1. **Classes**

|  |
| --- |
| DataStructureException |
|  |
|  |

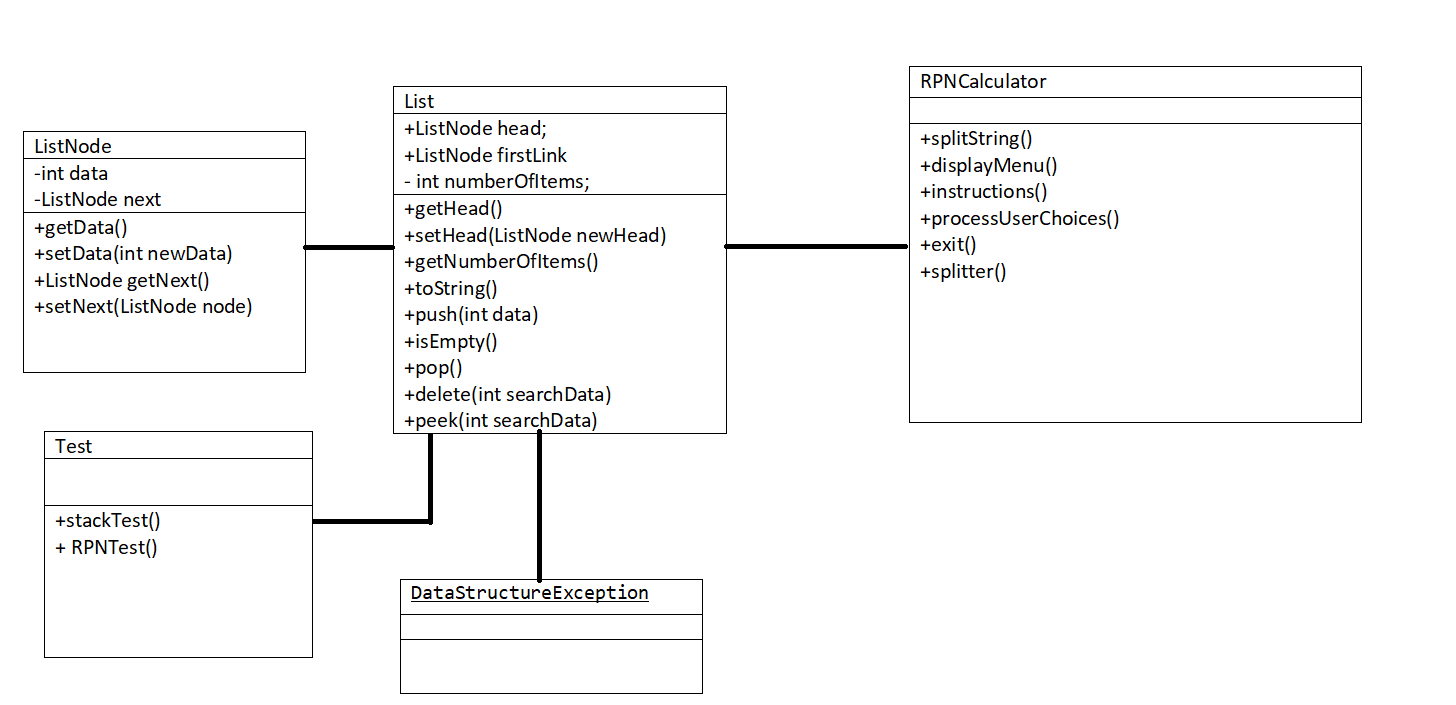
|  |
| --- |
| List |
| +ListNode head;  +ListNode firstLink  - int numberOfItems; |
| +getHead()  +setHead(ListNode newHead)  +getNumberOfItems()  +toString()  +push(int data)  +isEmpty()  +pop()  +delete(int searchData)  +peek(int searchData) |

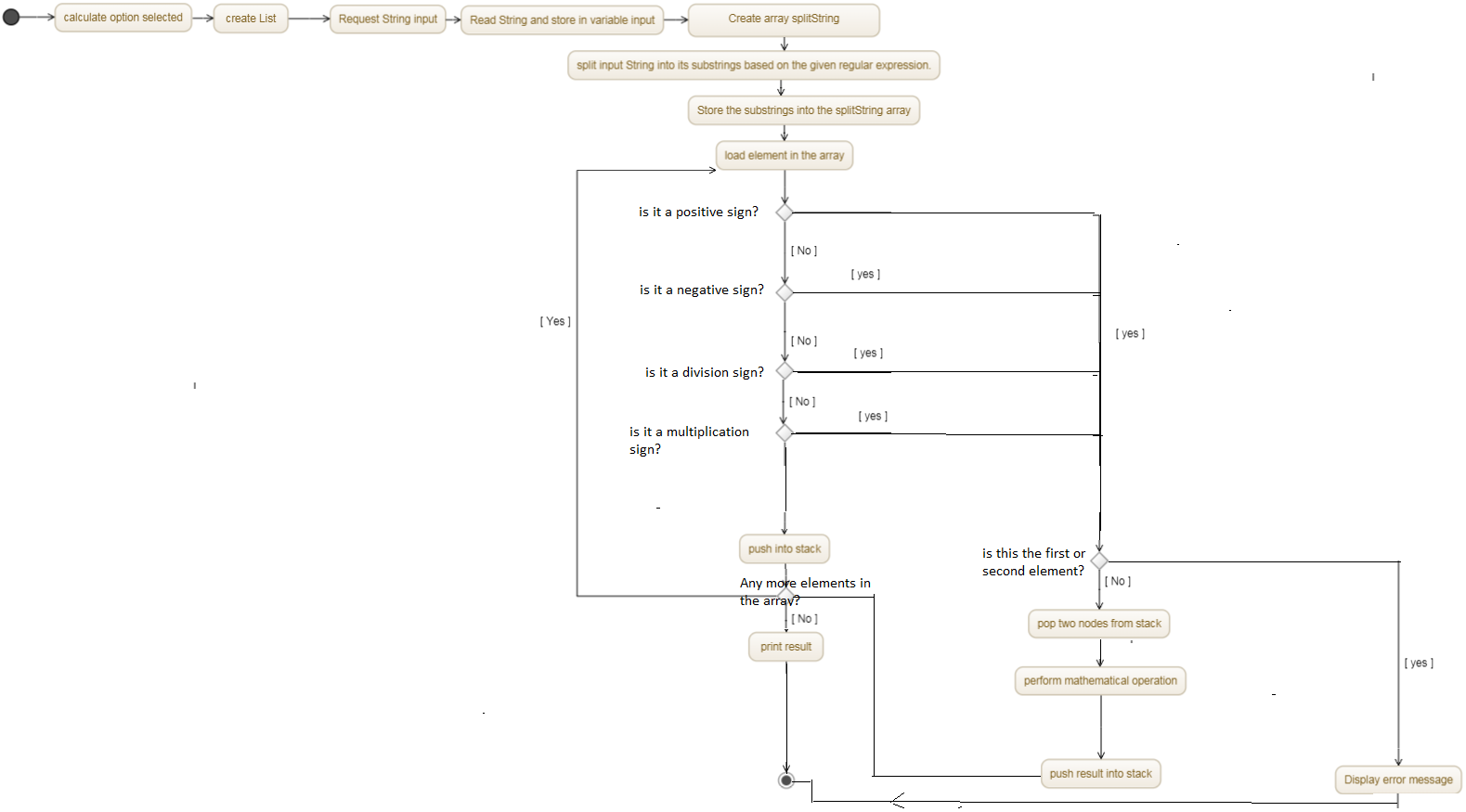
|  |
| --- |
| ListNode |
| -int data  -ListNode next |
| +getData()  +setData(int newData)  +ListNode getNext()  +setNext(ListNode node) |

|  |
| --- |
| Test |
|  |
| +stackTest()  + RPNTest() |

|  |
| --- |
| RPNCalculator |
|  |
| +splitString()  +displayMenu()  +instructions()  +processUserChoices()  +exit()  +splitter() |

1. **Class Diagram**

****

1. **Pseudo code (or activity diagram) for the main method**

**A summary of the requirements – saying which you were able to tackle and which were successful**

|  |  |  |
| --- | --- | --- |
| **Requirements** | **Attempted** | **Completed(Y=yes,N= no)** |
| **Documentation** |  |  |
| Report & Designs (pseudo code and/or flowcharts), selfevaluation. |  | **Y** |
| Test plan and completed test sheets |  | **Y** |
| **Code** |  |  |
| Listnode and List classes (including List delete method): |  | **Y** |
| Stack: Push, Pop and other stack method(s) |  | **Y**  **(I had the methods of my stack class in the list class)** |
| Manipulation of strings, i.e. splitting into tokens |  | **Y** |
| Process RPN expressions |  | **Y** |
| User interface, including input validation |  | **I manage to complete all input validation aside for validation when user inputted a mathematical string which does not terminate;** 4,6,2,+.  **Instead of prompting the user with an error message, the program prints 4,8.** |
| Optional requirements | **I have managed to implement a user defined class for handling exceptions and the division operation.** |  |

1. **Test plan**

Test number/date/version: 29/01/18 ……………………………………….……..

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Description** | **Test Data** | **Expected result** | **Worked?** |
| Push node onto an empty stack, then print node. | Data= 8 | Code completes, node is outputted with correct data (8) with correct number of items in stack | Y |
| Push single node onto a stack with a single node, then print nodes | Data=6 | Code completes, node is outputted with correct data (6) and is placed before the previous node with correct number of items in stack | Y |
| Push multiple nodes to a stack | 5 items pushed on stack [data =4,8,2,3,2] | Code completes, node is outputted with correct data (4,8,2,3,2) and in correct order in which the data was pushed into the stack with correct number of items in stack. | Y |
| Pop node from an empty stack | none | Outputs empty stack message; Empty Stack: Cannnot be popped! | Y |
| Pop single node from existing stack | Data=2; | Last node that was pushed into the stack will be popped out of stack and existing values in stack displayed. | Y |
| Pop multiple node from existing stack | Data=3,2; | Last two nodes that was pushed into the stack was removed according to order they was pushed in( LIFO) with the data value being displayed accordingly (3,2). | Y |
| Finding node with a specific value from existing stack. | Data=6; | Stack is searched for data value 6 and value displayed. | Y |
| Finding node with a specific value which does not exist from existing stack. | Data= 100 | Outputs not found message; Not found: 100. | Y |
| Delete various nodes with specific values which are placed in various sections of the stack(start , middle, end). | Data=8,6,8 | Nodes will be deleted and all existing value left in stack will be displayed. | Y |
| Deleting a non existent number in the stack | Data=99; | Outputs not found message; Not found: 99 Cannot be deleted. | Y |
| RPN calculation | 15,7 1,1,+,- /,3,\*,2,1 1,+,+,- | 5 | Y |
| RPN calculation | 4,6,2,+ | Outputs error as stack is not empty after answer is popped. | N  (4,8 )is left in stack |
| RPN calculation | 4,+ | Outputs error message;  DataStructureException: Empty Stack: Cannnot be poped! | Y |
| RPN calculation | 9 4 - 10 5 - \* | 25 | Y |
| RPN calculation | 68a 0 - | Outputs error message; NumberFormatException:  For input string: "68a”. | Y |
| RPN calculation | %^& | Outputs error message; NumberFormatException:  For input string: "%" | Y |