BBM104 Programming Assignment 4
Özge Kökyay
June 2020

1 Problem Definition

I was expected to implement queue and stack data structures and gain practice on the basics of these structures while doing different operations from command file to stack and queue files.

2 Solution Approach

In CommandExecutor class i readed command, stack and queue files.

I implemented all of the commands with push, pop, top, isEmpty methods of stack class and enqueue and dequeue methods of queue class:

- removeGreater
- calculateDistance
- addOrRemove
- reverse
- distinctElements
- sortElements

3 UML Diagram

<<Java Class>>
Main
(default package)

- Main()
- main(String[]):void

<<Java Class>>
CommandExecutor
(default package)

- command: String[]
- stackArray: String[]
- queueArray: String[]
- sfile: String
- qfile: String

- CommandExecutor(String)
- readFile(String):String[]
- commandExecutor():void
- FrontToLast(Queue,int):void
- pushInQueue(Queue,int,int):void
- sortQueue(Queue):void
- stackCreator():void
- queueCreator():void

<<Java Class>>
Queue
(default package)

- arr: int[]
- front: int
- rear: int
- capacity: int
- count: int

- Queue(int)
- dequeue():void
- enqueue(int):void
- top():int
- size():int
- isEmpty():Boolean
- isFull():Boolean

~queue
0..1

~stack 0..1

<<Java Class>>
Stack<T>
(default package)

- value: T
- capacity: int

- Stack(int)
- Stack(T)
- Stack(Stack<T>,T)
- push(T):void
- pop()
- top()
- isEmpty():boolean
- isFull():boolean
- size():int
- search(Object):int

-previous
0..1