

605.202: Introduction to Data Structures

E. A. Calderon

Project 1 ADT Supplemental Information

Due Date: July 6, 2021

Dated Turned In: July 6, 2021

Project 1 ADT

DATA: Queue Class (Named QueueSelf.Class) & QStack Class

The list of items below includes methods thought of to use for the project/lab 1 Queue LinkedList Based Stack Class. Not all methods were used, though it was important to have the list ready before coding and running into trouble later on.

METHOD

isEmpty

Input: None
 Preconditions: None
 Process: Check whether queue is empty or isn't empty
 Postconditions: Returns Boolean value
 Output: True/false

Push

Input: There is a char for queue to add
 Preconditions: Char can only be a letter or number
 Process: Add the char into the queue combination
 Postconditions: Char added to queue in a LIFO sequence with other queue
 Output: none

Pop

Input: None
 Preconditions: 1. Stack/queue is not empty
 2. LIFO has been successfully implemented (not FIFO with queue)
 Process: Remove char from the stack/queue system and return removed char
 Postconditions: Return removed char
 Output: Character

Size

Input: None
 Preconditions: None
 Process: Return the number of elements in the queue/stack class
 Postconditions: Return of an integer value regardless of whats in the stack/queue
 Output: Integer

Clearstack

Input: None
 Preconditions: Stack/queue is empty. Since we cant use the library functions I need to make sure I am popping everything. Since pop needs to check stack for empty I have to do the same thing here.
 Process: Remove all cars from the stack/queue
 Postconditions: None
 Output: None

Peek

Input: None

Preconditions: 1. Stack/queue is not empty
2. LIFO has been successfully implemented (not FIFO with queue)

Process: Display the LIFO char entered (need this to compare to other char)

Postconditions Stack/queue is unchanged

Output: Character

DATA: Main project Lab1.Class

The list of items below includes methods thought of to use for the project/lab 1 Main project Lab1.Class. Not all methods were used, though it was important to have the list ready before coding and running into trouble later on.

METHOD

Read in File

Input: Input text file

Preconditions: Input txt file exists in the same location Lab1.Class is

Process: 1. Read character by character of the file to input to stack
2. make sure to know end of line

Postconditions Halt at end of line

Output: Characters

Write out File

Input: 1. Algorithm identified palindrome
2. text verbiage

Preconditions: Algorithm has identified if it is a palindrome

Process: Write out aftermath of algorithm

Postconditions Outputted text file 1. No hard coded paths, 2. In same location of where files are located.

Output: .txt