

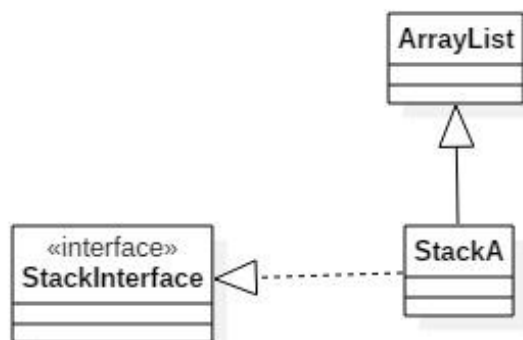
**GIT Department of Computer Engineering**  
**CSE 222/505 - Spring 2017**  
**Homework 4**  
**Due date: March 23 2017 – 23:55**

**SCENARIO:**

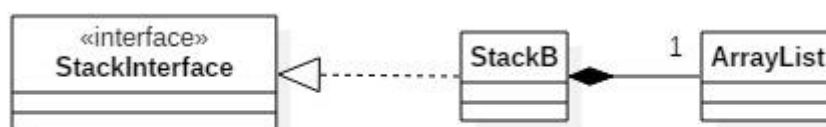
**Q1:**

Create an interface, StackInterface and implement it with four different Stack classes as shown below. Implement push, pop, isEmpty and size methods.

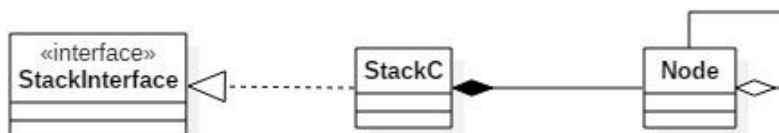
a)



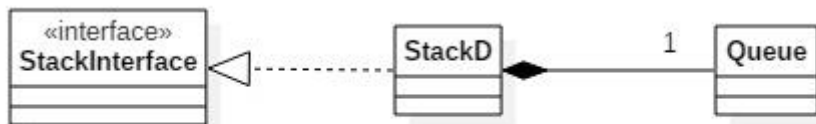
b)



c)



d)



Test:

Read test.csv file which may have lines of integers, floating points, characters, or strings. Create four different stacks (StackA, StackB, ...) for each line. Create testResult\_1.csv file from your stacks adding the size of the stack as the first column of each line. (The size of your result csv file will be x4 of the size of your input csv file.)

Write detailed analysis in your report comparing your four different stack classes.

## Q2:

1. Implement a myQueue class which extends KWLinkedList class (in your book). Implement a reverse method in myQueue which reverses the queue. Do not create any new node while reversing.
2. Write a reverseQueue method which gets a Queue as a parameter and reverses the queue recursively.

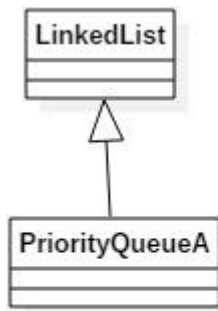
Test:

Read test.csv file which may have lines of integers, floating points, characters, or strings. Create testResult\_2.csv file which includes the content of test.csv file with reversed lines and reversed columns. (Use myQueue objects for all inversion operations.)

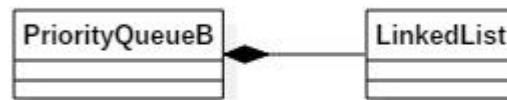
Do the same test with the reverseQueue method.

## Q3 (Bonus):

Define two priorityQueue classes as shown below. Implement size, isEmpty, deleteMin (Deletes and returns the highest priority element. Lower values have higher priority.) and insert methods.



a)



b)

Test:

Read test.csv file which may have lines of integers, floating points, characters, or strings. Create two different queues (`PriorityQueueA`, `PriorityQueueB`) for each line. Create testResult\_3.csv file from your queues. (The size of your result csv file will be x2 of the size of your input csv file.)

#### RESTRICTIONS:

- Write a different mainClass for each question. (Give proper names.)
- Create an IntelliJ project and submit only one project.
- Don't use any other third part library

#### GENERAL RULES:

- For any question firstly use course news forum in moodle, and then the contact TA.
- You can submit assignment one day late and will be evaluated over twenty percent (%20).
- Register [github student pack](#) and create private project and upload your projects into github.
- Your appeals are considered over your github project process.

#### TECHNICAL RULES:

- Use given CSE222-VM to develop and test your homeworks (your code must be working on CSE222-VM), CSE222-VM download link will be given on Moodle.
- Implement clean code standards in your code;
  - o Classes, methods and variables names must be meaningful and related with the functionality.
  - o Your functions and classes must be simple, general, reusable and focus on one topic.
  - o Use standart java code name conventions.

#### REPORT RULES:

- Add all [javadoc](#) documentations for classes, methods, variables ...etc. All explanation must be meaningful and understandable.

- You should submit your homework code, javadoc and report to Moodle in a studentid\_hw#.tar.gz file.
- Use the given homework format including selected parts:

Detailed system requirements	
The Project usecase diagrams (extra points)	
Class diagrams	
Other diagrams	
Problem solutions approach	
Test cases	X
Running command and results	X

#### GRADING :

- No error handling : -50
- No javadoc documentation : -50
- No report : -90
- Disobey restrictions : -100
- **Cheating : -200**
- Your solution is evaluated over 100 as your performance.

#### CONTACT :

Nur Banu Albayrak