

23.2.2025 20:28:21

stack\_implementation.py

Page 1/2

```

1
2 # HSLU / ICS/AI ML : Modul ADS : Algorithmen & Datenstrukturen
3 # Path : uebung02/al/aufgabe03
4 # Version: Sun Feb 23 20:28:21 CET 2025
5
6 from uebung02.al.aufgabe03.empty_stack_exception import EmptyStackException
7 import sys
8
9
10 class StackImplementation:
11     """
12     Stack: a collection of objects that are inserted and removed according
13     to the last-in first-out principle.
14     """
15     # --- nested _Node class: -----
16     class _Node:
17
18         def __init__(self, elem):
19             self._element = elem
20             self._next = None
21
22         def append_node(self, nextNode):
23             self._next = nextNode
24
25         def get_next(self):
26             return self._next
27
28         def get_element(self):
29             return self._element
30
31     # --- stack methods: -----
32
33     def __init__(self):
34         self._top = None
35         self._size = 0
36
37     def __len__(self):
38         return self._size
39
40     def size(self):
41         # TODO: Implement here...
42         pass
43
44     def is_empty(self):
45         # TODO: Implement here...
46         return True
47
48     def top(self):
49         # TODO: Implement here...
50         return None
51
52     def push(self, element):
53         # TODO: Implement here...
54         pass
55
56     def pop(self):
57         # TODO: Implement here...
58         return None
59
60     def printout(self):
61         print("Stack: (", self._to_string(self._top, ""), ")")
62
63     def _to_string(self, node, content):
64         if node == None:
65             return content
66         if not content == "":
67             content += ", "
68         content += str(node.get_element())
69         return self._to_string(node.get_next(), content)
70
71

```

23.2.2025 20:28:21

stack\_implementation.py

Page 2/2

```

72
73 if __name__ == '__main__':
74     stack = StackImplementation()
75     stack.printout()
76     TEST_SIZE = 4
77     for i in range(TEST_SIZE):
78         stack.push(i)
79         stack.printout()
80         if stack.size() != i+1:
81             print("ERROR: Stack.size() != ", i+1)
82             sys.exit()
83         if stack.top() != i:
84             print("ERROR: Stack.top() != ", i)
85             sys.exit()
86     for i in range(TEST_SIZE-1, 0, -1):
87         if stack.pop() != i:
88             print("ERROR: Stack.pop() != ", i)
89             sys.exit()
90     stack.printout()
91     if stack.size() != i:
92         print("ERROR: Stack.size() != ", i)
93         sys.exit()
94     if stack.top() != i-1:
95         print("ERROR: Stack.top() != ", i-1)
96         sys.exit()
97     if stack.pop() != 0:
98         print("ERROR: Stack.pop() != ", 0)
99         sys.exit()
100     stack.printout()
101     if not stack.is_empty():
102         print("ERROR: Stack.empty() != true")
103         sys.exit()
104     if stack.size() != 0:
105         print("ERROR: Stack.size() != 0")
106         sys.exit()
107     try:
108         stack.top()
109         print("ERROR: no EmptyStackException for stack.top()!")
110         sys.exit()
111     except EmptyStackException:
112         pass
113     try:
114         stack.pop()
115         print("ERROR: no EmptyStackException for stack.pop()!")
116         sys.exit()
117     except EmptyStackException:
118         pass
119
120
121 """ Session-Log:
122
123 Stack: ( )
124 Stack: ( 0 )
125 Stack: ( 1, 0 )
126 Stack: ( 2, 1, 0 )
127 Stack: ( 3, 2, 1, 0 )
128 Stack: ( 2, 1, 0 )
129 Stack: ( 1, 0 )
130 Stack: ( 0 )
131 Stack: ( )
132
133 """
134

```

23.2.2025 20:28:21

**empty\_stack\_exception.py**

Page 1/1

```
1
2 # HSLU / ICS/AIML : Modul ADS : Algorithmen & Datenstrukturen
3 # Path   : uebung02/al/aufgabe03
4 # Version: Sun Feb 23 20:28:21 CET 2025
5
6 class EmptyStackException(Exception):
7
8     def __init__(self, err):
9         super().__init__(err)
10
11
```