

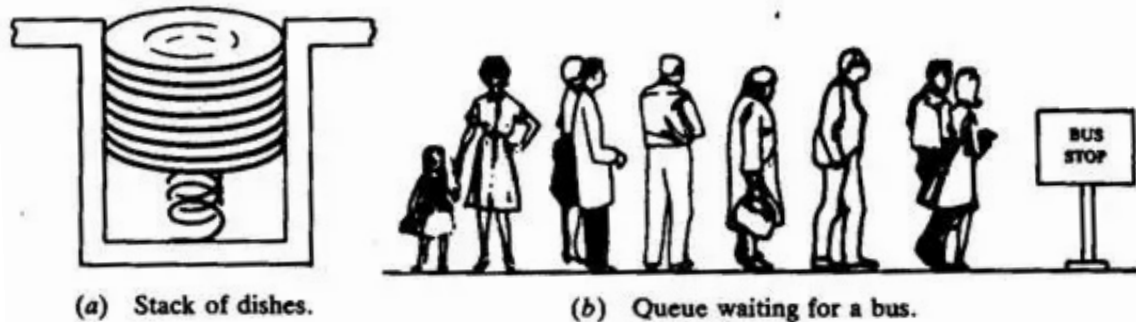
## QUIZ 5

**Issue Date : 02.06.2023 - Friday**

**Due Date : 05.06.2023 - Monday (23:59:59)**

**Advisor : R.A. Görkem AKYILDIZ**

**Programming Language : Java 8 (Oracle)**



### 1 Introduction

In this quiz, you are expected to gain practice on some of the custom generic data structures and self referential classes. So far you have dealt with some static array structures and some built-in collections of Java. Normally, static array structures is enough to store data while programming but it has some flaws, the most important flaw of it is array's size must be defined beforehand which means it always takes up room as its given size, so, it consumes same amount of space whether it is full or empty. If someone wants to shrink or increase the size, he/she must move the entire array to newly created array. Java collections take role at this part, it employs some clever solutions to this issue. For instance, linked-list, it is a structure that expands and shrinks immediately during removals and additions, for sure it has a cost as "link" but if someone needs a simply expandable and shrinkable data structure, linked-list is one of the best choices. The question is, what is linked-list and how it overcomes this problem. Firstly, instead of occupying a chunk of memory, linked-list occupies the memory element by element, which means each element of linked-list can be at anywhere, but for sure it comes with a problem, how to access these fractions, the answer is using the "links". Each element has a link to element that is next to it according to imaginary whole, and in some implementations they also have a link to its previous, which is doubly linked-list structure. Moreover, if first element of that list is assumed to be next element of the last element of the list, it is called as circular linked list. So, as they occupies more size to store these links, due to not occupying a chunk for elements that are not used, it saves memory especially for the scenarios that always have additions and removals. Moreover, Java collections also provides two more data structures as stacks and queues, their structures are LIFO and FIFO accordingly, stack depends on LIFO structure which means last-in-first-out, it can be imagined as a pile of clothes where you can only get the topmost cloth and put to topmost of that pile, on the other hand, queue depends on FIFO structure which means first-in-first-out, it can be imagined as "fair" real life queues such as in bus-stops and dining halls etc.

## 2 Implementation

You are requested to implement a generic and dynamic (with self-referencing nodes) circular doubly linked-list structure and use it as underlying data structure for your generic and dynamic stack and generic and dynamic queue to solve following problems. Note that, for queue, you must add to tail and remove from head while for stack you must both add to and remove from tail.

## 3 Problems

### 3.1 Integer Base Conversion

Normally, humankind are using base ten while computers are using base two, and it causes some problems during interaction between humankind and computers, so, your task is employing a mechanism to convert numbers from base ten to base two. You must use stacks for solving this problem. You are allowed to use only one stack, any other variables, arrays etc. are forbidden.

### 3.2 Balanced Parenthesis

Say that there is an expression that contains some kind of parenthesis which are '(', ')', '{', '}', '[', and ']'. For calling an expression as valid, each opening parenthesis must be closed with appropriate parenthesis -balanced- (which means '(' must be closed with ')', '{' must be closed with '}', and '[' must be closed with ']'). Moreover, there cannot be any closing parenthesis before opening parenthesis, and also two parenthesis cannot nest each other. Which means "()" is invalid expression while "[]" is valid. So, your task is checking if given expression is valid or not. You must use stacks for solving this problem. You are allowed to use only one stack, any other variables, arrays etc. are forbidden. Moreover, there may be some extra characters in addition to these six, you can simply ignore them during this check.

### 3.3 Palindrome

Palindrome is a word (or sentence -excluding white space and punctuation characters-) that remains the same in backwards (case insensitive), such as racecar, noon, Ey Edip Adana'da pide ye. So, your task is checking if given word or sentence is palindrome or not. You must use stacks for solving this problem. You are allowed to use only one stack, any other variables, arrays etc. are forbidden. Note that, please do not filter punctuation marks one by one, use a general approach to discard punctuation marks and white spaces for getting full credit.

### 3.4 Counting up to N in binary

In this problem, your task is counting from 1 to n (which will be given in decimal to you) in binary manner. You must use queues for solving this problem. You are allowed to use only one queue, any other variables, arrays etc. are forbidden.

## 4 Restrictions

- Your code must be able to compiled and executed on our department's developer server (dev.cs.hacettepe.edu.tr).
- You must obey given submit hierarchy and get score (1 point) from the submit system.
- Your code must be clean, do not forget that main method is just driver method that means it is just for making your code fragments to run, not for using them as main container, create classes and methods in necessary situations but use them as required. Moreover, use the four pillars of Object-Oriented Programming (Abstraction, Encapsulation, Inheritance, Polymorphism) if there is such a need, remember that your code must satisfy Object-Oriented Programming Principles, also you can benefit from exceptions and even if create your own exception class if you need any.
- You are encouraged to use lambda expressions which are introduced with **Java 8**.
- You can benefit from Internet sources for inspiration but do not use any code that does not belong to you.
- You can discuss high-level (design) problems with your friends but do not share any code or implementation with anybody.
- You must use JavaDoc commenting style for this project, and you must give brief information about the challenging parts of your code, do not over comment as it is against clean code approach. Design your comments so that some wants to read your code can easily understand what is going on. You can check [here](#) to access Oracle's own guide about JavaDoc Sytle.
- Do not miss the submission deadline.
- Source code readability is a great of importance. Thus, write READABLE SOURCE CODE, comments, and clear MAIN function. This expectation will be graded as "clean code".
- Use UNDERSTANDABLE names to your variables, classes, and functions regardless of the length. The names of classes, attributes and methods should obey Java naming convention. This expectation will be graded as "coding standards".
- You can ask your questions through course's piazza group, and you are supposed to be aware of everything discussed in the piazza group. General discussion of the problem is allowed, but **DO NOT SHARE** answers, algorithms, source codes and reports.
- All assignments must be original, individual work. Duplicate or very similar assignments are both going to be considered as cheating.

## 5 Execution and Test

Your code must be compiled and executed under **Java 8** and dev.cs.hacettepe.edu.tr. If your code does not compile and execute under developer server, then you will be graded as 0 for code part even if it works on your own machine. Sample run command is as follows:

Compilation: javac Main.java

Run: java Main input.txt output.txt

## 6 Grading

Task	Point
Integer Base Conversion	25
Balanced Parenthesis	25
Palindrome	25
Counting up to N in binary	25
Total	100

### 6.1 Score Multipliers

There will be two major overall multipliers which are going to be scaled over 1 and they will be multiplied by your grade, they are Correct Submit Hierarchy Multiplier, Design Multiplier.

- Correct Submit Hierarchy Multiplier: If you do not obey the submit hierarchy (you have to score exactly 1 point at submit system, otherwise it will not accepted as correct hierarchy even if your hierarchy fits to format), you will be penalized with 20% of point deduction.
- Design Multiplier: **You must use your own generic and dynamic stack, generic and dynamic queue, generic and dynamic circular doubly linked-list, and generic node structures. Any other structures, such as built-in structures of Java (arrays, lists etc.) are strictly forbidden. You are allowed to use them only for reading the input, for sure, you can use one instance variable for each problem just for holding the input (for example just the sentence itself for palindrome problem), but using any other instance variables etc. are strictly forbidden.** So, if you do not obey these rules, you will be penalized by deduction of design multiplier from 1 towards 0.

## 7 Submit Format

File hierarchy must be zipped before submitted (Not .rar, only not compressed .zip files because the system just supports not compressed .zip files).

- b<studentid>.zip
  - <src>
    - Main.java
    - \*.java