Tutorial 6

Introduction

In this tutorial, students are asked to implement Stack and Queue data structure using array and singly linked list.

The **Stack ADT** introduced in the lecture session supports the following operators:

- isEmpty(): Check if the stack is empty
- isFull(): Check if the stack is full (array-based stack only)
- push(itemType newItem): Add a new item to the stack
- pop(): Take out an item from the stack
- peek(): Retrive an item from the stack (without deleting it)

Example 01 demonstrates the implementation of an array-based stack.

The following operators are supported in **Queue ADT**

- isEmpty(): Check if the queue is empty
- isFull(): Check if the queue is full (array-based queue only)
- enqueue(itemType newItem): Add a new item to the queue
- dequeue(): Take out an item from the queue
- peek(): Retrive an item from the queue (without deleting it)

Example 02 demonstrates the implementation of a list-based queue.

Examples

1. Example 01 – ArrayStack implementation

Student should create classes that implements an array stack. Each item of the stack is a String. In this example two classes should be created:

- Class ArrayStack implements the **Stack ADT**
- Class ArrayStackApp which containts the main function, demonstrates how to use the implemented ArrayStack. There are two utility methods in ArrayStackApp:
    - ✓ copyStack(src, des): copy content of stack src to des.
    - ✓ printStack(s): show all items of stack s.

Please refer to class ***ArrayStack*** and ***ArrayStackApp*** in the tutorial example code.

2. Example 02 – List based Queue implementation

Student should create classes that implements a queue using singly linked list. Each item of the queue is an integer. In this example three classes should be created:

- Class QueueNode implements the Queue Node
- Class SLLQueue implements the **Queue ADT**.

- Class SLLQueueApp which contains the main function, demonstrates how to use the implemented list-based Queue. It also contains method
  - printQueue(q) to show all items of the queue q.
  - findMax(q): find the maximum number in the queue q
  - findMin(q): find the minimum number in the queue q

Exercises

1. Exercise 1

Please implement a stack using singly linked list. Each item in this stack is a character.

2. Exercise 2

Please implement a queue using array. Each item in this queue is an integer.