

SIT221 –DATA STRUCTURES AND ALGORITHMS

LAB5: SIMPLIFIED STACKS & QUEUE

LAB OBJECTIVE:

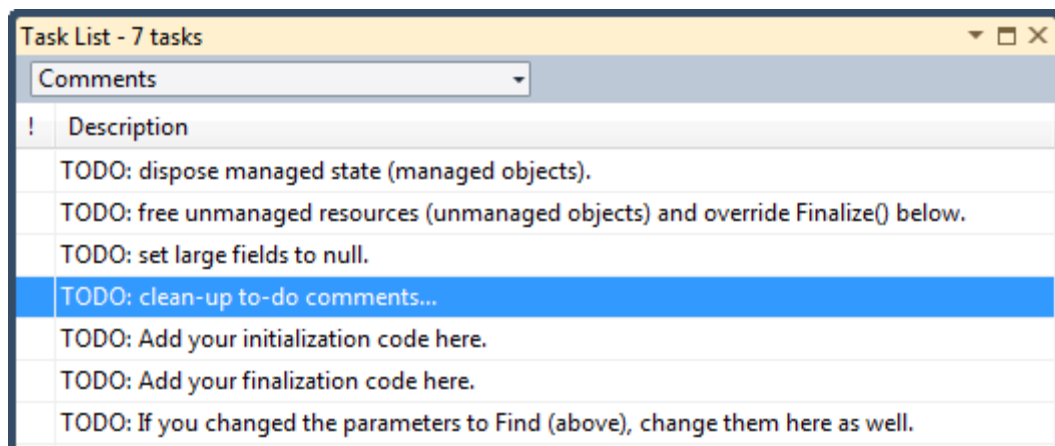
The objective of this lab is to understand how Stacks & Queues work.

SUBMISSION INSTRUCTIONS

Please submit your work to Week 5 assignment folder. Make sure to zip the whole solution in one file and submit it.

PREPARATION

1. Download the template project available in week05 resources folder. The solution has two projects: **DataStructures_Algorithms & Runner** projects.
2. The content of this week prac can be found in **Week05** folder of the project **DataStructures_Algorithms**. You are also given two Runner classes (**Runner05_Task1.cs** and **Runner05_Task2.cs**) to test the implementation of stack and queue.
3. How to find what methods you need to complete? You will not be able to complete until you understand the whole solution, but here is a new way you can try, if you want
In visual studio – You can from the View menu open the Task list and filter comments: you should see something like the below screen:



In Xamarin Studio: You should have similar window:

✓ Tasks			
Comments ▼			
Line	Description	File	Path
47	TODO: Insert - please add your indexof code here	LinkedList.cs	Week03
55	TODO: IndexOf - please add your indexof code here	LinkedList.cs	Week03
65	TODO: RemoveAt - Please add your remove at code here	LinkedList.cs	Week03
127	TODO: GetEnumerator - Add your GetEnumerator Code here	LinkedList.cs	Week03
24	TODO: MoveNext - Please add your MoveNext implementation here	ListEnumerator.cs	Week03

LAB TASKS

1. RPN CALCULATOR

Reverse Polish Notation: is a mathematical notation in which every operator follows all of its operands. For example $5 + 3$ is written as $5\ 3\ +$, $10 * 5$ is written as $10\ 5\ *$ and so on. We want to write an RPN Calculator class that receives as input an RPN expression in a collection object and returns the value of the expression: for example $5\ 1\ 2\ +\ 4\ \times\ +\ 3\ -$, should return 14, for a detailed execution (Step by step) and algorithm details, check here: https://en.wikipedia.org/wiki/Reverse_Polish_notation

1. In the Given project, you will find an **RPN Calculator** class, and **Add** method is already implemented. This is a very simple version of RPN Calculator with one stack to contain operands (numbers). You will notice in the constructor we receive a vector called expression and try to read the elements in each entry and map it either to an operand, so we push it to the operands stack, or an operator so we perform the corresponding operation.
2. You need to complete the three methods: **Subtract, Multiply, and Divide**.
3. Test your implementation using the following expressions:
 - a. $3\ |\ 5\ |\ +$
 - b. $5\ |\ 5\ |\ 5\ |\ 3\ |\ +\ |\ -\ |\ *$
 - c. $1\ |\ 2\ |\ +\ |\ 4\ |\ \times\ |\ 5\ |\ +\ |\ 3\ |\ -$

2. QUEUE DATA STRUCTURE

In this task we want to implement a simple Queue data structure using a linked list class (See the Queue class) with two methods: Enque and Deque methods.

1. `public void Enque(T element) // add element to the end of the list`
2. `public T Deque() //Removes the first element from the queue and returns it.`