IT306 Take Home Assignment 5

Assignment Directions

It is mandatory that you attempt this assignment on your own. You may use any printed resources you would like (text, notes, Internet – not recommended!). You may ask your instructor and TA questions on a limited basis. This is **not** a group assignment; however, limited discussion with your classmates is permitted. You may not work with a tutor or receive assistance on this assignment from outside resources. Sharing your work with a classmate or receiving assistance from someone outside of the course is considered a violation of the Mason Honor Code and results in 0 for the assignment in the first attempt and F in the course for the second attempt.

- **LATE ASSIGNMENT IS NOT ACCEPTED after 11:59 p.m. on the due date. Please don't send your late assignments to me or TA via emails as they will be discarded. You need to submit the softcopy to the myMason Portal on or before 11:59 p.m. If not submitted on time, it is classified as LATE.**
- **All the naming conventions used have to be followed the guidelines:**
- **All the variables have to be declared and initialized inside the method before they are used.**
- **All the arrays have to be declared before they are used.**
- **No global variables are allowed to be used in your application class.**
- **All methods must be structured programs.**

Programming:

In all of the following programming exercises, provide proper documentation to your code. Submit your java files with proper naming. Upload your archived (.zip) file.

1- (50 points) Use the java classes uploaded on BB under "Course Content> Module 6> Programming Sample>Implementations of Stack" and complete the following tasks:

   a. develop the following method for classes Stack and LinkedlistStack:
      public Object peek()
      i. Description: Looks at the object at the top of this stack without removing it from the stack.
      ii. Returns: the object at the top of this stack (the last item of the Vector object).
      iii. Throws: EmptyStackException - if this stack is empty.

   b. develop the following method for classes Stack, LinkedListStack, and ArrayStack:
      public int search(Object o)
      i. Description: Returns the 1-based position where an object is on this stack. If the object o occurs as an item in this stack, this method returns the distance from the top of the stack of the occurrence nearest the top

        of the stack; the topmost item on the stack is considered to be at distance 1. The equals method is used to compare o to the items in this stack.

        ii. Parameters: o - the desired object.

        iii. Returns: the 1-based position from the top of the stack where the object is located; the return value -1 indicates that the object is not on the stack.

2- (30 points) Use the java classes uploaded on BB under “Course Content> Module 6> Programming Sample>Implementations of Queue” and complete the following tasks:

    a. develop the following method for classes Queue and LinkedListQueue:
    Object element()

        i. Description: Retrieves, but does not remove, the head of this queue. This method throws an exception if this queue is empty.

        ii. Returns: the head of this queue

        iii. Throws: NoSuchElementException - if this queue is empty

    b. Develop the following method for class ArrayQueue:
    boolean offer(E e)

        i. Description: Inserts the specified element into this queue if it is possible to do so immediately without violating capacity restrictions. When using a capacity-restricted queue, this method is generally preferable to add(Object), which can fail to insert an element only by throwing an exception.

        ii. Parameters: e - the element to add

        iii. Returns: true if the element was added to this queue, else false

        iv. Throws: ClassCastException - if the class of the specified element prevents it from being added to this queue NullPointerException - if the specified element is null and this queue does not permit null elements IllegalArgumentException - if some property of this element prevents it from being added to this queue

3- (20 points) Write the java code to implement a method that receives a mathematical expression as an array of characters and validates it. Complete the isValid method, i.e., the //TODO section in the provided Application.java class on BB. Use the Stack class provided in Java (Java library java.util.Stack.*;) to solve this problem. For help, follow the algorithm explained in module 6- slide #23.