

Seoul National University

Data Structure

Fall 2015, Kang

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

Due: Oct. 8, 11:00 am, Submit to eTL

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)
- Write a program in Java.
- Do not use Java Collection Framework.

1. How to submit the programming assignment

- 1) Create a JAR file including 'src' folder that contains your sources files, but without '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 correctly.

- 2) 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 needs to include your student id, name, how to execute your program, and any other information TA needs to know.
- 3) The name of the compressed 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

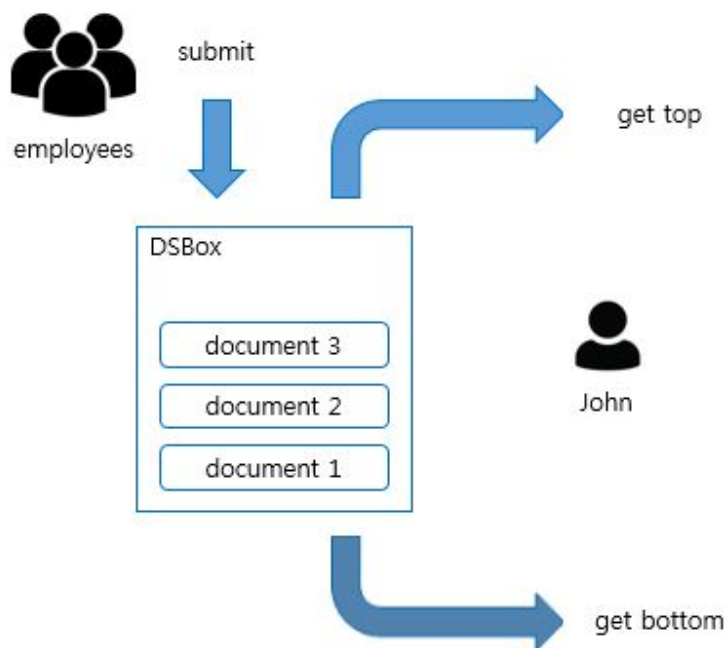
⇒ 01: the first programming assignment

⇒ 2015-12345: your student ID

- 4) Submit the compressed file to eTL.

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 empty DSBox, 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 which uses 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)

3. ADT of Data Structure

1) submit

Function
void submit(String document)
Description
<ul style="list-style-type: none">- 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.

2) get_top

Function
String get_top()
Description
<ul style="list-style-type: none">- 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".

3) get_bottom

Function
void get_bottom()
Description
<ul style="list-style-type: none">- 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".

4) view_top

Function
String view_top()
Description
<ul style="list-style-type: none">- 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".

5) view_bottom

Function
String view_bottom()
Description
<ul style="list-style-type: none">- 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".

6) size

Function
int size()
Description
<ul style="list-style-type: none">- John sees the number of the documents kept in DSBox.- The return value means the number of the document kept in DSBox.

4. Specification of I/O

The program should accept only the inputs listed below and print the listed 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.	
Example Input	Example Output
submit DMLab	SUBMIT: DMLab

2) 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'.	
Example Input	Example Output
get_top	GET: DMLab

3) 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'.	
Example Input	Example Output
get_bottom	GET: DMLab

4) view_top

Input	Output
view_top	TOP: (document name)
Description	
<ul style="list-style-type: none"> - (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'. 	
Example Input	Example Output
view_top	TOP: DMLab

5) view_bottom

Input	Output
view_bottom	BOTTOM: (document name)
Description	
<ul style="list-style-type: none"> - (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'. 	
Example Input	Example Output
view_bottom	BOTTOM: DMLab

6) size

Input	Output
size	CURRENT SIZE: (# of documents)
Description	
<ul style="list-style-type: none"> - (# of documents) is the number of the documents kept in DSBox. 	
Example Input	Example Output
size	CURRENT SIZE: 1

5. Sample Input

submit DMLab

size

view_bottom

get_top

view_top

6. Sample Output

SUBMIT: DMLab

CURRENT SIZE: 1

BOTTOM: DMLab

GET: DMLab

Box is empty