

人工智能 实验说明文档 - 实验 1

年级		专业（方向）	
学号		姓名	
电话		Email	
开始日期		完成日期	

1. 实验题目

实验 1——python 编程实践入门

2. 实验目的

通过完成四道不同难度、不同需求的 python 编程题目，体会与理解 python 的一些特性和数据结构、数据类型等成分，为使用 python 编程完成后续实验打下一定的基础。

3. 程序设计与测试运行

1) Task1

本任务中的程序要对输入的字符串做判断——如果输入的字符串是“回文字符串”，则打印 True，否则打印 False。对于输入，我们只考虑字母和数字。

根据给出的测试样例，我们了解到需要对输入的字符串做如下操作：将所有英文字母转化为小写（或者大写）、去除字符串中除数字和英文字母以外的所有成分。操作完成后，只需比较该字符串做反转操作后得到的字符串与原字符串是否一致即可。从而编写程序如下：

```
1. s = input("请输入待检验的字符串：\n")
2. s = s.lower()#所有字母转换为小写
```

```

3. s = ''.join(filter(str.isalnum,s))#仅保留英文字母和数字
4. if s == s[::-1]:
5.     print("True")
6. else :
7.     print("False")

```

上面我们使用了 python 中对于字符串的一些操作。该程序测试运行情况如下：

(包含个人信息，图片已经移除)

(包含个人信息，图片已经移除)

(包含个人信息，图片已经移除)

任务 1 完成。

2) Task2

本任务中我们要制作一个简单的石头剪刀布游戏。我们获取两位玩家的输入，并按游戏规则对胜负进行判定，输出结果即可。一局游戏结束后，可以根据玩家意见开启新的对局或退出游戏。

为了让我们的程序足够健壮，我们还是要考虑一些特殊情况。比如玩家需要输入正确的单词（石头：Rock，剪刀：Scissors，布：Paper）才能使得游戏正常进行，再比如玩家在一局游戏结束后必须正确输入“y”或“n”，才能使程序明确下一步的操作。所以对于非法的用户输入，我们应当有所处理。

我们的程序设计如下：

```

1. def print_result(player1_input,player2_input):
2.     if player1_input == player2_input:
3.         print('Ended in a draw!')
4.     elif (player1_input == "Rock" and player2_input == "Scissors") or (player1_input == "Scissors" and player2_input == "Paper") or (player1_input == "Paper" and player2_input == "Rock"):
5.         print('Congratulate Player1!')
6.     else:
7.         print('Congratulate Player2!')
8.
9. def is_not_vaild(player_input):#如果输入非法，则返回 true

```

```

10.     if player_input == "Rock":
11.         return False
12.     elif player_input == "Scissors":
13.         return False
14.     elif player_input == "Paper":
15.         return False
16.     return True
17.
18. s = "y"
19. print('欢迎来到 两人石头剪刀布游戏 ')
20. while s == "y":
21.     s = ""
22.     player1_input = ""
23.     player2_input = ""
24.     while is_not_vaild(player1_input) or is_not_vaild(player2_input):
25.         if player1_input != "":
26.             print("您的输入有误!")
27.             print("请输入: Rock/Scissors/Paper")
28.             player1_input = input("Player1 input:")
29.             player2_input = input("Player2 input:")
30.         print_result(player1_input,player2_input)
31.     while s != "y" and s != "n":
32.         s = input("Try new game?(y/n)")
33.         if s != "y" and s != "n":
34.             print("您的输入有误! ")
35. print("byebye!")

```

测试程序是否正常，我们只需要游玩一下这个游戏：

(包含个人信息，图片已经移除)

上面是用户做了正常输入的情况，对于错误的输入，我们应当提醒用户，并要求其重新输入：

```

Congratulate Player1!
Try new game?(y/n)no
您的输入有误!
Try new game?(y/n)

```

```
D:\python\python.exe "H:/Desktop/20337269_赵一平_实验1/Task 2/main.py"
欢迎来到 两人石头剪刀布游戏
请输入: Rock/Scissors/Paper
Player1 input:AI2022SYSU
Player2 input:Paper
您的输入有误!
请输入: Rock/Scissors/Paper
Player1 input:zhaoyy
Player2 input:ROCK
您的输入有误!
请输入: Rock/Scissors/Paper
```

可以发现我们的游戏程序还是比较健壮的。任务2完成。

3) Task 3

这个任务中，我们需要实现一个可以打印出帕斯卡三角形前 n 行的函数。帕斯卡三角形的数学规律是“每个数等于它上方两数之和”，根据这一数学规律，我们可以根据输入的数字（代表要输出的是该三角形的前 n 行）来生成一个双重列表，然后对每一个位置的数值进行计算。计算完成后，按顺序输出列表即可。

当然，我们还是要考虑一下用户输入的这个代表前 n 行的数字是否是合法的。

设计程序如下：

```
1. def print_triangle(n):
2.     s = [[0 for j in range(i+1)] for i in range(n)]
3.
4.     for i in range(n):
5.         for j in range(i+1):
6.             if j == 0:
7.                 s[i][j] = 1
8.             elif j != i:
9.                 s[i][j] = s[i-1][j-1] + s[i-1][j]
10.            else:
11.                s[i][j] = s[i-1][j-1]
12.        print("下面是帕斯卡三角形前",n,"行")
13.    for i in range(n):
```

```
14.         print(s[i])
15.
16.n = 0
17.while n < 1:
18.    n = int(input('请输入行数: '))
19.print_triangle(n)
```

我们用一个非法数据和一个合法数据简单测试一下：

(包含个人信息，图片已经移除)

任务 3 完成。

4) Task 4

这个任务中，我们主要需要编写一个有指定自定义功能的列表类，从而熟悉 python 中类的定义方式。

这个类里需要包含通过传入的列表进行初始化的构造函数（需要用到深拷贝），然后定义如下五个方法：

添加元素： push(args)，参数元素类型为字符串或整型，将 args 添加到列表末尾；

获取元素： get(num)，参数为整型，从列表中随机获取 num 个元素并打印；

获取列表长度： length()，打印列表长度；

删除元素： delete()，删除列表第一个元素并打印；

清空列表： clear()，清空列表元素。

基本上我们都可以用 list 中的特性实现。但在“获取元素”中，有随机获取指定数量的元素并打印的需求，所以我们引入了 random 库来实现随机获取，同时还对传入的数字做了合法性的判断，来保证程序运行的稳定。

设计程序如下：

```
1. import copy
2. import random
```

```

3. class MyList():
4.     memory = []
5.     def __init__(self):
6.         return None
7.     def __init__(self, list_input):
8.         self.memory = copy.deepcopy(list_input)
9.     def push(self, args):
10.        self.memory.append(args)
11.    def get(self, num):
12.        copy_list = copy.deepcopy(self.memory)
13.        n = int(num)
14.        if n < 0 or n > len(self.memory):
15.            print("参数非法, 已输出所有元素")
16.            print(copy_list)
17.        else:
18.            print(random.sample(copy_list, num))
19.    def length(self):
20.        print(len(self.memory))
21.    def delete(self):
22.        print(self.memory.pop(0))
23.    def clear(self):
24.        self.memory = []

```

接着我们编写一段用到此类的程序进行测试：

```

1. temp_list = MyList([123, 'abc'])
2. print("列表当前情况:", temp_list.memory)
3. temp_list.push(456)
4. print("列表当前情况:", temp_list.memory)
5. temp_list.push(789)
6. print("列表当前情况:", temp_list.memory)
7. temp_list.get(2)
8. print("列表当前情况:", temp_list.memory)
9. temp_list.get(-1)
10. print("列表当前情况:", temp_list.memory)
11. temp_list.length()
12. print("列表当前情况:", temp_list.memory)
13. temp_list.delete()
14. print("列表当前情况:", temp_list.memory)
15. temp_list.delete()
16. print("列表当前情况:", temp_list.memory)
17. temp_list.clear()
18. print("列表当前情况:", temp_list.memory)

```

运行之后，我们得到结果如下图（下一页）：

(包含个人信息，图片已经移除)

可以看到这个类的五种方法都得到了实现。任务4完成。