山东大学网络空间安全学院

Python高级程序设计课程实验报告

|  |  |  |  |
| --- | --- | --- | --- |
| 学号：202100460065 | 姓名：李昕 | | 班级：21级密码2班 |
| 实验题目：异常处理+模块 | | | |
| 实验学时：2 | | 实验日期：2022/10/24 | |
| 实验目的：熟悉Python异常处理和模块。 | | | |
| 硬件环境：  硬件环境：  AMD Ryzen 7 5800H with Radeon Graphics     3.20 GHz  机带 RAM 16.0 GB (13.9 GB 可用) | | | |
| 软件环境：IDLE | | | |
| 实验步骤与内容：  本次实验包括三个内容：   1. 猜数游戏，编写程序模拟猜数游戏。程序运行时，系统生成一个100以内的随机数，然后提示用户进行猜测，并根据用户输入进行必要的提示（猜对了、太大了、太小了），如果猜对则提前结束程序，如果次数用完仍没有猜对，提示游戏结束并给出正确答案。要求用到异常处理，处理输入不是数字的情况。 2. 编写代码，读取文本文件内容并统计其中每个英文字母的出现次数，如果文件不存在则给出友好提示，使用异常处理结构实现。 3. 学习fileinput模块，编写代码在一个文件的每一行文本的前面添加 ‘#行号’字样。 编写代码，同时处理多个文件，在每一个文件的每一行文本的前面添加 ‘#行号’字样。   **代码实现：**  【**猜数游戏**】   1. import random 2. print("已经随机生成了一个数字") 3. num=int(input("设置尝试次数：")) 4. flag=0 5. k=random.randint(1,100) *#生成一个随机数* 6. while num>0: 7. a=input("请输入数：") 8. num=num-1 9. try: 10. a=int(a) 11. except: 12. print("输入格式错误，请重新输入") 13. continue 14. if a== k: 15. print("猜对了！") 16. flag=1 17. break 18. elif a>k: 19. print("太大了，请重新输入") 20. continue 21. else : 22. print("太小了，请重新输入") 23. continue 24. if not flag: 25. print('次数用完了，游戏结束')     运行结果：    【读取文件】  1.使用一个长度为52的列表用来分别存储小写和大写的字母出现的次数  2.如果对应路径找不到相应的文件，则捕获错误，并打印“没有这个文件”  3.利用ASCII码值来判定是哪一个大小写字母，将对应列表中的数值加一  **代码实现：**   1. a=[0]\*52 *#分别存储小写和大写的个数* 2. try : 3. k=open(r"C:\Users\lenovo\Documents\1.txt",mode='r',encoding = 'utf-8') 4. except FileNotFoundError: 5. print("没有这个文件") 6. else : 7. m=k.readlines() 8. for i in m: 9. for j in i: 10. if j.isupper(): 11. a[ord(j)-39]=a[ord(j)-39]+1 12. elif j.islower(): 13. a[ord(j)-97]=a[ord(j)-97]+1 14. for i in range(26): 15. print(chr(i+97),'有',a[i],'个',end='\n') 16. for i in range(26,52): 17. print(chr(i+39),'有',a[i],'个',end='\n ') 18. k.close()   执行结果：    【fileinput模块】  通过预习了解到，同过fileinput模块中的 input()  函数，我们能同时打开指定的多个文件，还可以逐个读取这些文件中的内容  本次实验要求编写代码在一个文件的每一行文本的前面添加 ‘#行号’字样。 编写代码，同时处理多个文件，在每一个文件的每一行文本的前面添加 ‘#行号’字样。  同时使用glob模块一次同时处理多个类似变量名文件。  代码实现：   1. import fileinput 2. import glob 3. for line in fileinput.input(glob.glob(r"C:\Users\waldeinsamkeit\Desktop\test\*.txt")): 4. if fileinput.isfirstline(): 5. print( '-'\*20, 'Reading %s...' % fileinput.filename(), '-'\*20) 6. print( str(fileinput.filelineno()) + ': ' + line.upper());   实验截图： | | | |
| 结论分析与体会：  通过本次实验，我掌握并练习了fileinput模块，熟悉了异常处理和模块用法，并成功运用于程序，加深了对python语言的理解。 | | | |