MODUL PRAKTIKUM

ALGORITMA DAN STRUKTUR DATA

INF1008

Penyusun :

Naufal Azmi Verdikha, M.Eng.

Teknik Informatika
Fakultas Sains & Teknologi
Universitas Muhammadiyah Kalimantan Timur

Samarinda, 2019

Praktikum 7:
Struktur Data Dasar (2)

Pokok Bahasan:

- ❖ Queue.
- ❖ Deque.
- ❖ Unordered List.
- ❖ Ordered List.

Tujuan Pembelajaran:

- ✓ Memahami implementasi *Queue* pada struktur data *Python.*
- ✓ Memahami implementasi *Deque* pada struktur data *Python.*
- ✓ Memahami implementasi *Unordered List* pada struktur data *Python.*
- ✓ *Memahami implementasi Ordered List* pada struktur data *Python.*

Queue:

Percobaan & Latihan 7.1

Jalankan class dan perintah berikut!

```
1  class Queue:
2      def __init__(self):
3          self.items = []
4
5      def isEmpty(self):
6          return self.items == []
7
8      def enqueue(self, item):
9          self.items.insert(0,item)
10
11     def dequeue(self):
12         return self.items.pop()
13
14     def size(self):
15         return len(self.items)
16
17 q=Queue()
18
19
20 q.enqueue(4)
21 q.enqueue('dog')
22 q.enqueue(True)
23
24 print(q.size())
25 print(q.isEmpty())
26 print(q.enqueue(8.4))
27 print(q.dequeue())
28 print(q.dequeue())
29 print(q.size())
```

Soal :

a) Berikan tampilan output dari perintah diatas!

b) Jelaskan fungsi baris kode 17 hingga 29!

Percobaan & Latihan 7.2

Buatlah implementasi simulasi *Hot Potato* berikut:

```
1  from pythonds.basic.queue import Queue
2
3  def hotPotato(namelist, num):
4      simqueue = Queue()
5      for name in namelist:
6          simqueue.enqueue(name)
7
8      while simqueue.size() > 1:
9          for i in range(num):
10             simqueue.enqueue(simqueue.dequeue())
11
12         simqueue.dequeue()
13
14     return simqueue.dequeue()
15
16 print(hotPotato(["Bill","David","Susan","Jane","Kent","Brad"],7))
```

Soal :

a) Berikan tampilan output dari perintah diatas!

b) Ubahlah nilai 7 pada baris 16 (print(...,7)) dengan nilai 6, kemudian berikan tampilan dan berikan analisanya!

```
16 print(hotPotato(["Bill","David","Susan","Jane","Kent","Brad"],6))
```

Percobaan & Latihan 7.3

Buatlah implementasi simulasi *Printing Tasks* berikut:

```
1  from pythonds.basic.queue import Queue
2
3  import random
4
5  class Printer:
6      def __init__(self, ppm):
7          self.pagerate = ppm
8          self.currentTask = None
9          self.timeRemaining = 0
10
11     def tick(self):
12         if self.currentTask != None:
13             self.timeRemaining = self.timeRemaining - 1
14             if self.timeRemaining <= 0:
15                 self.currentTask = None
16
17     def busy(self):
18         if self.currentTask != None:
19             return True
20         else:
21             return False
22
23     def startNext(self,newtask):
24         self.currentTask = newtask
25         self.timeRemaining = newtask.getPages() * 60/self.pagerate
26
27 class Task:
28     def __init__(self,time):
29         self.timestamp = time
30         self.pages = random.randrange(1,21)
31
32     def getStamp(self):
33         return self.timestamp
34
35     def getPages(self):
36         return self.pages
37
38     def waitTime(self, currenttime):
39         return currenttime - self.timestamp
40
41
42 def simulation(numSeconds, pagesPerMinute):
```

```
43
44      labprinter = Printer(pagesPerMinute)
45      printQueue = Queue()
46      waitingtimes = []
47
48      for currentSecond in range(numSeconds):
49
50        if newPrintTask():
51           task = Task(currentSecond)
52           printQueue.enqueue(task)
53
54        if (not labprinter.busy()) and (not printQueue.isEmpty()):
55          nexttask = printQueue.dequeue()
56          waitingtimes.append( nexttask.waitTime(currentSecond))
57          labprinter.startNext(nexttask)
58
59        labprinter.tick()
60
61      averageWait=sum(waitingtimes)/len(waitingtimes)
62      print("Average Wait %6.2f secs %3d tasks remaining."%(averageWait,printQueue.size())
63
64  def newPrintTask():
65      num = random.randrange(1,181)
66      if num == 180:
67          return True
68      else:
69          return False
70
71  for i in range(10):
72      simulation(3600,5)
```

Soal :

a) Berikan tampilan ouput dari perintah diatas dan hasil analisa!
b) Sebutkan rata-rata waktu terlama dan tercepat dari hasil ouput!

Deque:

Percobaan & Latihan 7.4

Jalankan class dan perintah berikut!

```
1  class Deque:
2      def __init__(self):
3          self.items = []
4
5      def isEmpty(self):
6          return self.items == []
7
8      def addFront(self, item):
9          self.items.append(item)
10
11     def addRear(self, item):
12         self.items.insert(0,item)
13
14     def removeFront(self):
15         return self.items.pop()
16
17     def removeRear(self):
18         return self.items.pop(0)
19
20     def size(self):
21         return len(self.items)
22
23 d=Deque()
24 print(d.isEmpty())
25 d.addRear(4)
26 d.addRear('dog')
27 d.addFront('cat')
28 d.addFront(True)
29 print(d.size())
30 print(d.isEmpty())
31 d.addRear(8.4)
32 print(d.removeRear())
33 print(d.removeFront())
```

Soal :

a) Berikan tampilan output dari perintah diatas!

b) Jelaskan fungsi baris kode 23 hingga 33!

Percobaan & Latihan 7.5

Buatlah implementasi Palindrome-Checker berikut:

```
from pythonds.basic.deque import Deque

def palchecker(aString):
    chardeque = Deque()

    for ch in aString:
        chardeque.addRear(ch)

    stillEqual = True

    while chardeque.size() > 1 and stillEqual:
        first = chardeque.removeFront()
        last = chardeque.removeRear()
        if first != last:
            stillEqual = False

    return stillEqual

print(palchecker("lsdkjfskf"))
print(palchecker("radar"))
```

Soal: Berihkan hasil ouput dan analisa!

Unordered:

Percobaan & Latihan 7.6

Jalankan class dan perintah berikut!

```
class Node:
    def __init__(self,initdata):
        self.data = initdata
        self.next = None

    def getData(self):
        return self.data

    def getNext(self):
        return self.next

    def setData(self,newdata):
        self.data = newdata

    def setNext(self,newnext):
        self.next = newnext


class UnorderedList:

    def __init__(self):
        self.head = None

    def isEmpty(self):
        return self.head == None

    def add(self,item):
        temp = Node(item)
        temp.setNext(self.head)
        self.head = temp

    def size(self):
        current = self.head
        count = 0
        while current != None:
            count = count + 1
            current = current.getNext()

        return count

```

```
41      def search(self,item):
42          current = self.head
43          found = False
44          while current != None and not found:
45              if current.getData() == item:
46                  found = True
47              else:
48                  current = current.getNext()
49
50          return found
51
52      def remove(self,item):
53          current = self.head
54          previous = None
55          found = False
56          while not found:
57              if current.getData() == item:
58                  found = True
59              else:
60                  previous = current
61                  current = current.getNext()
62
63          if previous == None:
64              self.head = current.getNext()
65          else:
66              previous.setNext(current.getNext())
67
68  mylist = UnorderedList()
69
70  mylist.add(31)
71  mylist.add(77)
72  mylist.add(17)
73  mylist.add(93)
74  mylist.add(26)
75  mylist.add(54)
76
77  print(mylist.size())
78  print(mylist.search(93))
79  print(mylist.search(100))
80
81  mylist.add(100)
82  print(mylist.search(100))
83  print(mylist.size())
84
85  mylist.remove(54)
86  print(mylist.size())
87  mylist.remove(93)
88  print(mylist.size())
89  mylist.remove(31)
90  print(mylist.size())
91  print(mylist.search(93))
```

Soal :

a) Berikan tampilan output dari perintah diatas!
b) Jelaskan fungsi baris kode 68 hingga 91!

Ordered:

Percobaan & Latihan 7.7

Jalankan class dan perintah berikut!

```
class Node:
    def __init__(self,initdata):
        self.data = initdata
        self.next = None

    def getData(self):
        return self.data

    def getNext(self):
        return self.next

    def setData(self,newdata):
        self.data = newdata

    def setNext(self,newnext):
        self.next = newnext


class OrderedList:
    def __init__(self):
        self.head = None

    def search(self,item):
        current = self.head
        found = False
        stop = False
        while current != None and not found and not stop:
            if current.getData() == item:
                found = True
            else:
                if current.getData() > item:
                    stop = True
                else:
                    current = current.getNext()

        return found

    def add(self,item):
        current = self.head
        previous = None
        stop = False
        while current != None and not stop:
            if current.getData() > item:
                stop = True
            else:
                previous = current
                current = current.getNext()

        temp = Node(item)
        if previous == None:
            temp.setNext(self.head)
            self.head = temp
        else:
            temp.setNext(current)
            previous.setNext(temp)

    def isEmpty(self):
        return self.head == None

```

```
60      def size(self):
61          current = self.head
62          count = 0
63          while current != None:
64              count = count + 1
65              current = current.getNext()
66
67          return count
68
69
70  mylist = OrderedList()
71  mylist.add(31)
72  mylist.add(77)
73  mylist.add(17)
74  mylist.add(93)
75  mylist.add(26)
76  mylist.add(54)
77
78  print(mylist.size())
79  print(mylist.search(93))
80  print(mylist.search(100))
```

Soal :

a) Berikan tampilan output dari perintah diatas!

b) Jelaskan fungsi baris kode 70 hingga 80!

Laporan Resmi:

1. Buatlah summary dan analisa dari **Percobaan & Latihan** pada pratikum ini.
2. Berikan kesimpulan dari praktikum ini.