

实验 4 问题的描述—数据结构（2）

1. 实验目的

- （1）了解如何用较为复杂数据结构描述问题
- （2）掌握用 Python 语言去实现数据结构进行

北航6系 计算思维导论 计算机教学实验中心 版权所有

2. 实验任务

实验任务 4-1 建立一个类，并编写代码对其进行测试。要求调用该类能完成以下问题：

- (1) 能够构造任意半径的圆
- (2) 提供类方法计算圆的面积和周长。
- (3) 提供类方法能够建立以该圆为截面的圆柱，只要用户输入一个圆柱的高，就能调用该类计算出该圆柱体的体积和表面积。
- (4) 编写测试程序，对你编写的类进行测试

实验目的：

1. 理解类的定义
2. 会定义类方法
3. 会调用所定义类

参考代码：

```
import math

class Circle: #定义一个圆形类

    def __init__(self,radius):#构造函数
        self.r=radius

    def girth(self):#计算圆周长
        g=2*math.pi*(self.r)
        return g

    def area(self):#计算圆面积

    def cylinderArea(self,h):#计算圆柱表面积

    def cylinderVolume(self,h):#计算圆柱体积

if __name__=="__main__":#测试开始
```

实验任务 4-2（与实验任务 4-3 选做一题）在用计算机输入数学公式的时候，常常遇到括号的匹配问题，尤其是当公式非常复杂的时候，往往担心漏输入的情况，请你设计一个程序，选用合适的数据结构来解决该问题。当出现不匹配的情况时，能给予相应的提示。

判断包含有括号{, [, <, (,), >,], }的字符串是否是合法匹配。

例如以下是合法的括号匹配：

() , [] , (()) , ([]), ()[] , ()[]

以下是不合法括号匹配的：

(, [,] ,)(, ([]), ([() , {()}

程序要求：提示用户输入一行字符串（长度范围为 1~200）。如果字符串中括号匹配是合法的输出“yes”，不合法的输出“no”。

实验目的：

学习使用合适的数据结构来解决相关问题。

实验指导：

1. 理解用堆栈的数据结构来解决该问题。

括号匹配问题是堆栈的一个典型应用。由于每一个右括号，必定是与在其之前的所有未被匹配的左括号中最靠右的一个进行匹配。若我们按从左到右的顺序读入字符串，并把所有遇到的左括号都放入堆栈中等待匹配，若在读入的过程中遇到一个右括号，则栈顶的左括号一定是与其匹配的左括号，否则，则表示在其之前不存在未被匹配的左括号，匹配失败。

2. 如果你对列表的属性和方法比较熟悉，可以将列表当作堆栈使用。此时会用到列表的 `append()`、`pop()` 方法。这两个方法具体使用的例子如下：

```
>>> st = [3, 4, 5]
>>> st.append(6)
>>> st.append(7)
>>> st
[3, 4, 5, 6, 7]
>>> st.pop()
7
>>> st
[3, 4, 5, 6]
>>> st.pop()
6
>>> st.pop()
5
>>> st
[3, 4]
```

3. 你也可以用课堂老师所给出的 `stack` 类来完成此问题，类的定义如下：

`#Stack.py`

`class Stack:`

`#创建一个空栈`

`def __init__(self):`

```
        self.items=[]

    #元素入栈
    def push(self,item):
        self.items.append(item)

    #元素出栈
    def pop(self):
        return self.items.pop()

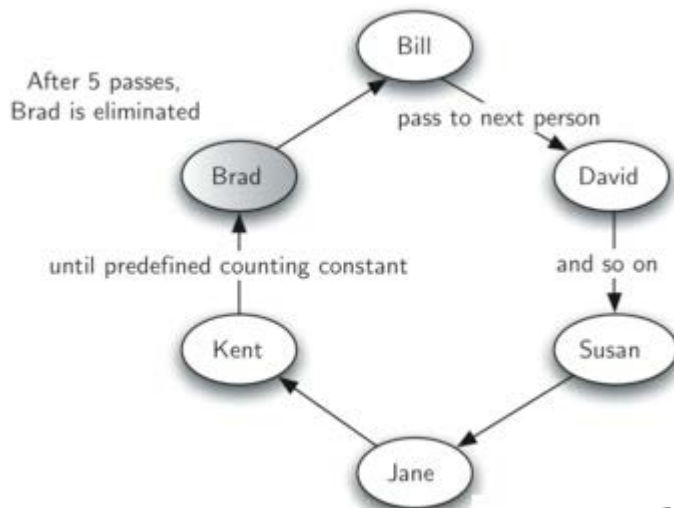
    #返回栈顶的元素
    def peek(self):
        return self.items[len(self.items)-1]

    #判断栈是否为空
    def isEmpty(self):
        return self.items==[]

    #返回栈的大小
    def size(self):
        return len(self.items)
```

北航6系 计算思维导论 计算机教学实验中心 版权所有

实验任务 4-3（与实验任务 4-2 选做一题） 观察一种儿童游戏，叫烫手的山芋（hotpotato），在这个游戏中（图 2），孩子们排成一圈，把手里的东西一个传一个，在某种情形下，停止传递，手上拿着烫手的山芋的人就要被清出来，其他的人继续玩，直到只剩一个人。请你设计一个程序，模拟该游戏过程。

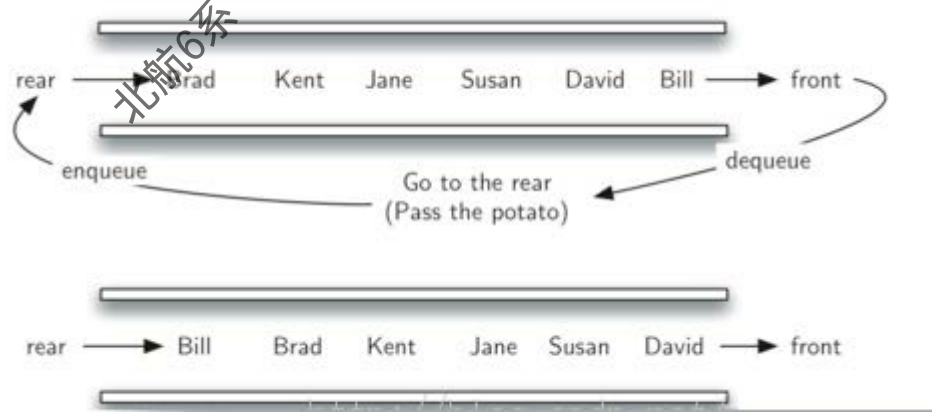


实验目的：

学习使用合适的数据结构来解决相关问题。

实验指导：

（1）问题分析：我们引入一个烫手的山芋的模拟过程，参数是一个名字列表和一个常数 `num`。`num` 用来计数，最后函数返回经多轮计数后，剩下的最后一个人的名字。为了模拟这个圆圈，我们使用队列。假定开始拿着山芋的孩子站在队伍的前端，一经传出山芋后，模拟程序只需要简单地把这个孩子移出队列，然后再将他加入尾部，然后他在尾部再逐步前移，直到再次轮到他。经过 `num` 次出队入队之后，前端的孩子最终被完全清出队列，然后剩余的人继续游戏，直到最后一个。



2. 如果你对列表的属性和方法比较熟悉，可以将列表当作队列使用。此时会用到列表的 `append()`、`pop()` 方法。这两个方法具体使用的例子如下：

```
>>> queue = ['a', 'b', 'c']
>>> queue.append('d')
>>> queue.append('e')
>>> queue
['a', 'b', 'c', 'd', 'e']
>>> queue.pop(0)
'a'
>>> queue
['b', 'c', 'd', 'e']
>>> queue.pop(0)
'b'
>>> queue
['c', 'd', 'e']
```

3. 你也可以用课堂老师所给出的 queue 类来完成此问题，类的定义如下：

#queue_class.py

class Queue:

def __init__(self):
 self.items = []

def isEmpty(self):
 return self.items == []

def enqueue(self,item):
 self.items.insert(0,item)

def dequeue(self):
 return self.items.pop()

def size(self):
 return len(self.items)