

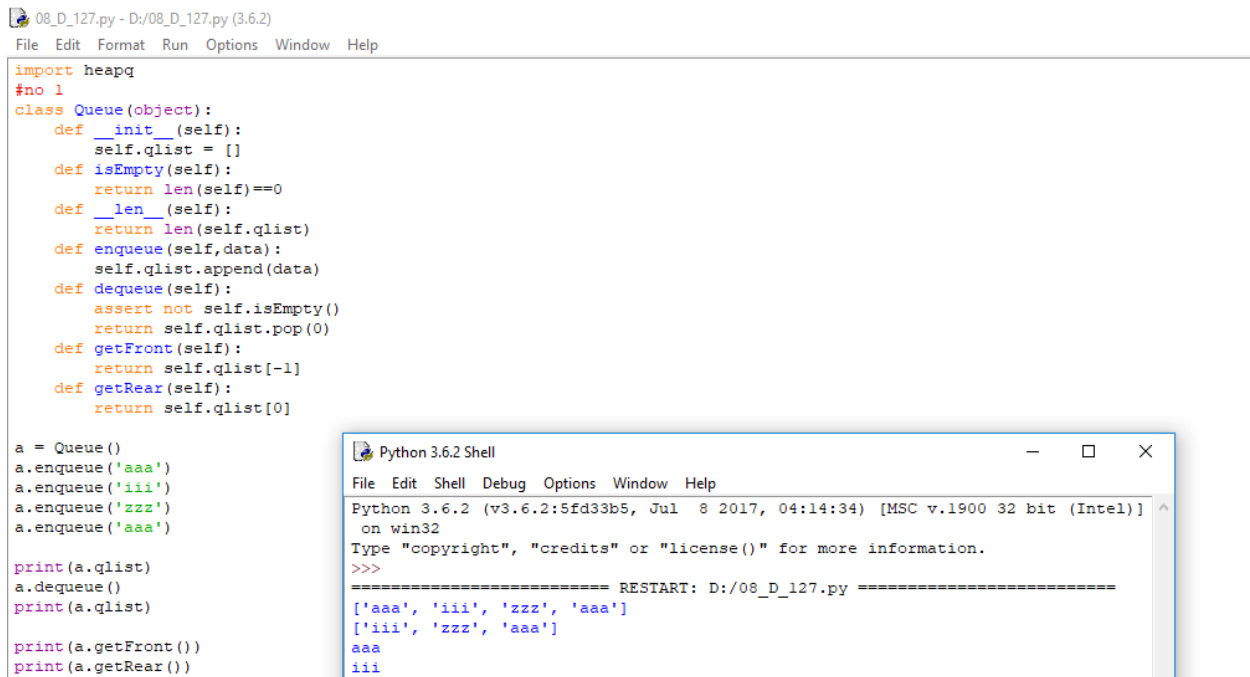
NAMA : AGATHA FEBIANANDA P

NIM : L200170127

KELAS : D

MODUL : 8

QUEUE



The image shows a Python 3.6.2 IDE window titled '08_D_127.py - D:/08_D_127.py (3.6.2)'. The code defines a `Queue` class using a list to store elements. The class methods include `__init__`, `isEmpty`, `len`, `enqueue`, `dequeue`, `getFront`, and `getRear`. Below the class definition, an instance `a` of the `Queue` class is created, and several operations are performed: enqueuing 'aaa', 'iii', 'zzz', and 'aaa'; printing the queue list; dequeuing an element; and printing the front and rear elements.

```
import heapq
#no 1
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('aaa')
a.enqueue('iii')
a.enqueue('zzz')
a.enqueue('aaa')

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

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

The Python 3.6.2 Shell window shows the execution of the code. It displays the restart message and the output of the operations: the queue list `['aaa', 'iii', 'zzz', 'aaa']`, the front element `aaa`, and the rear element `iii`.

```
Python 3.6.2 Shell
File Edit Shell Debug Options Window Help
Python 3.6.2 (v3.6.2:5fd33b5, Jul 8 2017, 04:14:34) [MSC v.1900 32 bit (Intel)]
on win32
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: D:/08_D_127.py =====
['aaa', 'iii', 'zzz', 'aaa']
aaa
iii
```

08_D_127.py - D:/08_D_127.py (3.6.2)

File Edit Format Run Options Window Help

```
#no 2
class Prior(object):
    def __init__(self):
        self.qlist= []
    def __len__(self):
        return len(self.qlist)
    def isEmpty(self):
        return len(self)==0
    def enqueue(self, data, prior):
        heapq.heappush(self.qlist, (prior, data))
        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]

a = Prior()

a.enqueue('aaa', 2)
a.enqueue('iii', 8)
a.enqueue('zzz', 3)

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

Output :

```
[(2, 'aaa'), (3, 'zzz'), (8, 'iii')]
[(2, 'aaa'), (3, 'zzz')]
```

```
#no 3
class Priority(object):
    def __init__(self):
        self.qlist= []
    def __len__(self):
        return len(self.qlist)
    def isEmpty(self):
        return len(self)==0
    def enqueue(self, data, prior):
        heapq.heappush(self.qlist, (prior, data))
        self.qlist.sort()
    def dequeue(self):
        return self.qlist.pop(-1)

a = Priority()

a.enqueue('aaa', 9)
a.enqueue('iii', 2)
a.enqueue('zzz', 4)

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

```
[(2, 'iii'), (4, 'zzz'), (9, 'aaa')]
[(2, 'iii'), (4, 'zzz')]
>>> |
```