Practical 3

You should tackle this assignment individually. It will count towards your class mark. Demonstrator help with this assignment will be available during your 2 hour week 5 lab time – please make use of this help. The work should be completed and **submitted by all students on MyPlace before 9am on Monday week 6 (23rd October)**. Late submission of up to 4 days is permissible, with deductions as outlined in a separate document on MyPlace. **Marking, on demonstration, will take pace during your week 6 and 7 labs** – more detail of individual demo times will follow on MyPlace. (A new exercise will be issued for you to undertake each week, with every 2nd one assessed.)
This practical exercise will be scaled to be worth 7% of your overall class mark.

Task *(Marked out of 10)*

In this task, for full marks, you are asked to complete 4 different implementations of the QueueADT interface (as described in lecture slides on MyPlace). You are provided on MyPlace with a number of classes in a zip file which you should unzip and bring in to one package for this practical. Code for the QueueADT interface, 2 exception classes, a Node class for use in part (iii) and 4 classes to perform tests are provided - these classes should not be altered. Classes ArrayQueue, ExtendableArrayQueue, NodeQueue and AdapterQueue are provided but are incomplete. They should be completed appropriately in this exercise. Parts i), iii) and iv) are independent of each other and can be tackled in any order. Part ii) relies on part i) being completed and correct. The tests in the associated test classes should not be altered and should be run after the relevant part as a partial check that your classes perform operations correctly.

i) Complete the implementation of the given QueueADT interface in class **ArrayQueue** using an array in a circular fashion. Pseudo-code for this was provided in the later lecture slides from week 4. This version should throw a FullStackException if trying to push too many elements on to the queue since the array has a fixed capacity. You are provided with some code in class ArrayQueue to start you off. Calling the constructor will create an array of size 4 (allowing for a queue of up to size 3). Test your ArrayQueue implementation by running the tests provided in the ArrayQueueTests class.

(3 marks)

ii) Only attempt this part if you have managed to get part i) working. Complete an implementation of the given QueueADT interface in class **ExtendableArrayQueue** using an array in a circular fashion. Use an 'extendable' circular array so that the queue cannot become full. Much of the code will be similar to that you produced in part i). Lecture slides 'Implementations of our Queue ADT' page 8 gives some guidance. However you'll need to think carefully where to place items in the larger array. Test by running tests in the provided ExtendableArrayQueueTests class. Note though that these tests alone do not confirm that your implementation is taking the requested approach.

(2 marks)

iii) Complete an implementation of the given QueueADT interface in class **NodeQueue** using a singly-linked list of nodes, as described in slides 'Implementations of our Queue ADT' slides 1-3. (Much of the code may be similar to that for the singly-linked list implementation of the StackADT given in slides 'Stack implemented using a singly-linked list'.) A **Node** class is provided on MyPlace for your use.

(3 marks)

iv) Complete provided class **AdapterQueue** which should be an implementation of the given QueueADT interface, using the **adapter pattern** with a field of type **java.util.ArrayList**. (See slides 1 and 2 from 'Implementing StackADT through use of a field of an existing collection type' and slide 9 from 'Implementations of our Queue ADT'.)

(2 marks)

The mark breakdown shown above is for guidance only – variations may be made on marking as appropriate. Part ii is perhaps the most challenging, so you may wish to tackle parts iii and iv before attempting part ii.

Submission of your work for marking:
Before the deadline (see above) you should zip your work for this exercise and submit it on MyPlace. Your provisional mark (and before any lateness penalty is applied) and verbal feedback will be given to you in the lab on demo to a marker. Marking will only take place on demonstration of the work you submitted. Demonstration will be in your allocated lab slot at an individual time during weeks 6 and 7. Individual students' week and time will appear on MyPlace later. Some general feedback will be given in a later lecture.

I.Ross Oct 2023