README - v2 - LinkedList -> MyStack

This program is currently being maintained and has recently been updated to Version 2 which still provides a class to aid in sorting a list of numbers from the scanned input. However, the program has now been updated to utilize a Stack algorithm as opposed to a LinkedList. Therefore, numbers provided by a user are now saved into a sorted stack. This program also shows examples of code **reuse** by using Stack and Collections.sort()

Features:

- Takes the input from a user, and then the list is stored from the smallest to the largest (ascending order).
- The list is stored using a Stack, updated from using a LinkedList from is .
- As with previous iterations of this code, this program also shows examples of code reuse such as using the Stack class and Collections.sort().
    - **The Stack class** provides built-in methods for pushing, popping, and peeking elements, making it easy to implement a LIFO (Last-In-First-Out) data structure without manually handling the underlying logic.
    - **Collections.sort()** provides a convenient way to sort elements in a data structure (such as a Stack or List) using the Timesort algorithm, which is efficient and optimized for various input sizes.

Other Updates

- Documentation and comments within the program have been updated to reflect algorithm changes.
- UML diagram updated to reflect changes.

Running Instructions

1. Compiling the Program - To compile the program, navigate to the folder where `MyStack.java` is located, and run the following command in your terminal:

```bash
javac MyStack.java
java MyStack.java
```

2. To run javadocs -Running javadocs for all classes, including private ones, use the following command:

```bash
javadoc -d docs MyStack.java
```