s-day-5-2-simple-queue-using-class

July 25, 2024

1 stack means last in first out

stack==when you empty the dish washer and put your plates away, you stack them up and then when you need a plate you take from the top of the stack

2 queue means first in last out

queue== We wait in a line for a movie

[1]:

```
#in stack push and pop are done in only top element. so,there we need only one pointer named top.
#in queue enqueue is same as push. so, it's rear pointer.
#     but dequeue is done on first not last so front pointer.
# so 2 pointer named rear and front pointers

queue=[]
queue.append(10)    # in  queue append is known as enqueue
queue.append(20)
queue.append(30)
print(queue)

queue.pop(0)      # in queue pop in known as dequeue and it's must be pop(0) not pop()
print(queue)
```

```
[10, 20, 30]
[20, 30]
```

[5]:

```
queue=[]

class queue:
    def __init__(self):
        self.queue=[]
    def enque(self,value):
        self.queue.append(value)
    def dequeue(self):
        self.queue.pop(0)
    def printqueue(self):
        print(self.queue)

q=queue()
q.enque(10)
q.enque(20)
q.enque(30)

q.dequeue()
q.printqueue()
```

```
[20, 30]
```

[26]:

```
queue=[]
class queue:
    def __init__(self):
        self.queue=[]
    def enqueue(self,value):
        self.queue.append(value)
    def dequeue(self):
        self.queue.pop(0)
    def length(self):
        print("length of queue is",len(self.queue))
    def printqueue(self):
        print(self.queue[0],"<-------front")
        for i in range (1,len(self.queue)-1):
                print(self.queue[i])
        print(self.queue[len(self.queue)-1],"<------rear")


q=queue()
q.enqueue(10)
q.enqueue(20)
q.enqueue(30)
q.enqueue(40)
q.enqueue(50)
q.dequeue()

q.length()
q.printqueue()
```

```
length of queue is 5
10 <-------front
20
30
40
50 <------rear
```

[1]:

```
## using front and rear pointers
queue=[]
class queue:
    def __init__(self):
        self.queue=[]
        self.front=self.rear=-1
    def enqueue(self,value):
        self.queue.append(value)
        if(front==-1):
            rear+=1
            front+=1
        else:
            rear+=1
    def dequeue(self):
        self.queue.pop(0)
        if(front==-1):
            print('queue i underflow')
        else:
            front-=1
    def length(self):
        print("length of queue is",len(self.queue))
    def printqueue(self):
        print(self.queue[0],"<-------front")
        for i in range (1,len(self.queue)-1):
                print(self.queue[i])
        print(self.queue[len(self.queue)-1],"<------rear")


q=queue()
q.enqueue(10)
q.enqueue(20)
q.enqueue(30)
q.enqueue(40)
q.enqueue(50)
q.dequeue()

q.length()
q.printqueue()
```

```
---------------------------------------------------------------------------
UnboundLocalError                         Traceback (most recent call last)
<ipython-input-1-86850fe8cd7b> in <module>
     28 
     29 q=queue()
---> 30 q.enqueue(10)
     31 q.enqueue(20)
     32 q.enqueue(30)
```

```
<ipython-input-1-86850fe8cd7b> in enqueue(self, value)
      7     def enqueue(self,value):
      8         self.queue.append(value)
----> 9         if(front==-1):
     10             rear+=1
     11             front+=1

UnboundLocalError: local variable 'front' referenced before assignment
```

```
[ ]:
```