《Python程序设计》课程实验报告

信息 学院 智能科学与技术 专业 2020 级

实验时间 2021 年 12 月 10 日

姓名 ***Steven*** 学号

实验名称 作业十

实验成绩

一、实验目的

练习使用Python操作文件的方法。

练习使用Python解决某些算法问题的思路

二、实验仪器设备及软件

电脑，Pycharm ，Anaconda Python 3.8

三、实验方案

1. 根据文件名递归地查找文件是否存在。

既然要递归，那么就得自己调用自己。要使用本函数必须要有一个查找的路径，那么思路就出来了：当遍历到文件夹时，用递归函数进入这个子目录中，并遍历。返回值就是是否存在。

2. 根据创建时间查找文件是否存在，并返回其路径。

相较于前题略麻烦一点，不过大体思路还是一样的，就是递归遍历整个目录。当遍历到当前目录存在某个时间创建的文件时，将其绝对路径存在一个pathList列表中。而遍历到当前目录的子目录时，如果其子目录下也存在符合条件的文件，那么就把其返回值合并。最后返回pathList。

3. 打酱油问题。

由于是求最多能买几瓶，那么一定要从优惠最多的开始算起。首先判断是否够买5瓶，如果够买5瓶，那么就买，然后更新瓶数和剩下的钱；随后依次判断是否够买3瓶、1瓶……重复以上步骤，最后返回一直更新后的瓶数。

4. 报数问题

其实本来在别的语言中不太好实现的，但是Python列表支持pop函数，那么就很好办了。就是让index循环遍历整个列表，附带一个计数器同时作为小朋友喊的号码。当喊的号码符合规则的话，就pop掉当前index对应的小朋友。随后更新index、num。直到整个列表中只剩下一个小朋友。

需要注意的是pop了index之后，其之后的小朋友会补位到index处，所以如果进行了pop操作，那么index就不必更新了。

四、实验步骤

1.

|  |
| --- |
| **def findTxt:** |
| **def** findTxt(currPath, findName): *# currPath是查找路径，findName是要查找的文件名* fileList = os.listdir(currPath)  exist = **False  for** file **in** fileList:  **if** file == **'$RECYCLE.BIN' or "System Volume Information"**: *# 回收站和卷标目录，Python没有访问权限，到这会卡住，故跳过* **continue**  path = os.path.join(currPath, file)  **if** os.path.isfile(path) **and** file == findName: *# 如果当前层存在目标文件，return True* print(**"目标文件存在，路径为："**, path, sep=**''**)  **return True  elif** os.path.isdir(path):  **if not** exist: *# exist为True的话，exist就不用更新了* exist = findTxt(path, findName)  **else**:  **return** exist |

2.

|  |
| --- |
| **def getFileByTime(currPath, findTime):** |
| **def** getFileByTime(currPath, findTime): *# currPath是查找路径，findName是要查找的文件创建时间* pathList = [] *# 存放所有符合条件的文件的路径* fileList = os.listdir(currPath)  **for** file **in** fileList:  **if** file == **'$RECYCLE.BIN' or "System Volume Information"**: *# 回收站和卷标目录，Python没有访问权限，到这会卡住，故跳过* **continue** path = os.path.join(currPath, file) *# 记录下文件/文件夹的路径* **if** os.path.isfile(path) **and** \  time.strftime(**"%Y-%m-%d"**, time.localtime(os.path.getctime(path))) == findTime: *# 如果当前层存在目标文件  # 通过绝对路径找文件，获取其创建的时间戳，转换成当地时间，再通过strftime函数将其格式化为“年-月-日”的格式，进行字符串比较* pathList.append(path) *# 相同则存进去* **elif** os.path.isdir(path): *# 如果是个文件夹，那么就递归进去遍历* subMenu = getFileByTime(path, findTime)  **if** len(subMenu) != 0: *# 如果通过递归遍历结果不是空，那么就依次放进列表中* pathList = pathList + subMenu  **return** pathList |

3.

|  |
| --- |
| **def buySauce():** |
| **def** buySauce():  num = 0  money = eval(input())  **if** money < 10 **or** money > 300 **or** money % 10 != 0:  print(**"ERROR"**)  **return  if** money > 50: *# 先看能不能买5瓶* num = num + 7 \* (money // 50) *# 实际瓶数* money = money - (money // 50) \* 50 *# 减去实际花掉的钱* **if** money > 30: *# 以下两个if的逻辑同上* num = num + 4 \* (money // 30)  money = money - (money // 30) \* 30  **if** money >= 10:  num = num + money // 10  **return** num |

4.

|  |
| --- |
| **def reportNum():** |
| **def** reportNum():  num = 1 *# 报到的序号从1开始* index = 0 *# 从1号小朋友开始* [n, k] = eval(input())  children = [i + 1 **for** i **in** range(n)] *# 存放的是小朋友的编号* **while** len(children) != 1:  **if** num % k == 0 **or** num - 10 \* (num // 10) == k:  children.pop(index) *# 表示出局* index = index - 1 *# 出去一个，那么下一个就在现在的位置了，这里是为了和后面的index+1抵消* **if** index + 1 != len(children):  index = index + 1  **else**:  index = 0  num = num + 1  **return** children[0] |

五、实验结果及分析

1.

图形用户界面, 文本, 应用程序

描述已自动生成

2.

图形用户界面, 文本

描述已自动生成

3.

文本

中度可信度描述已自动生成

这个结果是对的，5\*7+0\*4+2\*1=37

4.

文本

中度可信度描述已自动生成

六、实验总结及体会

Python的文件处理已经做的十分完善了，可能除了效率略低一点外就没有缺点了。

此外得益于Python对各种数据类的方法都十分完善，使得在处理一些实际问题建模的时候能够更好的契合常规思路，也就是所谓的“灵活”。

七、教师评语