from pythonds.basic import Stack  
from queue import LifoQueue  
class StackPythonds():  
 def \_\_init\_\_(self):  
 self.stack = Stack()  
   
 def push(self, item):  
 self.stack.push(item)  
   
 def pop(self) -> any:  
 if self.stack.isEmpty():  
 return None  
 return self.stack.pop()  
   
 def peek(self) -> any:  
 if self.stack.isEmpty():  
 return None  
 return self.stack.peek()  
   
 def size(self) -> any:  
 return self.stack.size()  
  
  
class LiFoQueueStack():  
 def \_\_init\_\_(self) -> None:  
 self.stack = LifoQueue()  
   
 def push(self, item):  
 self.stack.put(item)  
   
 def pop(self) -> any:  
 if self.stack.empty():  
 return None  
 return self.stack.get()  
   
 def peek(self) -> any:  
 if self.stack.empty():  
 return None  
 size = self.stack.\_qsize()  
 return self.stack.queue[size-1]  
   
 def size(self) -> any:  
 return self.stack.\_qsize()  
   
test\_items = [  
 10, 3.14, True, False, "Hello", "World", 42,  
 None, [1, 2, 3], (4, 5, 6), {"a": 1, "b": 2},  
 {7, 8, 9}, complex(2, 3), b"bytes", bytearray(b"array"),  
 frozenset([1, 2, 3]), range(5), memoryview(b"abc"),  
 -99, 0.0001  
 ]  
pythondsStack = StackPythonds()  
lifoStack = LiFoQueueStack()  
  
for item in test\_items:  
 pythondsStack.push(item)  
 lifoStack.push(item)  
  
stacks = [pythondsStack,lifoStack]  
for stack in stacks:  
 print(" the stack type is ", stack.\_\_class\_\_)  
 print("the stack size is \n", stack.size())  
 item = stack.peek()  
 print("the peeked item ", item)  
 print("the stack size shall not change", stack.size())  
 item = stack.pop()  
 print("the poped item is ", item)  
 print("the stack size shall decrease by one", stack.size())

the stack type is <class '\_\_main\_\_.StackPythonds'>  
the stack size is   
 20  
the peeked item 0.0001  
the stack size shall not change 20  
the poped item is 0.0001  
the stack size shall decrease by one 19  
 the stack type is <class '\_\_main\_\_.LiFoQueueStack'>  
the stack size is   
 20  
the peeked item 0.0001  
the stack size shall not change 20  
the poped item is 0.0001  
the stack size shall decrease by one 19

from queue import Queue  
from collections import deque  
  
# Queue using Queue  
class QueueExample:  
 def \_\_init\_\_(self):  
 self.queue = Queue()  
   
 def enqueue(self, value):  
 self.queue.put(value)  
   
 def dequeue(self):  
 if self.queue.empty():  
 return None  
 return self.queue.get()  
   
class DequeQueue:  
 def \_\_init\_\_(self):  
 self.queue = deque()  
   
 def enqueue(self, value):  
 self.queue.append(value)  
   
 def dequeue(self):  
 return self.queue.popleft()  
  
test\_items = [  
 10, 3.14, True, False, "Hello", "World", 42,  
 None, [1, 2, 3], (4, 5, 6), {"a": 1, "b": 2},  
 {7, 8, 9}, complex(2, 3), b"bytes", bytearray(b"array"),  
 frozenset([1, 2, 3]), range(5), memoryview(b"abc"),  
 -99, 0.0001  
 ]  
queueQ = QueueExample()  
dequeQ = DequeQueue()  
  
for item in test\_items:  
 queueQ.enqueue(item)  
 dequeQ.enqueue(item)  
  
queues = [queueQ,dequeQ]  
for queue in queues:  
 print(" the queue type is ", queue.\_\_class\_\_)  
 item = queue.dequeue()  
 print("the dequeue item is ", item)

the queue type is <class '\_\_main\_\_.QueueExample'>  
the dequeue item is 10  
 the queue type is <class '\_\_main\_\_.DequeQueue'>  
the dequeue item is 10