《Python与算法设计实验》作业

2023.11

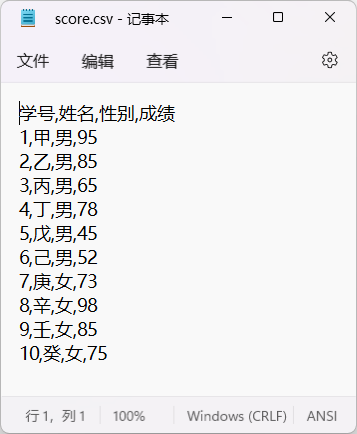
**【实验一：数据分析及可视化】**

* 设计思路

首先利用csv模块读取"score.csv"文件中的数据，将每一行数据转换成字典形式存储在列表stu中。然后统计各类别人数，并计算男女生比例。接着使用matplotlib库绘制基于学生成绩的饼状图，展示了各等级学生比例。接下来对学生的成绩降序排序，并将排序结果写入新的CSV文件"sorted.csv"中。

* 程序实现

创建一个CSV格式文件score.csv，存储十位学生的学号、姓名、性别、成绩信息（如下图所示）：



程序代码如下：

import csv

import matplotlib.pyplot as plt

with open("score.csv",'r') as f: # with open语句确保在处理完文件后，文件被正确关闭

dic=csv.DictReader(f) # 将文件的内容转换成字典格式

stu=[line for line in dic] # 列表推导式，将dic每一行数据以字典形式依次添加到列表stu中

m,fm,a,b,c,d,e=0,0,0,0,0,0,0 # 分别表示男生、女生、五个等级的人数

for s in stu:

if s["性别"]=="男":

m+=1

elif s["性别"]=="女":

fm+=1

s=int(s["成绩"])

if s>=90:

a+=1

elif s>=80:

b+=1

elif s>=70:

c+=1

elif s>=60:

d+=1

else:

e+=1

mrate=m/len(stu)

fmrate=fm/len(stu)

print('男生比例：{:.2%}'.format(mrate))

print('女生比例：{:.2%}'.format(fmrate))

print('优秀：{}人'.format(a))

print('良好：{}人'.format(b))

print('中等：{}人'.format(c))

print('及格：{}人'.format(d))

print('不及格：{}人'.format(e))

plt.rcParams['font.family']='Microsoft YaHei'

# 字体设置为“微软雅黑”，确保在生成的图表中，中文文本能够正常显示

labels=['优秀','良好','中等','及格','不及格']

sizes=[a,b,c,d,e]

plt.pie(sizes,labels=labels)

plt.title('学生成绩统计饼状图')

plt.legend(loc='upper right') # 设置标签位置为窗口右上角

plt.show()

stu.sort(key=lambda x:int(x['成绩']),reverse=True)

# sort为升序排序，使用reverse=True方法反转为降序排序

with open('sorted.csv','w',newline='') as file:

writer=csv.writer(file) # 创建一个CSV文件写入器，用来将数据写入到指定文件中

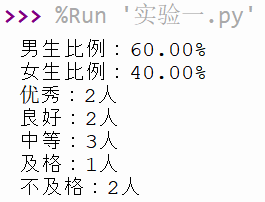
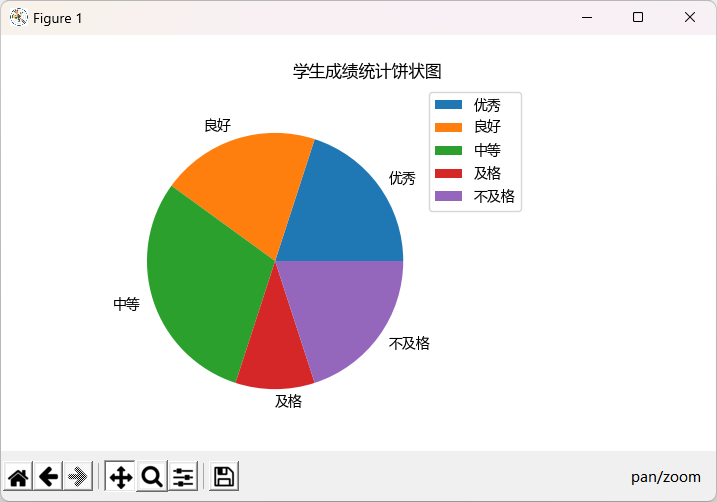
writer.writerow(['学号','姓名','性别','成绩'])

for s in stu:

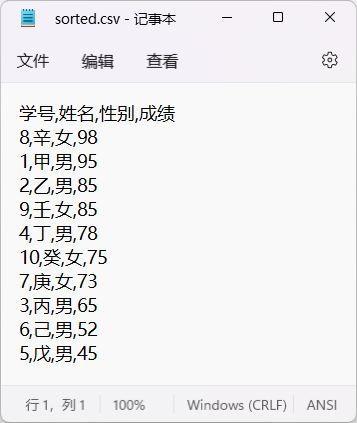
writer.writerow([s['学号'],s['姓名'],s['性别'],s['成绩']]) # 将一行数据写入CSV文件

* 运行结果

输出结果如下图所示：

新生成的sorted.csv文件如图所示：



**【实验二：GUI设计】**

* 设计思路

导入Tkinter库，创建一个Tkinter应用程序。使用entry小部件来显示输入的内容和输出结果，使用Button小部件来实现计算器的按钮。并巧妙利用try和except语句来处理异常情况。

* 程序实现

import tkinter as tk

app=tk.Tk()

app.title("简易计算器 by LJX from HDU")

input\_text=tk.StringVar() # 创建字符串变量（输入框）

output\_text=tk.StringVar() # 创建字符串变量（输出框）

input\_entry=tk.Entry(app, textvariable=input\_text,width=20)

input\_entry.grid(row=0,column=0,columnspan=4,padx=10,pady=10)

# 创建一个文本输入框，并指定了输入框的一些属性

# 使用textvariable=input\_text将之前创建的input\_text字符串变量与这个文本输入框关联起来

# 将文本输入框input\_entry放置在网格布局中的指定位置，并设置其跨越多列以及周围的间距。

output\_label=tk.Label(app,textvariable=output\_text,width=20,relief="solid")

output\_label.grid(row=1,column=0,columnspan=4,padx=10,pady=10)

# 创建一个文本输出框，并指定了输入框的一些属性

# 语句relief="solid"指定了标签的外边框样式为实线边框（不必深究，是为了和输入框显得不一样）

# 将文本输出框output\_entry放置在网格布局中的指定位置，并设置其跨越多列以及周围的间距。

def button\_click(value):

current=input\_text.get() # 从文本变量input\_text中获取当前文本内容，赋值给current

if value=="=":

try:

result=eval(current)

# 使用内置eval函数执行字符串表达式

output\_text.set(result)

except:

output\_text.set("错误")

"""

try和except是Python中的异常处理语句。try语句包含一段可能会引发异常的代码，而except语句则定义了当出现异常时应该执行的代码。在本例中，如果表达式无法计算或者存在语法错误，eval函数会引发一个异常。为了避免这种情况下程序崩溃，我们使用try语句来捕获异常，使用except语句当遇到异常后在输出框中显示"错误"。

"""

elif value=="清除":

# 使用set函数将两个框内容改为空字符串

input\_text.set("")

output\_text.set("")

else:

input\_text.set(current+value) # 新输入的值跟在当前已输入值之后

buttons=['7','8','9','/','4','5','6','\*','1', '2', '3', '-','0','.','=','+','清除']

row,col=2,0

for button in buttons: # 循环语句，整齐放置button

tk.Button(app,text=button,width=10,bg='#ADD8E6',command=lambda value=button:

button\_click(value)).grid(row=row,column=col,padx=5,pady=5)

# 使用lambda表达式创建一个匿名函数，点击按钮会调用button\_click函数，传入value作为参数。

# 语句row=row, column=col指定了按钮在应用程序的网格布局中的行和列的位置。

# 语句padx=5, pady=5设置了按钮周围的水平和垂直间距为5个像素，以便在布局中留出一些空白。

col+=1

if col>3:

col=0

row+=1

app.mainloop()

# 启动GUI应用程序的主事件循环，使得应用程序能够响应用户的操作并持续运行

* 运行界面（图3考虑了表达式无法计算或者存在语法错误的情况）

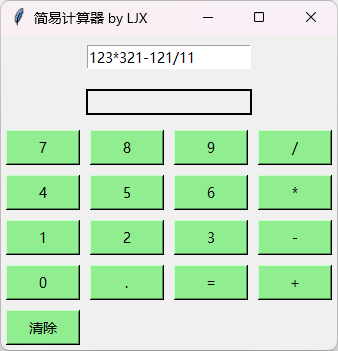
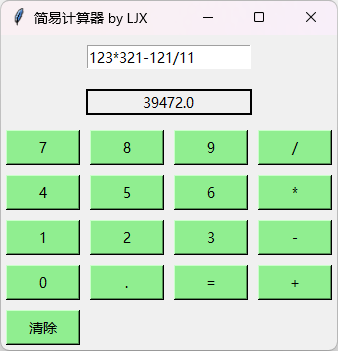
  

图1 图2 图3

**【实验三：自由实验】**

* 设计灵感

项目灵感来自于米哈游《原神》祈愿系统，笔者也考虑到杭电学子们心向远方，憧憬更好的未来，特设计一个简洁的祈愿程序鼓励学子们勇敢追求。

结合课程所学习的tkinter模块，PIL库中的Image模块、random模块、time时间模块，以及笔者有所浅学的class类的相关知识，设计了以下程序。

* 设计思路

这个程序是一个简单的抽奖应用，使用了tkinter进行界面设计，并利用PIL库来处理图像。

1. 创建LotteryApp类，该类包含了抽奖应用的主要逻辑和界面设计。
2. 在\_\_init\_\_方法中，首先初始化主窗口，设置窗口标题，读入一张图片并在窗口中显示。
3. 通过after方法设置延时两秒后调用show\_lottery\_page方法，以展示抽奖页面。
4. 在show\_lottery\_page方法中，销毁启动图片，创建用于显示祝贺信息和抽奖结果的标签，以及执行抽奖操作的按钮。
5. draw\_lottery方法用于执行抽奖操作，每次祈愿消耗160原石。如果原石充足，随机抽取并更新抽奖结果和原石数量；如果原石不足，提示充值。
6. 在\_\_main\_\_部分，创建主窗口并运行程序。

* 程序实现

import tkinter as tk

from PIL import Image, ImageTk

import random

import time

class LotteryApp: # 将抽奖逻辑封装在LotteryApp类中，使得程序结构清晰，易于理解和维护

def \_\_init\_\_(self, master): # \_\_init\_\_方法用于在对象实例化时初始化对象的属性或执行其他必要的设置操作

self.master = master # 初始化主窗口

self.master.title("Genshin Impact Wish System —— HDU Ver.")

image = Image.open("hduqd.png")

photo = ImageTk.PhotoImage(image) # 使用ImageTk模块创建图片对象

self.startup\_label = tk.Label(self.master, image=photo) # 创建标签并显示图片

self.startup\_label.image = photo # 保持引用，防止被垃圾回收

self.startup\_label.pack() # 将标签放置到主窗口上

self.master.after(2000, self.show\_lottery\_page) # 在2秒后调用show\_lottery\_page方法

self.stone\_count = 1600 # 初始化原石数量

self.stone\_label = tk.Label(self.master, text=f"原石：{self.stone\_count}", font=("宋体", 20)) # 创建标签显示原石数量

self.stone\_label.place(x=1300, y=50)

def show\_lottery\_page(self):

self.startup\_label.destroy() # 销毁启动图片

self.label = tk.Label(self.master, fg='red', text="恭喜您获得：", font=("宋体", 40)) # 创建标签显示祝贺信息

self.label.pack(pady=100)

self.result\_var = tk.StringVar() # 创建变量对象用于显示抽奖结果

self.result\_label = tk.Label(self.master, textvariable=self.result\_var, font=("等线", 90)) # 创建标签显示抽奖结果

self.result\_label.pack()

self.lottery\_button = tk.Button(self.master, bg='lightgreen', text="祈愿1次（原石\*160）", command=self.draw\_lottery, font=("宋体", 14))

# 创建按钮用于执行抽奖操作

self.lottery\_button.pack(pady=100)

def draw\_lottery(self):

cost = 160

if self.stone\_count >= cost:

candidates = ["PKU录取通知书", "THU录取通知书", "ZJU录取通知书", "SJTU录取通知书","HDU再读十年书"] # 卡池具体内容

winner = random.choice(candidates) # 使用choice函数随机抽取序列中的一个元素

self.result\_var.set(winner) # 设置抽奖结果

self.stone\_count -= cost # 扣除抽奖消耗的原石

self.stone\_label.config(text=f"原石：{self.stone\_count}") # 更新原石数量显示

else:

tk.messagebox.showinfo("提示", "原石不足，建议充值¥648！") # 如果原石不足，弹出提示框

if \_\_name\_\_ == "\_\_main\_\_":

root = tk.Tk() # 创建主窗口

root.geometry('1600x900-150-100') # 设置主窗口大小和位置

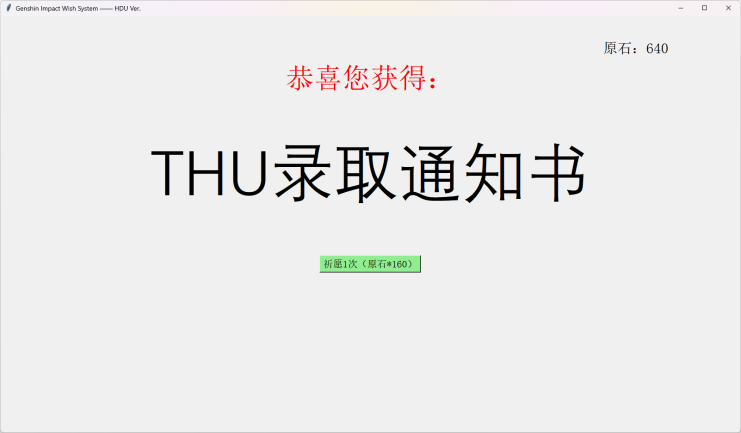
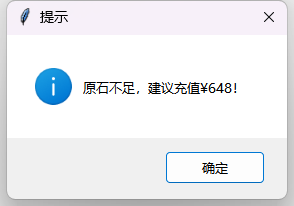
app = LotteryApp(root) # 创建抽奖应用实例

root.mainloop() # 运行主程序

* 运行界面

启动界面（显示2秒） 程序主界面

程序执行中界面 提示框界面

**课程总结**

《Python与算法设计实验》是一门非常实用和有趣的课程，它主要介绍了Python编程语言和算法设计的基本概念和实践方法。这门课程涵盖了Python的基础语法、函数式编程、面向对象编程、数据结构和算法等方面的内容，通过大量的编程实验和项目实践，让我们深入理解并掌握了Python编程和算法设计的核心技术。

通过这门课程的学习和实践，我深刻认识到Python编程语言的强大和灵活性，以及算法设计的重要性和挑战性。我学会了如何使用Python来解决各种实际问题，如文本处理、数据分析、图像处理等，也学会了如何设计和实现高效的算法来解决各种计算问题，如排序、查找等。这些知识和技能对我将来的学习和工作都非常有帮助。

总之，我认为《Python与算法设计实验》是一门非常优秀的课程，它不仅涵盖了Python编程和算法设计的核心内容，而且注重实践和创新，培养了我们的编程思维和创造力。在这门课程中，索老师是一位非常出色的指导者。他非常注重培养我们的编程思维和创新能力，鼓励我们在实验和项目中自由发挥，尝试不同的算法和实现方式，从而提高我们的编程水平和创造力。我非常感谢索老师，他的指导和帮助让我们受益匪浅。