Data Structure and Algorithms

Workshop – Using Queues or Stacks

# Objectives

* Reinforcing what Stacks and Queues can do.
* Using Stacks or Queues to solve problems.

# Introduction

Stacks and Queues are among the most common and important ADTs. Both store collections of items. While stacks allow items to be inserted and removed only at the top (Last In, First Out), queues allow items to be inserted at one end and removed at the other end (First In, First Out). They can help solve a wide range of problems.

In this workshop, you are given a problem and asked to apply Stacks or Queues to solve it.

# Problem

Mathematicians use parentheses (), square brackets [] and braces {} when writing arithmetic expressions. These delimiters must be paired correctly.

* An open delimiter must correspond to a close one.
* Pairs of different types of delimiters must not intersect.

Write a static method that returns whether a given arithmetic expression has balanced delimiters.

**Input 1**: *a* {*b* [*c* (*d* + *e*)/2 - *f*  ] + 1}

**Output 1**: true

**Input 2**: *a* {*b* [*c* (*d* + *e*)/2 - *f*  ] + 1

**Output 2**: false

**Explanation**: there is no respective close bracket for “{“ after “a”

**Input 3**: {*a* [*b* + (*c* + 2)/*d* ] + *e*) + *f* }

**Output**: false

**Explanation**: there is no respective open bracket for the close bracket “)” after “e”

***Hint***: you may need to use the 3 following helper methods.

private static bool IsOpenDelimiter(char ch) {

return ch == '(' || ch == '[' || ch == '{';

}

private static bool IsCloseDelimiter(char ch) {

return ch == ')' || ch == ']' || ch == '}';

}

private static bool IsPaired(char open, char close) {

return (open == '(' && close == ')') ||

(open == '[' && close == ']') ||

(open == '{' && close == '}');

}