# MovingAverage data structure

Interface IMovingAverage is a data structure for storing and calculating average of last N elements added where N is a positive integer. It stores the elements that are sub type of Number class.

MovingAverageImpl.java class provides efficient implementation of the interface.

Assumption

IMovingAverage data structure will store maximum N number of elements.

Technology used

Java, Junit for test cases.

# Instructions to run the program

JDK 1.8 is required for running the jar.

1. Download the ‘PaytmCodingChallenge.jar’.
2. Open terminal and go to the directory where jar file is downloaded.
3. Run following command

java -cp PaytmCodingChallenge.jar com.MovingAverageImplTester

# About the files

1. IMovingAverage.java

Provides the contract for storing and calculating average of last N elements added.

* Implementation classes should provide two "standard" constructors: a void (no arguments) constructor which creates an empty IMovingAverage with some default value of N and a constructor with a single argument of type positive integer to define N.
* Elements in the IMovingAverage are stored in FIFO order. when (N+1)th element is added, first element is removed from the data structure.
* Implementing class should ensure that total number of elements in the data structure should not go beyond N.
* Average is calculated using last N elements. When size of the data structure is less than N, actual size is used for average calculation.
* Elements stored can be accessed through Iterator interface.
* Null is not allowed in the data structure.

1. MovingAverageImpl.java :

* It uses ArrayDeque internally to maintain last N elements added.
* Default value of N is 5.
* If there is no element in the data structure, average is returned as ‘0’.

ArrayDeque is used because it has O(1) time complexity for insertion/deletion operations and it is faster implementation of Queue. We are calculating running sum of the elements when elements are added. Hence, time complexity of getting average is also O(1). I have tested it against stream of 10000000 integers and it is performing faster than LinkedList.

1. MovingAverageImplTest.java:

Junit test cases for testing ‘add’ and ‘getAverage’ methods of MovingAverageImpl class.

Methods included are

* Test add method with number of elements less than N.
* Test add method with number of elements equals to N.
* Test add method with number of elements greater than N.
* Test getAverage method for empty data structure.
* Test getAverage method for negative value of N.
* Test getAverage method with number of elements less than N where element type is integer.
* Test getAverage method with number of elements equals to N where element type is integer.
* Test getAverage method with number of elements greater than N where element type is integer.
* Test getAverage method with default value of N.
* Test getAverage method with negative integers.
* Test getAverage method with large integers.
* Test getAverage method with elements of type double.

1. MovingAverageImplTester.java:

Caller class that tests the functionality for MovingAverageImpl class.

1. Javadoc files:

Contains documentation for all the classes and interfaces.