Seoul National University

Data Structure

Fall 2015, Kang

Programming Assignment 1 : Lists, Stacks and Queues (Chapter 4)

Due: Oct. 8, 11:00 am, submit at class

**Reminders**

* The points of this homework add up to 100.
* Like all homeworks, this has to be done individually.
* Lead TA: Minsoo Jung ([qtyp456987@gmail.com](mailto:qtyp456987@gmail.com))
* Write a program in Java.
* Do not use Java Collection Framework.

1. How to submit the programming assignment
2. Create a JAR file including ‘src’ folder that contains your sources files, and excluding ‘release’ folder. (Refer to ‘1 – Introduction.pptx’ in the first lab session)

\* We will run your Main class in the JAR file to grade your programming assignments. Before submitting the JAR file, please check if your Main class in the JAR file works normally.

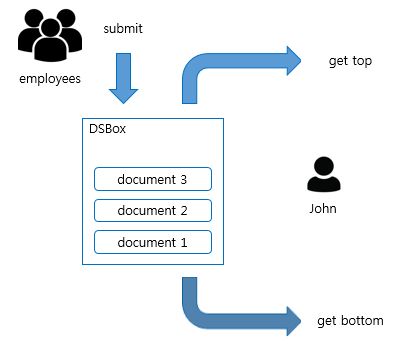
1. Compress the JAR file with a readme file and the project folder. If you don’t use Eclipse, include source files into the compressed file. The readme file need to include your student id, name, and how to execute your program as well as something else TAs need to know.
2. The name of the compress file (zip file) should be ‘PA\_##\_(StudentID).zip’ where ‘##’ is the programming assignment ID, and (StudentID) is your student ID.

ex) PA\_01\_2015-12345.zip

* 01: the first programming assignment
* 2015-12345: your student ID

1. Submit the compress file to the eTL (<http://etl.snu.ac.kr/>).
2. Problem

In a company, John is in charge of document processing. The following figure depicts the document processing of the company. In the office, there is a document submission box (DSBox) where employees in the company submit a document. The employees add new documents to DSBox and pile them up on the top of DSBox.



When John gets a document from DSBox for document processing, he can only get the document that is at the top or bottom of DSBox. In addition, John can see only 1) the name of the document that is at the top or bottom of DSBox, and 2) the number of documents kept in DSBox. If John tries to get or see a document from the DSBox without any document, DSBox prints the message “Box is Empty”.

Fill in the ‘SingleLinkedList.java’ and ‘DSBox.java’ in ‘PA\_01’ java project. Several rules that you **must** follow are as follows:

* Make your program run through Main class importing DSBox implemented in the first step. The main class should handle inputs and outputs using standard I/O in JAVA. (input from a keyboard, output to a monitor)

1. ADT of Data Structure
2. submit

|  |
| --- |
| Function |
| void submit(String document) |
| Description |
| * Employees submit a new document to DSBox. * Add the new document at the top of DSBox. * Parameter ‘String document’ means the document name that employees submit in DSBox. |

1. get\_top

|  |
| --- |
| Function |
| String get\_top() |
| Description |
| * John gets the document at the top of DSBox. * Remove the document at the top of DSBox. * If there are no documents in DSBox, print the message “Box is empty”. |

1. get\_bottom

|  |
| --- |
| Function |
| void get\_bottom() |
| Description |
| * John gets the document at the bottom of DSBox. * Remove the document at the bottom of DSBox. * If there are no documents in DSBox, print the message “Box is empty”. |

1. view\_top

|  |
| --- |
| Function |
| String view\_top() |
| Description |
| * John sees the document at the top of DSBox. * The return value means the name of the document at the top of DSBox. * If there are no documents in DSBox, print the message “Box is empty”. |

1. view\_bottom

|  |
| --- |
| Function |
| String view\_bottom() |
| Description |
| * John sees the document at the top of DSBox. * The return value means the name of the document at the top of DSBox. * If there are no documents in DSBox, print the message “Box is empty”. |

1. size

|  |
| --- |
| Function |
| int size() |
| Description |
| * John sees the number of the documents kept in DSBox. * The return value means the number of the document kept in DSBox. |

1. Specification of I/O

The program should accept only the inputs that are below and print the pertinent outputs. You **must** use standard I/O operations in Java, not file operations.

1. submit

|  |  |
| --- | --- |
| Input | Output |
| submit (document name) | SUBMIT: (document name) |
| Description | |
| * (document name) is the name of the document that employees submit. | |

1. get\_top

|  |  |
| --- | --- |
| Input | Output |
| get\_top | GET: (document name) |
| Description | |
| * (document name) is the name of the document at the top of DSBox. * If there are no documents in DSBox, output is ‘Box is empty’. | |

1. get\_bottom

|  |  |
| --- | --- |
| Input | Output |
| get\_bottom | GET: (document name) |
| Description | |
| * (document name) is the name of the document at the bottom of DSBox. * If there are no documents in DSBox, output is ‘Box is empty’. | |

1. view\_top

|  |  |
| --- | --- |
| Input | Output |
| view\_top | TOP: (document name) |
| Description | |
| * (document name) is the name of the document at the top of DSBox. * If there are no documents in DSBox, output is ‘Box is empty’. | |

1. view\_bottom

|  |  |
| --- | --- |
| Input | Output |
| view\_bottom | BOTTOM: (document name) |
| Description | |
| * (document name) is the name of the document at the bottom of DSBox. * If there are no documents in DSBox, output is ‘Box is empty’. | |

1. size

|  |  |
| --- | --- |
| Input | Output |
| size | CURRENT SIZE: (# of documents) |
| Description | |
| * (# of documents) is the number of the documents kept in DSBox. | |

1. Sample Input

submit DMLab

size

view\_bottom

get\_top

view\_top

1. Sample Output

SUBMIT: DMLab

CURRENT SIZE: 1

BOTTOM: DMLab

GET: DMLab

Box is empty