## **Initial Assignment 2 Design**

Nicholas Nguyen

## **UML** diagram

## Notation - stack: MyStack - queue: MyQueue + Notation() + evaluatePostfixExpression(postfixExpr: String): double + convertPostfixToInfix(postfix: String): String + convertinfixToPostfix (infix: String): String + leftParentheses(value: char): boolean + rightParentheses(value: char): boolean

## - previous: MyStack - element: T + isEmpty(): boolean + isFull(): boolean + pop(): T + top(): T + size(): int + push(T): boolean + toString(): String + toString(delimiter: String): String + fill(list: ArrayList<T>): void

# MyQueue<T> - elements: Object[] - front: int - rear: int - capacity: int - size: int + isEmpty(): boolean + isFull(): boolean + dequeue(): T + size(): int + enqueue(e: T): boolean + toString(): String + toString(delimiter: String): String + fill(list: ArrayList<T>): void

## **Pseudocode**

## MyQueue

implements given QueueInterface

will throw underflow and overflow exceptions when necessary

## MyStack

implements given StackInterface

will throw underflow and overflow exceptions when necessary

### Notation

throws exception when format is incorrect for Notation.

Using the queues and stacks created:

- Convert infix → postfix
- Convert postfix → infix
- Evaluate postfix expression