山东大学 软件工程 学院

2019-2020学年第1学期

Python程序设计与大数据分析 课程实验报告<模板>

**注：实验报告电子版文件名命名方式：学号+班级+姓名+实验\*。**

|  |  |  |  |
| --- | --- | --- | --- |
| 学号：201800301249 | 姓名：王帅 | | 班级： 软件2018.4班 |
| 实验题目：程序设计方法论 | | | |
| 实验学时：4 | | 实验日期： 2019年9月10日 | |
| 一、实验类型：验证型、设计型 二、建议学时：2 三、实验目的：  1.了解程序设计方法论  四、实验内容： 1、阅读教材《Python 语言程序设计基础（第2版）》第8章：程序设计方法 论  2、验证型练习：  2.1、验证8.2 实例15：体育竞技分析的整个流程和实例代码  2.2、验证8.7 实例16：pip 安装脚本 （不用提交，直接在自己机器上安装后进行测试）  3、设计型练习：教材P36 程序练习题8.1,8.4 | | | |
| 硬件环境：  个人电脑 | | | |
| 软件环境：  IDLE | | | |
| 实验步骤与内容：  实验步骤：   1. 打开IDLE，根据实验要求，熟悉其使用 2. 根据书上的代码，深入自己对python的理解。   实验内容：教材PPT第8章种的范例程序代码 | | | |
| 结论分析与体会：  Python确实是一个比大多数语言都要方便简单的语言，简洁明了，大大减少了编写代码的时间。  虽然语法较为简单，但是自己的特有方法并不少，都是使用起来可以使我们的编程更加简便的方法，应当合理使用。  注意在使用IDLE时控制缩进，否则可能会发生较大的错误。 | | | |

以下分别粘贴实验中验证型题目中通过的程序代码（详细编号）与运行结果；

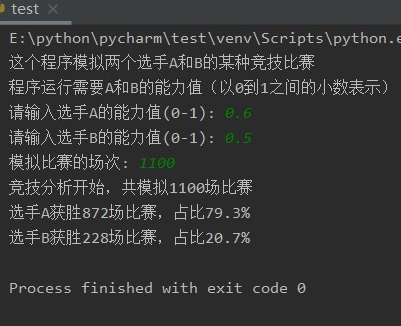
粘贴实验中设计型题目中调试通过的程序代码（详细编号）与运行结果

验证型题目：

8.2、

from random import random  
  
  
def printIntro():  
 print("这个程序模拟两个选手A和B的某种竞技比赛")  
 print("程序运行需要A和B的能力值（以0到1之间的小数表示）")  
  
  
def getInputs():  
 a = eval(input("请输入选手A的能力值(0-1): "))  
 b = eval(input("请输入选手B的能力值(0-1): "))  
 n = eval(input("模拟比赛的场次: "))  
 return a, b, n  
  
  
def simNGames(n, probA, probB):  
 winsA, winsB = 0, 0  
 for i in range(n):  
 scoreA, scoreB = simOneGame(probA, probB)  
 if scoreA > scoreB:  
 winsA += 1  
 else:  
 winsB += 1  
 return winsA, winsB  
  
  
def gameOver(a, b):  
 return a == 15 or b == 15  
  
  
def simOneGame(probA, probB):  
 scoreA, scoreB = 0, 0  
 serving = "A"  
 while not gameOver(scoreA, scoreB):  
 if serving == "A":  
 if random() < probA:  
 scoreA += 1  
 else:  
 serving = "B"  
 else:  
 if random() < probB:  
 scoreB += 1  
 else:  
 serving = "A"  
 return scoreA, scoreB  
  
  
def printSummary(winsA, winsB):  
 n = winsA + winsB  
 print("竞技分析开始，共模拟{}场比赛".format(n))  
 print("选手A获胜{}场比赛，占比{:0.1%}".format(winsA, winsA / n))  
 print("选手B获胜{}场比赛，占比{:0.1%}".format(winsB, winsB / n))  
  
  
def main():  
 printIntro()  
 probA, probB, n = getInputs()  
 winsA, winsB = simNGames(n, probA, probB)  
 printSummary(winsA, winsB)  
  
  
main()

结果：

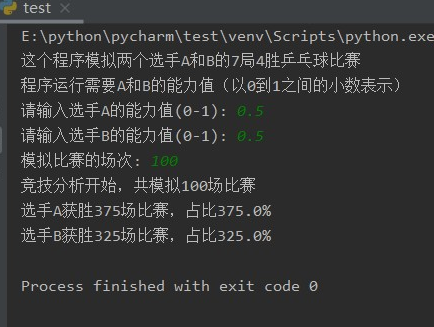


设计型题目：

8.1、

from random import random  
  
  
def printIntro(): # 打印程序介绍信息  
 print("这个程序模拟两个选手A和B的7局4胜乒乓球比赛")  
 print("程序运行需要A和B的能力值（以0到1之间的小数表示）")  
  
  
def getInputs(): # 获得程序运行参数  
 a = eval(input("请输入选手A的能力值(0-1): "))  
 b = eval(input("请输入选手B的能力值(0-1): "))  
 n = eval(input("模拟比赛的场次: "))  
 return a, b, n  
  
  
def simNGames(n, probA, probB): # 进行N场比赛  
 winsA, winsB = 0, 0  
 for i in range(n):  
 for j in range(7): # 进行7局4胜的比赛  
 scoreA, scoreB = simOneGame(probA, probB)  
 if scoreA > scoreB:  
 winsA += 1  
 else:  
 winsB += 1  
 return winsA, winsB  
  
  
def gameOver(a, b): # 正常比赛结束  
 return a == 11 or b == 11  
  
  
def gameOver2(a, b): # 进行抢12比赛结束  
 if abs((a - b)) >= 2:  
 return a, b  
  
  
def simOneGame(probA, probB): # 进行一场比赛  
 scoreA, scoreB = 0, 0 # 初始化AB的得分  
 serving = "A"  
 while not gameOver(scoreA, scoreB): # 用while循环来执行比赛  
 if scoreA == 10 and scoreB == 10:  
 return simtwoGame2(probA, probB)  
 if serving == "A":  
 if random() < probA: ##用随机数生成胜负  
 scoreA += 1  
 else:  
 serving = "B"  
 else:  
 if random() < probB:  
 scoreB += 1  
 else:  
 serving = "A"  
 return scoreA, scoreB  
  
  
def simtwoGame2(probA, probB):  
 scoreA, scoreB = 10, 10  
 serving = "A"  
 while not gameOver2(scoreA, scoreB):  
 if serving == "A":  
 if random() < probA:  
 scoreA += 1  
 else:  
 serving = "B"  
 else:  
 if random() < probB:  
 scoreB += 1  
 else:  
 serving = "A"  
 return scoreA, scoreB  
  
  
def printSummary(winsA, winsB):  
 n = int((winsA + winsB)/7)  
 print("竞技分析开始，共模拟{}场比赛".format(n))  
 print("选手A获胜{}场比赛，占比{:0.1%}".format(winsA, winsA / n))  
 print("选手B获胜{}场比赛，占比{:0.1%}".format(winsB, winsB / n))  
  
  
def main():  
 printIntro()  
 probA, probB, n = getInputs()  
 winsA, winsB = simNGames(n, probA, probB)  
 printSummary(winsA, winsB)  
  
  
main()

结果：



8.4、

# 导入扩展库  
import re # 正则表达式库  
import collections # 词频统计库  
import numpy as np # numpy数据处理库  
import jieba # 结巴分词  
import wordcloud # 词云展示库  
from PIL import Image # 图像处理库  
import matplotlib.pyplot as plt # 图像展示库  
  
# 读取文件  
fn = open('三国演义.txt', encoding = 'utf-8') # 打开文件  
string\_data = fn.read() # 读出整个文件  
fn.close() # 关闭文件  
  
# 文本预处理  
pattern = re.compile(u'\t|\n|\.|-|:|;|\)|\(|\?|"') # 定义正则表达式匹配模式  
string\_data = re.sub(pattern, '', string\_data) # 将符合模式的字符去除  
  
# 文本分词  
seg\_list\_exact = jieba.cut(string\_data, cut\_all=False) # 精确模式分词  
object\_list = []  
remove\_words = [u'的', u'，', u'和', u'是', u'随着', u'对于', u'对', u'等', u'能', u'都', u'。', u' ', u'、', u'中', u'在', u'了',  
 u'通常', u'如果', u'我们', u'需要'] # 自定义去除词库  
  
for word in seg\_list\_exact: # 循环读出每个分词  
 if word not in remove\_words: # 如果不在去除词库中  
 object\_list.append(word) # 分词追加到列表  
  
# 词频统计  
word\_counts = collections.Counter(object\_list) # 对分词做词频统计  
word\_counts\_top10 = word\_counts.most\_common(10) # 获取前10最高频的词  
print(word\_counts\_top10) # 输出检查  
  
# 词频展示  
mask = np.array(Image.open('三国演义.jpg')) # 定义词频背景  
wc = wordcloud.WordCloud(  
 font\_path='C:/Windows/Fonts/simhei.ttf', # 设置字体格式  
 mask=mask, # 设置背景图  
 max\_words=200, # 最多显示词数  
 max\_font\_size=100 # 字体最大值  
)  
  
wc.generate\_from\_frequencies(word\_counts) # 从字典生成词云  
image\_colors = wordcloud.ImageColorGenerator(mask) # 从背景图建立颜色方案  
wc.recolor(color\_func=image\_colors) # 将词云颜色设置为背景图方案  
plt.imshow(wc) # 显示词云  
plt.axis('off') # 关闭坐标轴  
plt.show() # 显示图像

结果：

