Assignment 3: Stacks and Queues

CSCI 2270 - CS2: Data Structures
Kos - Summer 2015

Instructions

In this assignment, you will be implementing a stack and a queue using an array, a single linked list, and a double linked list. These data structures will be populated through user input to add or remove items one at a time.

Read the entire assignment before beginning any coding.

In main, present the user with a menu that includes the following:

1. Create Stack
2. Create Queue
3. Exit program

If 1 or 2 is selected, ask the user if they want and array, single linked list, or double linked list implementation.

a) arrays
b) single linked lists
c) double linked lists

Each combination of data structure and implementation should be handled in a separate class (or a function that holds that class). For example, you could have a class called stackArray if the user wants to create a stack using an array. Of you could have a function that will be called if the user wants an array stack; then within this function you will manipulate the stackArray class.

If the user selects “Create stack” in the menu, present the following 4 choices repeatedly:

(1) PUSH (Enter integer for insertion into stack)
(2) POP (Display integer and delete it from stack)
(3) PRINT STACK (Display stack contents without deleting anything, last element first)
(4) Exit program

If the user selects “Create queue”, then the user is given the following 4 choices repeatedly:

(1) ENQUEUE (Enter integer for insertion into queue)
(2) DEQUEUE (Display and delete integer from queue)
(3) PRINT QUEUE (Display queue contents without deleting anything, first element first)
(4) Exit program

If option 4 was selected, perform garbage collection on any stacks or queues you created, then exit the program. You do not have to go back to the first menu like you did in Assignment 2.

All functionality for that choice can be handled in corresponding function or method, including asking for user input.

Use a class for your single and double linked list implementations. For the single linked list, you can implement the class as a double linked list and just set the pointers in the unused direction to NULL.

For the array implementations, choose a reasonable starting size for your array, such as 40. If you would like to implement an array doubling algorithm you are welcome to do so, but it is not a requirement. The array doubling would happen when the stack or queue is full and more memory needs to be allocated. Generate a new array that is double the size of the current array, and copy everything in the stack or queue to the new array.

Hints

If you are unclear on how to start this assignment, here are some steps you could take.

1. In the last assignment you were required to implement a double while loop with two menus. If you had trouble with that, implement the menu selections first. Do now start with any of the data structures, but just have each menu print that you have arrived at the correct option and redisplay the menu again. Make sure that you can traverse the menu options before moving on to implementing the stacks and queues.

2. This assignment required 3 implementations of the each data structure This will mean that you will have multiple classes creating each type of data structure. The possible list of classes that you will need are listed:

    a. ArrayStack
    b. SingleLinkedListStack
    c. DoubleLinkedListStack
    d. ArrayQueue
    e. SingleLinkedListQueue
    f. DoubleLinkedListQueue
    g. SingleNode
    h. DoubleNode

    This is a large list of possible classes, but you have seen quite a few of these before. If it helpful, you may use old code from previous assignments or labs to aid with your implementations here.

Submission Instructions

To submit your work, zip all files together and submit the zip file. All files should include a comment block at the top of the file with your name, instructor’s name, and assignment number.