**python学生成绩管理系统-增删查改**

1.1问题描述及要求

问题描述：

用Python编写一个学术成绩管理系统，功能包括但不限于以下：

1.学生注册登录

2.学生成绩录入

3.个人成绩查询

4.统计个人成绩

5.统计班级成绩

要求：

用csv文件保存个学生信息，可以包括姓名、学号、成绩(至少包括三门课程)

成绩查询支持根据姓名或学号查询

统计个人成绩包括总分及班级排名

其余部分自由发挥

1.2实验内容

1.2.1设计思路

目的

在大二上学期“Python”的课程中，我们学习掌握了python程序设计的基本操作。python语言作为一门强大的的编程语言，其中的编程技巧绝不是作为学生的我们在一个学期内可以完全掌握的。为了更好的掌握python语言编程，理解编程思想，我们以此课程设计为契机，结合生活中的需求，设计了一个学生成绩管理系统。

思路设计：

1.通过列表的知识实现学生成绩信息（学号、姓名、院系、四门课成绩、总分等）信息处理与操作；

2.能够实现学生成绩信息的保存和读取（使用数据库对数据进行存取）；

3.实现所有相关信息的输入、输出、查找、删除、修改等功能；

4.系统界面应至少实现控制台界面（使用桌面窗体界面进行交互）；用tinkter搭建GUI交互平台。

5.登录注册，使用账号密码（小写字母）登录注册

6.运行程序应该预先下载matplotlib、pandas模块用于数据分析

7.程序模块化设计与归类

8.通过csv文件进行保存；需要注意的是，本程序采用“utf-8”编码，因此采用WPS进行操作csv时应该注意编码问题，避免内容丢失。

功能描述

（1）添加功能：程序能够添加不同学生的记录，要求姓名学号要唯一，如果添加了重复学号的记录时，则提示数据添加重复并取消添加。

（2）查询功能：可根据学号、姓名信息对已添加的学生记录进行查询，如果未找到，给出相应的提示信息，如果找到，则显示相应的记录信息。

（3）显示功能：可显示当前系统中所有学生的记录，每条记录占据一行。

（4）修改功能：可根据查询结果对相应的记录进行修改，修改时注意学号的唯一性。

（5）删除功能：主要实现对已添加的学生记录进行删除。如果当前系统中没有相应的记录，则提示“记录为空！”并返回操作

（6）统计功能：主要实现对已添加的学生记录进行统计。直观化地用直方图显示各种信息。

（7）排名与计算平均数

（8）登录注册功能：输入账号密码（小写字母），进行登录注册

参考资料

嵩天等. python语言程序设计基础[M]. 北京: 高等教育出版社,2014

编写规范

1. 命名规则

变量命名，尽量使用英文表达变量的准确定义，部分变量可使用拼音解释含义。

2. 注释

函数功能都要在函数原型后，部分测试者难以理解的而算法和流程应该给出相应的注释。

编程环境

pycharm和windows下的python3.9 64bit，在python3.9 64bit的IDLE下运行

代码：LoginPage.py

from tkinter import \*

import tkinter.messagebox as messagebox

import tkinter.font as tkFont

from view import \* #菜单栏对应的各个子页面

class denglu(object):

def \_\_init\_\_(self, master=None):

self.root = master #定义内部变量root

self.root.geometry('%dx%d' % (480,320)) #设置窗口大小

self.username = StringVar()

self.password = StringVar()

self.createPage()

def createPage(self):

self.page = Frame(self.root) #创建Frame

self.page.pack()

Label(self.page, text="学生成绩管理系统", font=("宋体", 20)).grid(row=0, pady=40, stick=W)

Label(self.page, text = '账户:', font=tkFont.Font(size=12)).grid(row=1, stick=W, pady=10)

Entry(self.page, textvariable=self.username).grid(row=1, stick=E)

Label(self.page, text = '密码:', font=tkFont.Font(size=12)).grid(row=2, stick=W, pady=10)

Entry(self.page, textvariable=self.password, show='\*').grid(row=2, stick=E)

Button(self.page, text='登录', command=self.dengluCheck, font=tkFont.Font(size=12)).grid(row=3, stick=W,pady=30)

Button(self.page, text='注册', command=self.register, font=tkFont.Font(size=12)).grid(row=3,stick=E, pady=30)

def dengluCheck(self):

name = self.username.get()

password = self.password.get()

if self.isLegalUser(name,password):

self.page.destroy()

MainPage(self.root)

else:

messagebox.showinfo(title='错误', message='账号或密码错误！')

def isLegal(self,string):

alp = ['a','b','c','d','e','f','g','h','i','j','k','l','m','n','o','p','q','r','s','t','u','v','w','x','y','z']

for i in string:

if i in alp:

pass

else:

return False

return True

def isLegalUser(self,name,password):

f = open('账号密码.csv','r',encoding='utf-8')

for line in f.readlines():

info = line[:-1].split(",")

if len(info)<2:

break

if info[0].strip()==name and info[1].strip()==password :

f.close()

return True

return False

def register(self):

name = self.username.get()

password = self.password.get()

if len(name)==0 or len(password)==0:

messagebox.showinfo(title='错误', message='账号密码不能为空')

return

for i in password:

if i ==',' or i == ' ':

messagebox.showinfo(title='错误', message='密码不能含有非法字符')

return

if self.isLegal(name):

pass

else:

messagebox.showinfo(title='错误', message='账号不能含有非法字符')

return

f = open('账号密码.csv','r',encoding='utf-8')

for line in f.readlines():

info = line[:-1].split(",")

if len(info)<2:

break

if info[0].strip()==name:

messagebox.showinfo(title='结果', message ="已存在该用户信息！")

f.close()

return

f.close()

f = open('账号密码.csv','a',encoding='utf-8')

f.write('{},{}\n'.format(name,password))

f.close()

messagebox.showinfo(title='提示', message ="注册成功")

class MainPage(object):

def \_\_init\_\_(self, master=None):

self.root = master # 定义内部变量root

self.root.geometry('%dx%d' % (600, 400)) # 设置窗口大小

self.createPage()

def createPage(self):

self.tianjiaPage = tianjiaFrame(self.root) # 创建不同Frame

self.deletePage = DeleteFrame(self.root)

self.xiugaiPage = xiugaiFrame(self.root)

self.searchPage = searchFrame(self.root)

self.countPage = CountFrame(self.root)

self.tianjiaPage.pack() # 默认显示数据录入界面

menubar = Menu(self.root)

menubar.add\_command(label='添加', command=self.tianjiaData)

menubar.add\_command(label='删除', command=self.deleteData)

menubar.add\_command(label='修改', command=self.xiugaiData)

menubar.add\_command(label='查找', command=self.searchData)

menubar.add\_command(label='个人统计', command=self.overlookData)

menubar.add\_command(label='班级统计', command=self.countData)

self.root['menu'] = menubar # 设置菜单栏

def tianjiaData(self):

self.tianjiaPage.pack()

self.searchPage.pack\_forget()

self.deletePage.pack\_forget()

self.xiugaiPage.pack\_forget()

self.countPage.pack\_forget()

def deleteData(self):

self.tianjiaPage.pack\_forget()

self.searchPage.pack\_forget()

self.deletePage.pack()

self.xiugaiPage.pack\_forget()

self.countPage.pack\_forget()

def xiugaiData(self):

self.tianjiaPage.pack\_forget()

self.searchPage.pack\_forget()

self.deletePage.pack\_forget()

self.xiugaiPage.pack()

self.countPage.pack\_forget()

def searchData(self):

self.tianjiaPage.pack\_forget()

self.searchPage.pack()

self.deletePage.pack\_forget()

self.xiugaiPage.pack\_forget()

self.countPage.pack\_forget()

def overlookData(self):

OverlookFrame(self.root)

def countData(self):

self.tianjiaPage.pack\_forget()

self.searchPage.pack\_forget()

self.deletePage.pack\_forget()

self.xiugaiPage.pack\_forget()

self.countPage.pack()

main.py

from tkinter import \*

from LoginPage import \*

root = Tk()

root.title('学生成绩管理系统')

denglu(root)

root.mainloop()

view.py

from tkinter import \*

from tkinter import ttk

import matplotlib.pyplot as plt

import pandas as pd

import LoginPage as mmp

import tkinter.font as tkFont

import tkinter.messagebox as messagebox

class tianjiaFrame(Frame): # 继承Frame类

def \_\_init\_\_(self, master=None):

Frame.\_\_init\_\_(self, master)

self.root = master #定义内部变量root

self.E1 = Entry(self)

self.E2 = Entry(self)

self.E3 = Entry(self)

self.E4 = Entry(self)

self.E5 = Entry(self)

self.E6 = Entry(self)

self.createPage()

def Isspace(self,text):

temp = 0

for i in text:

if not i.isspace():

temp = 1

break;

if temp==1:

return 0

else:

return 1

def write(self,name,num,chinese,math,english,physics):

f = open('成绩.csv','r',encoding='utf-8')

for line in f.readlines():

info = line[:-1].split(",")

if len(info)<4:

break

if info[0] ==name and info[1] ==num:

messagebox.showinfo(title='结果', message ="已存在该学生科目信息！")

f.close()

return

f.close()

f = open('成绩.csv','a',encoding='utf-8')

f.write('{},{},{},{},{},{},{}\n'.format(name,num,chinese,math,english,physics,int(chinese)+int(math)+int(english)+int(physics)))

f.close()

messagebox.showinfo(title='提示', message ="写入成功")

return

def click(self):

name = self.E1.get()

num = self.E2.get()

chinese = self.E3.get()

math = self.E4.get()

english=self.E5.get()

physics=self.E6.get()

if self.Isspace(name) or self.Isspace(num) or self.Isspace(chinese) or self.Isspace(math) or self.Isspace(english) or self.Isspace(physics):

messagebox.showinfo(title='提示', message ="输入项为空")

else:

self.write(name,num,chinese,math,english,physics)

def createPage(self):

Label(self).grid(row=0, stick=W, pady=10)

Label(self, text = '姓名: ').grid(row=1, stick=W, pady=10)

self.E1.grid(row=1, column=1, stick=E)

Label(self, text = '学号: ').grid(row=2, stick=W, pady=10)

self.E2.grid(row=2, column=1, stick=E)

Label(self, text = '语文: ').grid(row=3, stick=W, pady=10)

self.E3.grid(row=3, column=1, stick=E)

Label(self, text = '数学: ').grid(row=4, stick=W, pady=10)

self.E4.grid(row=4, column=1, stick=E)

Label(self, text='英语: ').grid(row=5, stick=W, pady=10)

self.E5.grid(row=5, column=1, stick=E)

Label(self, text='物理: ').grid(row=6, stick=W, pady=10)

self.E6.grid(row=6, column=1, stick=E)

Button(self, text='录入',command=self.click).grid(row=7, column=1, stick=E, pady=10)

class DeleteFrame(Frame): # 继承Frame类

def \_\_init\_\_(self, master=None):

Frame.\_\_init\_\_(self, master)

self.root = master #定义内部变量root

self.E1 = Entry(self)

self.E2 = Entry(self)

self.createPage()

def Isspace(self,text):

temp = 0

for i in text:

if not i.isspace():

temp = 1

break;

if temp==1:

return 0

else:

return 1

def delete(self,name,num):

temp = 0

with open("成绩.csv","r",encoding="utf-8") as f:

lines = f.readlines()

with open("成绩.csv","w",encoding="utf-8") as f\_w:

for line in lines:

info = line[:-1].split(",")

if info[0] ==name and info[1] ==num:

temp = 1

continue

f\_w.write(line)

if temp==0:

messagebox.showinfo(title='提示', message ="没有该信息")

else:

messagebox.showinfo(title='提示', message ="删除成功")

def click(self):

name= self.E1.get()

num = self.E2.get()

if self.Isspace(name) or self.Isspace(num):

messagebox.showinfo(title='提示', message ="输入项为空")

else:

self.delete(name,num)

def createPage(self):

Label(self).grid(row=0, stick=W, pady=10)

Label(self, text = '姓名: ').grid(row=1, stick=W, pady=10)

self.E1.grid(row=1, column=1, stick=E)

Label(self, text = '学号: ').grid(row=2, stick=W, pady=10)

self.E2.grid(row=2, column=1, stick=E)

Button(self, text='删除',command=self.click).grid(row=6, column=1, stick=E, pady=10)

class xiugaiFrame(Frame): # 继承Frame类

def \_\_init\_\_(self, master=None):

Frame.\_\_init\_\_(self, master)

self.root = master #定义内部变量root

self.E1 = Entry(self)

self.E2 = Entry(self)

self.E3 = Entry(self)

self.E4 = Entry(self)

self.E5 = Entry(self)

self.E6 = Entry(self)

self.createPage()

def Isspace(self,text):

temp = 0

for i in text:

if not i.isspace():

temp = 1

break;

if temp==1:

return 0

else:

return 1

def xiugai(self,name,num,chinese,english,math,physics):

temp = 0

with open("成绩.csv","r",encoding="utf-8") as f:

lines = f.readlines()

with open("成绩.csv","w",encoding="utf-8") as f\_w:

for line in lines:

info = line[:-1].split(",")

if info[0] ==name and info[1] ==num:

temp = 1

f\_w.write('{},{},{},{},{},{},{}\n'.format(name,num,chinese,math,english,physics,int(chinese)+int(math)+int(english)+int(physics)))

continue

f\_w.write(line)

if temp==0:

messagebox.showinfo(title='提示', message ="没有该信息")

else:

messagebox.showinfo(title='提示', message ="修改成功")

def click(self):

name = self.E1.get()

num = self.E2.get()

chinese = self.E3.get()

math = self.E4.get()

english = self.E5.get()

physics = self.E6.get()

if self.Isspace(name) or self.Isspace(num) or self.Isspace(chinese) or self.Isspace(math) or self.Isspace(english) or self.Isspace(physics):

messagebox.showinfo(title='提示', message ="输入项为空")

else:

self.xiugai(name,num,chinese,math,english,physics)

def createPage(self):

Label(self).grid(row=0, stick=W, pady=10)

Label(self, text = '姓名: ').grid(row=1, stick=W, pady=10)

self.E1.grid(row=1, column=1, stick=E)

Label(self, text = '学号: ').grid(row=2, stick=W, pady=10)

self.E2.grid(row=2, column=1, stick=E)

Label(self, text = '语文: ').grid(row=3, stick=W, pady=10)

self.E3.grid(row=3, column=1, stick=E)

Label(self, text = '数学: ').grid(row=4, stick=W, pady=10)

self.E4.grid(row=4, column=1, stick=E)

Label(self, text='英语: ').grid(row=5, stick=W, pady=10)

self.E5.grid(row=5, column=1, stick=E)

Label(self, text='物理: ').grid(row=6, stick=W, pady=10)

self.E6.grid(row=6, column=1, stick=E)

Button(self, text='修改',command=self.click).grid(row=7, column=1, stick=E, pady=10)

class searchFrame(Frame): # 继承Frame类 查找

def \_\_init\_\_(self, master=None):

Frame.\_\_init\_\_(self, master)

self.root = master #定义内部变量root

self.E1 = Entry(self)

self.E2 = Entry(self)

self.createPage()

def Isspace(self,text):

temp = 0

for i in text:

if not i.isspace():

temp = 1

break;

if temp==1:

return 0

else:

return 1

def paixu(self, arr, n):

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

for j in range(0, len(arr) - i):

if int(arr[j][n]) < int(arr[j + 1][n]):

arr[j], arr[j + 1] = arr[j + 1], arr[j]

return arr

def search(self,name,num):

f = open('成绩.csv','r',encoding='utf-8')

lls = []

i = 0

for line in f:

line = line.replace("\n", "")

lls.append(line.split(","))

ls = lls[:0]

ls.extend(lls[1:])

ls = self.paixu(ls, 6)

for i in range(len(ls)):

if ls[i][0]==name and ls[i][1]==num:

messagebox.showinfo(title='结果', message ="姓名:"+ls[i][0] +" 学号:"+ls[i][1] +"\n语文:"+ls[i][2] +" 数学:"+ls[i][3]+"\n英语:"+ls[i][4]+" 物理:"+ls[i][5]+"\n总分"+ls[i][6]+" 排名"+str(i+1))

f.close()

return

messagebox.showinfo(title='提示', message ="没有该信息")

f.close()

return

def click(self):

name = self.E1.get()

num = self.E2.get()

if self.Isspace(name) or self.Isspace(num):

messagebox.showinfo(title='提示', message ="输入项为空")

else:

self.search(name,num)

def createPage(self):

Label(self).grid(row=0, stick=W, pady=10)

Label(self, text = '姓名: ').grid(row=1, stick=W, pady=10)

self.E1.grid(row=1, column=1, stick=E)

Label(self, text = '学号: ').grid(row=2, stick=W, pady=10)

self.E2.grid(row=2, column=1, stick=E)

Button(self, text='查找',command=self.click).grid(row=6, column=1, stick=E, pady=10)

class OverlookFrame(Frame): # 继承Frame类

def score\_rank(self,ls, n, score):

rank = 1

for i in range(len(ls)):

if int(ls[i][n])> int(score):

rank += 1

return str(rank)

def click(self):

self.window.destroy()

def paixu(self,arr,n):

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

for j in range(0, len(arr) - i):

if int(arr[j][n]) < int(arr[j + 1][n]):

arr[j], arr[j + 1] = arr[j + 1], arr[j]

return arr

def \_\_init\_\_(self, parent\_window):

# parent\_window.destroy()

self.window = Tk()

self.window.title("学生成绩管理系统")

self.window.geometry("600x400")

self.window.resizable(0, 0)

Label(self.window, text="统计个人成绩", font=("宋体", 20)).pack(pady=20)

columns = ("语文", "数学", "英语","物理","总分")

tree = ttk.Treeview(self.window, show='headings',

column=("排名","姓名", "学号", "语文", "数学", "英语","物理","总分")) # 创建表格对象

Button(self.window, text="查看完毕", width=12, font=tkFont.Font(size=12), command=self.click).place(x=210,

y=350)

self.xsb = Scrollbar(self.window, orient="horizontal", command=tree.xview()) # x滚动条

self.ysb = Scrollbar(self.window, orient="vertical", command=tree.yview()) # y滚动条

tree.configure(yscroll=self.ysb.set, xscroll=self.xsb.set) # y滚动条关联

self.ysb.pack(side="right", fill="y")

tree.column("排名", width=40)

tree.column("姓名", width=60) # 设置列

tree.column("学号", width=40)

tree.column("语文", width=40)

tree.column("数学", width=40)

tree.column("英语", width=40)

tree.column("物理", width=40)

tree.column("总分", width=40)

tree.heading("排名", text="排名")

tree.heading("姓名", text="姓名") # 设置显示的表头名

tree.heading("学号", text="学号")

tree.heading("语文", text="语文")

tree.heading("数学", text="数学")

tree.heading("英语", text="英语")

tree.heading("物理", text="物理")

tree.heading("总分", text="总分")

fo = open("成绩.csv", "r",encoding='utf-8')

lls = []

i = 0

for line in fo:

line = line.replace("\n", "")

lls.append(line.split(","))

fo.close()

ls = lls[:0]

ls.extend(lls[1:])

ls=self.paixu(ls,6)

for i in range(len(ls)):

tree.insert("", i, values=(

i+1,ls[i][0], ls[i][1], ls[i][2], ls[i][3], ls[i][4], ls[i][5],ls[i][6],self.score\_rank(ls,6, ls[i][6])))

tree.pack(pady=30)

class CountFrame(Frame): # 继承Frame类 查找

def \_\_init\_\_(self, master=None):

Frame.\_\_init\_\_(self, master)

self.root = master # 定义内部变量root

self.createPage()

def createPage(self):

Label(self, text="查看表格", font=("宋体", 20)).grid(row=0, pady=40, stick=W)

Button(self, text='学生总成绩',font=tkFont.Font(size=12),command=self.zongfen).grid(row=2, stick=W, pady=10)

Button(self, text='各科平均分',font=tkFont.Font(size=12),command=self.pingjun).grid(row=3, stick=W, pady=10)

Button(self, text='各科最高分',font=tkFont.Font(size=12),command=self.zuigao).grid(row=4, stick=W, pady=10)

Button(self, text='各科最低分',font=tkFont.Font(size=12),command=self.zuidi).grid(row=5, stick=W, pady=10)

def zongfen(self):

plt.rcParams['font.sans-serif'] = ['SimHei']

plt.rcParams['axes.unicode\_minus'] = False

df = pd.read\_csv('成绩.csv', encoding='utf-8') # 读取student\_score.csv文件为DataFrame字符流

name = df.姓名

students\_scores =df.总分

plt.title('学生总成绩分布图')

plt.xlabel('学号')

plt.ylabel('总分')

plt.bar(name, students\_scores)

plt.show()

def pingjun(self):

plt.rcParams['font.sans-serif'] = ['SimHei']

plt.rcParams['axes.unicode\_minus'] = False

df = pd.read\_csv('成绩.csv', encoding='utf-8')

chinese\_avg = df.语文.mean()

math\_avg = df.数学.mean() # 高数平均分

english\_avg = df.英语.mean() # 英语平均分

physics\_avg = df.物理.mean()

plt.title('每门课程平均分展示图')

plt.xlabel('课程名')

plt.ylabel('平均分')

plt.bar('语文', chinese\_avg)

plt.bar('数学', math\_avg)

plt.bar('英语', english\_avg)

plt.bar('物理', physics\_avg)

plt.show()

def zuigao(self):

plt.rcParams['font.sans-serif'] = ['SimHei']

plt.rcParams['axes.unicode\_minus'] = False

df = pd.read\_csv('成绩.csv', encoding='utf-8')

chinese\_max = df.语文.max() # 语文最大值

math\_max = df.数学.max() # 高数最大值

english\_max = df.英语.max() # 英语最大值

physics\_max = df.物理.max() # 物理最大值

plt.title('每门课程最高分展示图')

plt.xlabel('课程名')

plt.ylabel('最高分')

plt.bar('语文', chinese\_max)

plt.bar('数学', math\_max)

plt.bar('英语', english\_max)

plt.bar('物理', physics\_max)

plt.show()

def zuidi(self):

plt.rcParams['font.sans-serif'] = ['SimHei']

plt.rcParams['axes.unicode\_minus'] = False

df = pd.read\_csv('成绩.csv', encoding='utf-8')

chinese\_min = df.语文.min() # 语文最小值

math\_min = df.数学.min() # 高数最小值

english\_min = df.英语.min() # 英语最小值

physics\_min = df.物理.min() # 物理最小值

plt.title('每门课程最低分展示图')

plt.xlabel('课程名')

plt.ylabel('最低分')

plt.bar('语文', chinese\_min)

plt.bar('数学', math\_min)

plt.bar('英语', english\_min)

plt.bar('物理', physics\_min)

plt.show()

————————————————

版权声明：本文为CSDN博主「长星照耀十三州府ˇKris」的原创文章，遵循CC 4.0 BY-SA版权协议，转载请附上原文出处链接及本声明。

原文链接：https://blog.csdn.net/weixin\_47336144/article/details/122538235