随机点名小程序

问题描述

随机点名程序（越不来上课的人，被点中的概率越高，实现抽查问题、预警等功能）

问题分析

采用python 的 tkinter 库实现用户图形界面，采用 pickle 存储数据，可以对点名对象进行插入或删除。user_info.pickle文件中存储着一个字典，包括名字和缺勤次数，根据迟到的次数往列表中增加名字的个数，越不来上课的人被点中的概率越高。

代码分析

导入相应的包：

```
import tkinter as tk
import tkinter.messagebox
import pickle
import random
```

定义随机点名对象，方法有

1. def init_data(self): 初始化数据
2. def init_lable(self): 初始化标签
3. def run(self): 运行

```
class RandomNameGame(object):
    def __init__(self):
        self.window = tk.Tk()  #建立底层窗口
        self.window.title('随机点名程序')   #窗口名称
        self.window.geometry('500x500')   #窗口大小
        self.var = tk.StringVar()   #被点到成员的名字
        self.status = True  #随机状态控制
```

初始化数据：从pickle上下载数据

```
    def init_data(self):
        try:
            with open('users_info.pickle','rb') as user_file:
                self.user_info = pickle.load(user_file)
                print(self.user_info)
                self.user_name = [i for i in self.user_info.keys()]
                self.length = len(self.user_info)   #成员数量
                print(self.length)
        except FileNotFoundError:  #初始化成员
            with open('users_info.pickle','wb') as user_file:
                self.user_info = {'teacher':'0'}    #成员名字，旷课次数
                pickle.dump(self.user_info,user_file)
```

初始化标签：

```
    def init_lable(self):
        #刷新
        def Refresh():
            self.init_data()
            init_student_label()
            print(self.user_info)

        def init_student_label():
            #初始化标签
            self.var_x = 40
            self.var_y = 150
            for item in self.user_name:
                if(self.var_x>=450):
                    tk.messagebox('Error','The student is too much')
                if(self.var_y>=350):
                    self.var_x += 100
                    self.var_y = 150
                    self.generate_label(item,self.var_x,self.var_y)
                    self.var_y +=60
                else:
                    self.generate_label(item,self.var_x,self.var_y)
                    self.var_y += 60
        init_student_label()

        #添加新成员
        def Add():
            #判断是否注册过，如果没注册则添加
            def sign_to():
                np = new_user_name.get()
                with open('users_info.pickle','rb') as user_file:  #对比有没重复
                    exist_user_info = pickle.load(user_file)
                if np in exist_user_info:
                    tk.messagebox('Error','The student has already in !')
                else:
                    exist_user_info[np] = 0
                    with open('users_info.pickle','wb') as user_file:
                        pickle.dump(exist_user_info,user_file)
                    tk.messagebox.showinfo('successful','You have successfully
add !')
                    window_sign_up.destroy()

            window_sign_up = tk.Toplevel(self.window)
            window_sign_up.geometry('300x100')
            window_sign_up.title('Add new student')
            tk.Label(window_sign_up,text='New User name :').place(x=20,y=30)
            new_user_name = tk.StringVar()  #定义新成员变量
            new_user_name.set('请输入新成员名字')
            #add entry
            entry_new_user_name =
tk.Entry(window_sign_up,textvariable=new_user_name)
            entry_new_user_name.place(x=140,y=30)
            #Add Button
            btn_sign_up = tk.Button(window_sign_up,text='Add',command=sign_to)
            btn_sign_up.place(x=130,y=65)

        #删除成员
        def Drop():
```

```
            #self.init_data()
            window_drop_out = tk.Toplevel(self.window)
            window_drop_out.title('drop out')
            window_drop_out.geometry('400x200')

            list1 = self.user_name
            lb = tk.Listbox(window_drop_out,listvar = list1)
            for item in list1:
                lb.insert('end',item)
            lb.place(x=80,y=0)
             #撤销
            def cancel():
                lb.delete(0, "end")  #删除所有元素，用于更新列表
                window_drop_out.destroy()
            def dropout():
                value = lb.get(lb.curselection())
                print(value)
                #lb.delete(value)
                with open('users_info.pickle','rb') as user_file:  #对比有没重复
                    exist_user_info = pickle.load(user_file)
                    del exist_user_info[value]
                with open('users_info.pickle','wb') as user_file:
                    pickle.dump(exist_user_info,user_file)
                tk.messagebox.showinfo('successful','You have drop out it!')
                window_drop_out.destroy()
            #定义删除按钮
            b1 = tk.Button(window_drop_out,text = 'drop',width = 15,height =
2,command=dropout,fg='white',bg='green')
            b1.place(x=250,y=40)
            b2 = tk.Button(window_drop_out,text = 'cancel',width = 15,height =
2,command=cancel,fg='white',bg='green')
            b2.place(x=250,y=100)
        def startup():
            list1 = []
            #根据迟到的次数往列表中增加名字的个数
            for key,value in self.user_info.items():
                for i in range(value+1):
                    list1.append(key)  #往名单中增加名字个数
            on_label = random.randint(0,len(list1)-1)  #随机抽取名单中的序列
            self.name = list1[on_label]  #得到点到人的名字
            self.var.set(self.name)
            print(self.name)


        def Attend():  #出勤
            tk.messagebox.showinfo('提示','perfect')
        def Absent():  #缺勤，记录次数
            self.user_info[self.name] +=1
            print(self.user_info)
            with open('users_info.pickle','wb') as user_file:
                pickle.dump(self.user_info,user_file)
            if(self.user_info[self.name]<5):
                tk.messagebox.showinfo('提示','你已经缺
勤'+str(self.user_info[self.name])+'次')
            elif(self.user_info[self.name]>=5):
                tk.messagebox.showinfo('提示','你已经缺
勤'+str(self.user_info[self.name])+'次'+'。您将被通知家长')
```

```
        #标题
        Label_title = tk.Label(self.window,text = '点名啦',font=
('Arial','24'),fg='blue',height=2).pack()
        #点名名字的标题栏
        l = tk.Label(self.window,textvariable = self.var ,bg='yellow',font=
('Arial',12),width = 15,height = 2)
        l.pack()
        #刷新，添加，删除
        Button_Refresh = tk.Button(self.window,text='刷新',bg='green',font=
('Arial',12),fg='white',command=Refresh).place(x=60,y=420)
        Button_Add = tk.Button(self.window,text='添加',bg='green',font=
('Arial',12),fg='white',command=Add).place(x=220,y=420)
        Button_Drop = tk.Button(self.window,text='删除',bg='green',font=
('Arial',12),fg='white',command=Drop).place(x=380,y=420)
        #Label = tk.Label(self.window,text = self.user_info.values()).pack()
        #开始点名按钮
        Button_Stop = tk.Button(self.window,text='Start',bg='green',font=
('Arial',20),fg='white',command=Start).place(x=55,y=30)
        #到
        Button_Attend = tk.Button(self.window,text='出勤',bg='green',font=
('Arial',13),fg='white',command=Attend).place(x=420,y=15)
        #没到
        Button_Absent = tk.Button(self.window,text='缺勤',bg='green',font=
('Arial',13),fg='white',command=Absent).place(x=420,y=70)
```

运行：

```
    def run(self):
        self.init_data()
        self.init_lable()
        self.window.mainloop()
```

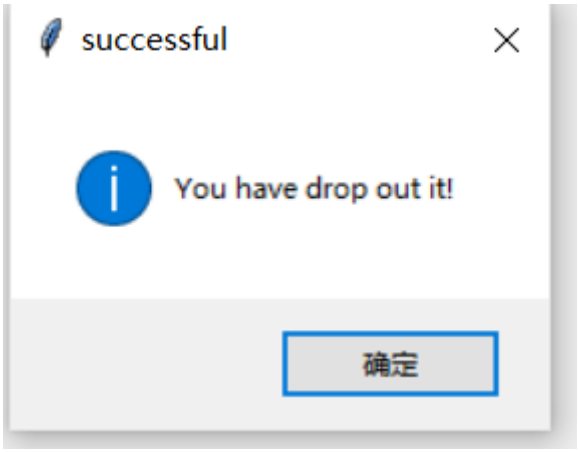
实现结果

添加名字：


随机点名程序
Start
点名啦
出勤
缺勤
Add new student
New User name :
小红
Add
刷新
添加
删除

删除名字:


随机点名程序
Start
点名啦
出勤
缺勤
drop out
小红
小王
小明
张三
李四
王五
drop
cancel
刷新
添加
删除


successful
You have drop out it!
确定

缺勤:

改进

迟到五次以上实现发短信提醒的功能

```
# 创建发送短信窗口
def send_message():
    email = entry_email.get()
    content = "你已经迟到"+str(self.user_info[self.name])+'次，请通知家长'
    flag = send_mail(content, email)
    if(flag == 1):
        tk.messagebox.showinfo('提示', '信息发送成功')
    else:
        tk.messagebox.showerror('失败', '请重新发送')
        window_send_message = tk.Toplevel(self.window)
        window_send_message.geometry('400x100')
        window_send_message.title('Add new student')
        tk.Label(window_send_message, text='Email :').place(x=20, y=30)
        new_user_name = tk.StringVar()  # 定义新成员变量
        new_user_name.set('输入邮箱')
```

```
        # add entry
        entry_email = tk.Entry(window_send_message, textvariable=new_user_name)
        entry_email.place(x=140, y=30)
        # Add Button
        btn_send_message = tk.Button(window_send_message, text='发送',
command=send_message)
        btn_send_message.place(x=330, y=30)
       #tk.messagebox.showinfo('提示', '你已经缺勤' +
str(self.user_info[self.name]) + '次' + '。您将被通知家长')
```

定义发短信的模块,可以设定发送方邮箱，需在QQ邮箱中开通POP3/SMTP服务。send_message.py模块如下：

```
import smtplib
from email.mime.text import MIMEText

def send_mail(content, email):
    msg_from = '1724647576@qq.com'  # 发送方邮箱
    passwd = 'ezdgvskeyvpvdbdh'  # 填入发送方邮箱的授权码
    msg_to = email
    subject = "点名"
    msg = MIMEText(content)
    msg['Subject'] = subject
    msg['From'] = msg_from
    msg['To'] = msg_to
    try:
        s = smtplib.SMTP_SSL("smtp.qq.com", 465)  # 发送一般使用465端口，使用163邮箱
的话，需要更换成smtp.163.com
        s.login(msg_from, passwd)
        s.sendmail(msg_from, msg_to, msg.as_string())
    except Exception as e:
        return 0
        print('error')
        return False
    else:
        return 1
        print('邮件发送成功')

    finally:
        s.quit()
    return True

if __name__ == '__main__':
    message = '测试'
    email = '1724647576@qq.com'
    send_mail(message, email)
```


随机点名程序
Start
点名啦
出勤
缺勤
张三
Add new student
Email :
输入邮箱
发送
李四
刷新
添加
删除

显示：发送成功！