

Ivanovitz A.A.R

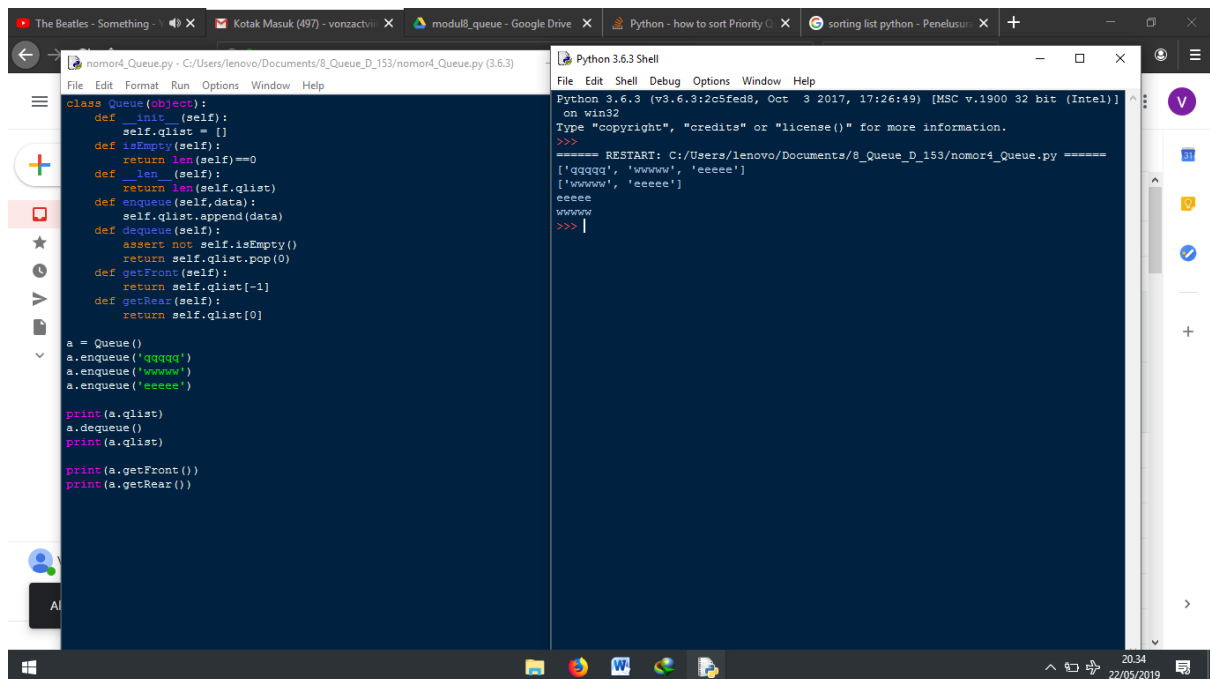
L200170153

Kelas D

Modul 8

Nomor 4

Queue



The screenshot shows a Windows desktop with a code editor and a Python 3.6.3 Shell. The code editor displays the implementation of a Queue class and its usage. The Python Shell shows the execution of the code, including the restart command and the output of the Queue operations.

```
class Queue(object):
    def __init__(self):
        self.qlist = []
    def isEmpty(self):
        return len(self)==0
    def __len__(self):
        return len(self.qlist)
    def enqueue(self, data):
        self.qlist.append(data)
    def dequeue(self):
        assert not self.isEmpty()
        return self.qlist.pop(0)
    def getFront(self):
        return self.qlist[-1]
    def getRear(self):
        return self.qlist[0]

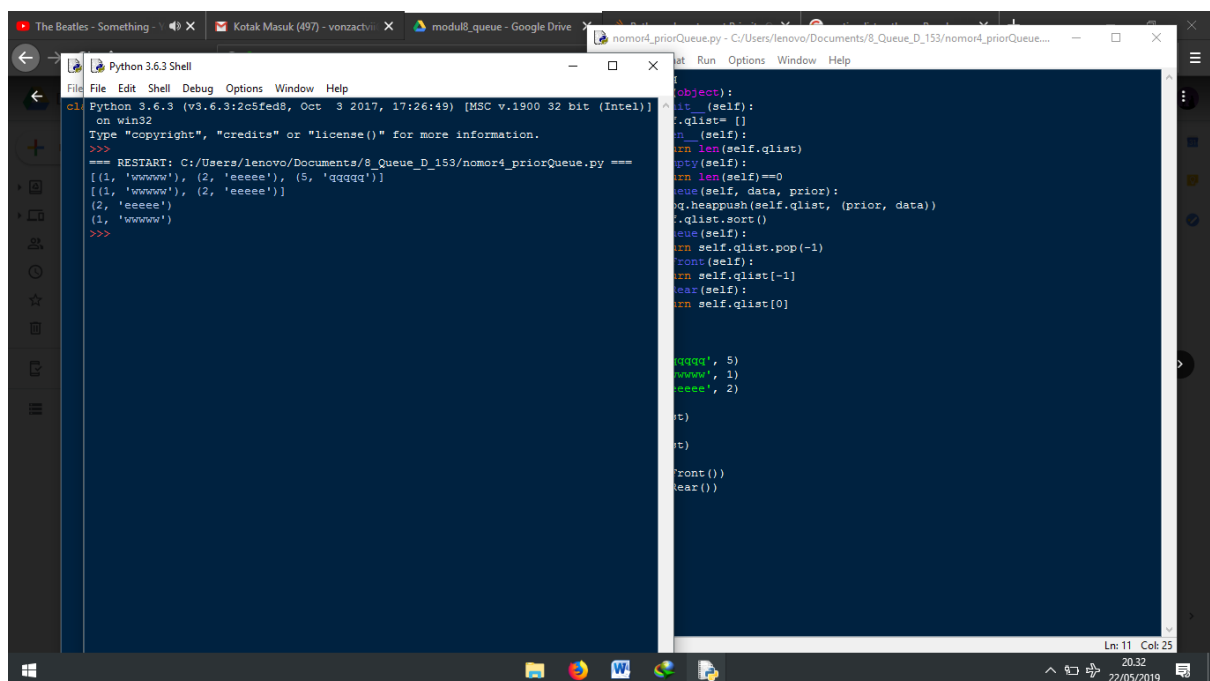
a = Queue()
a.enqueue('qqqqq')
a.enqueue('wwwww')
a.enqueue('eeeeee')

print(a.qlist)
a.dequeue()
print(a.qlist)

print(a.getFront())
print(a.getRear())
```

```
Python 3.6.3 (v3.6.3:2c5fed8, Oct 3 2017, 17:26:49) [MSC v.1900 32 bit (Intel)]
on win32
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Users/lenovo/Documents/8_Queue_D_153/nomor4_Queue.py =====
['qqqqq', 'wwwww', 'eeeeee']
['wwwww', 'eeeeee']
eeeeee
wwwww
>>>
```

Priority Queue



The screenshot shows a Windows desktop with a code editor and a Python 3.6.3 Shell. The code editor displays the implementation of a Priority Queue class and its usage. The Python Shell shows the execution of the code, including the restart command and the output of the Priority Queue operations.

```
class PriorityQueue(object):
    def __init__(self):
        self.qlist = []
    def __len__(self):
        return len(self.qlist)
    def isEmpty(self):
        return len(self)==0
    def enqueue(self, data, priority):
        self.qlist.append((data, priority))
        self.qlist.sort()
    def dequeue(self):
        return self.qlist.pop(-1)
    def getFront(self):
        return self.qlist[-1]
    def getRear(self):
        return self.qlist[0]

pq = PriorityQueue()
pq.enqueue('qqqqq', 5)
pq.enqueue('wwwww', 1)
pq.enqueue('eeeeee', 2)

print(pq.qlist)
pq.dequeue()
print(pq.qlist)
pq.getFront()
pq.getRear()
```

```
Python 3.6.3 (v3.6.3:2c5fed8, Oct 3 2017, 17:26:49) [MSC v.1900 32 bit (Intel)]
on win32
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Users/lenovo/Documents/8_Queue_D_153/nomor4_priorQueue.py =====
[(1, 'wwwww'), (2, 'eeeeee'), (5, 'qqqqq')]
[(1, 'wwwww'), (2, 'eeeeee')]
(2, 'eeeeee')
(1, 'wwwww')
>>>
```

## Nomor 5

