Programming Assignment 3

Due: 22 November 2024

CSC311, Data Structures

Abstract Data Structures, such as Stacks and Queues, are instrumental tools. In this assignment, you will employ both Stacks and Queues to address intriguing challenges.

Objectives

The main objectives of this assignment are:

- Implement Stack ADT
- Use Stack and Queue to solve software problems.
- Write Junit test cases

To complete this assignment, follow the steps listed below:

1. **Prepare the environment**

   Download and import the zip file: PA3.zip for Eclipse:

   (a) Open Eclipse

   (b) Select File, then Import

   (c) Select General category

   (d) Select Existing Projects into Workspace, and then click on Next

   (e) Select the button for "Select Archive File" and browse to/select the zip file.

   (f) Click on Finish. A new project List has been added to the workspace.

2. **Complete the following:**

   - Write code to simulate the Josephus game. You are required to use a queue as we discussed in class.
   - Implement Stack ADT. Use DoublyLinkedList from assignment 2 as the basic internal data structure for the Stack
   - Write code to evaluate a given postfix expression. You are required to use a stack as we discussed in class.

- Write code to convert a given infix expression to a postfix expression. You are required to use a stack as we discussed in class.
- Create test cases for the above methods.

(a) Download and import the following zip file: `Assignment3.zip`. For Eclipse:

  i. Open Eclipse

  ii. Choose "File → Import"

  iii. Select the "General" category

  iv. Pick "Existing Projects into Workspace"

  v. Click on "Next"

  vi. Select the button for "Select Archive File" and browse to/select the zip file.

  vii. Click on "Finish"

(b) This assignment depends on your implementation of DoublyLinkedList, CircularArrayQueue in the assignment 2. Please copy and overwrite the following files and copy other files that you might have created related to the implementation of them

  i. `DoublyLinkedList.java`

  ii. `CircularArrayQueue.java`

(c) The big things that you need to complete:

  i. Write at least 4 test cases in `JosephusTest.java`

  ii. Implement the `order()` method in `Josephus.java`

  iii. Implement the methods in `DLLStack.java`. The base data structure should be DLL.

  iv. Write at least 8 test cases in `PostfixExpressionTest.java`

  v. Implement the `evaluate()` method in `PostfixExpression.java`

  vi. Write at least 8 test cases in `InfixToPostfixTest.java`

  vii. Implement the `convert()` method in `InfixToPostfix.java`

(d) *Note:* You may ignore the following files: `ArrayList.java`, `ArrayListStack.java`

3. **Submission:**

(a) Before submission, be sure to review the following checklist:

  i. Does the program compile different machines? (Programs that don't compile for us will not be graded)

  ii. Do you ever import from `java.util`. If so, be sure you only import allowed components (like `Iterator`, Exceptions, etc.). Unless the assignment specifically mentions it is permissible, you should never include any of java's native data structures.

iii. Does the program meet all required interfaces?

iv. Is the indentation easily readable? You can have Eclipse correct indentation by highlighting all code and select "Source → Correct Indentation".

v. Are comments well organized and concise?

(b) After thoroughly reviewing your code and ensuring that all criteria on the checklist are met, proceed with the outlined procedure to export the files:

i. Open Eclipse, and choose File, then Export

ii. Select General, and then Archive File

iii. Click Next

iv. Click the down arrow for project "List", check the box next to src.

v. Browse for the destination/enter the archive file name as PA3.zip (not .rar)

vi. Click on "Finish"

vii. file PA3.zip List will be created under the destination of your choice.

(c) Double check the zip file has the correct folder structure and ALL the java source code files are under the correct folders. Missing any source file or directory could cause compilation failure. When the project does not compile, you will get 0 point. For example, for the first assignment, the zip file should have folder list/src/csc311 and all the source files under csc311.

(d) To submit, log in to Canvas and upload your PA3.zip file to the designated assignment section. You are welcome to submit multiple times; only the most recent submission will be evaluated for grading.

Grading Criteria

- The `DLLStack` correctly implement all methods and interfaces: 20
- The `order()` works correctly: 20
- The `evaluate()` works correctly: 20
- The `convert()` works correctly: 20
- The 4 good representative test cases for `order()`: 2 points
- The 8 good representative test cases for `evaluate()`: 4 points
- The 8 good representative test cases for `convert()`: 4 points
- Clear concise code with good commenting: 10