在"  # return self.users  def all(self):  return self.students   def insert(self,student):  self.students.append(student)  def delete\_by\_username(self,name):  for student in self.students:  print(student)  if student['name']==name:  self.students.remove(student)  return True,f'{name}删除用户成功'  return False,f'{name}用户不存在'   def seacher\_by\_username(self, name):  for student in self.students:  print(student)  if student['name'] == name:   print(student)  return True, student   return False, f'{name}用户不存在'  def update(self,stu):  for student in self.students:  print(student)  if student['name'] == stu['name']:  student.update(stu)  print(stu['name'])  return True,f'{stu["name"]}用户数据修改成功'   return False, f'{stu["name"]}用户不存在' db=MysqlDatebases() if \_\_name\_\_ == '\_\_main\_\_':  print(db.check\_login('admin','123456'))  print(db.all())  print(db.seacher\_by\_username('1')) |

## stus.json

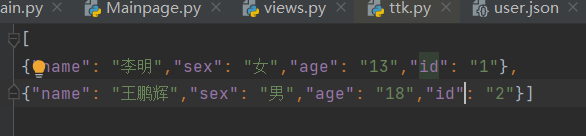
|  |
| --- |
| [ {"name": "李明","sex": "女","age": "13","id": "1"}, {"name": "王鹏辉","sex": "男","age": "18","id": "2"}] |

# 运行截图

1. 可以在表面各个栏目选择所需要进行的功能



1. 录入功能：开始json文件中只存在2组数据





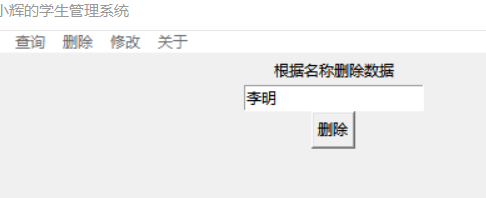


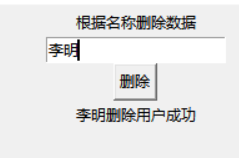
1. 查询功能:在录入数据后在查询页面点击刷新数据可以看到目前的数据

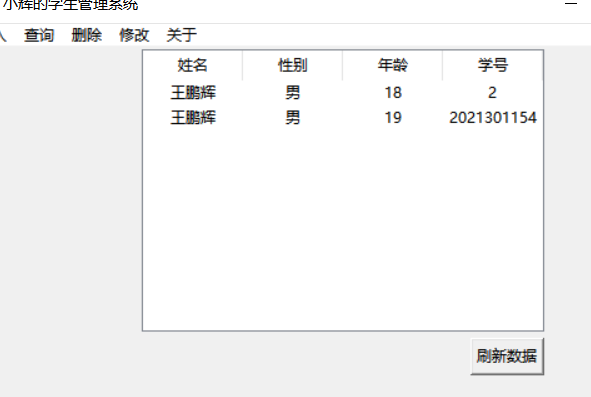


1. 删除数据

输入李明，并点击删除







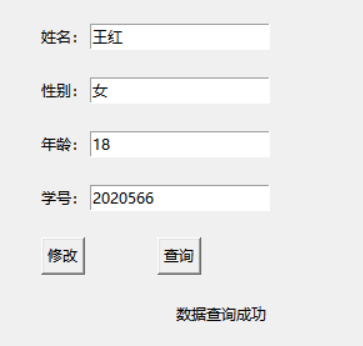
1. 查询数据

先插入一组数据





在修改页面只输入姓名点击查询



将会自动在剩余三个空位补充数据

如果查询李红（数据库中不存在的人名），则会显示用户不存在



1. 修改数据

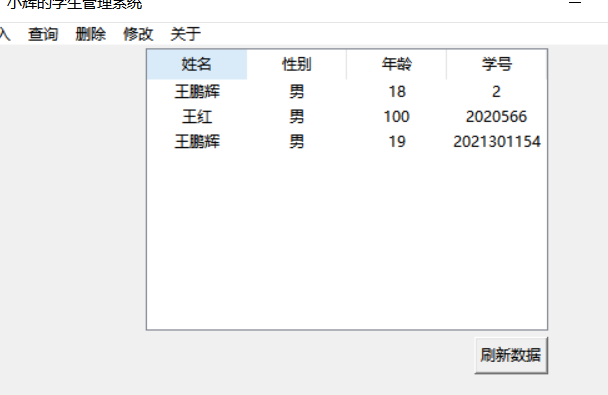
先输入王红点击查询



然后在此上直接进行修改



点击修改即可修改



由此可见数据库数据已经被修改

1. 关于介绍页面

可以在关于页面添加一些其他的东西

