* Don’t forget to set your Eclipse workspace and working set.
* **You must submit the JAR file, exported (with source code), from your Eclipse project.**
* **You must check your JAR file to make sure all the source files (.java files) are present. It can be opened with file compression programs such as 7-zip or Winrar.**
* **Failure to export properly will result in your work not getting marked.** 
  1. **To submit:**
* **Export your project to a JAR file, with source code.**
* **Name your JAR file ID\_Week07\_Q2.jar. For example, 6623110021\_Week07\_Q2.jar**
* **Submit the JAR file on MyCourseville.**

You are given all classes for coding Double ended queue (DeQ).

You are coding a simulation of a queue in a bank (class BankQueue). A bank can have any number of regular queues, plus one special queue.

Graphical user interface, text, application

Description automatically generated

1. (12 marks) Write method

**public** **void** distribute() throws Exception

* This method simulates the opening of a special queue. Some people in the regular queues will go to the special queue so that each regular queue becomes shorter.
* This method assumes that:
  + each regular queue has at least one person in it.
  + The special queue is originally empty.
  + If the longest regular queue has n people in it. Each other regular queue will have n or n-1 people.
* To distribute people into the special queue:
  + Calculate the “needed queue length” using the number of people/number of queues, including the special queue.
    - Calculate the difference between the “needed queue length” and its integer value.
      * If the difference is less than 0.5 then the “needed queue length” becomes that integer value.
      * Otherwise, the “needed queue length” becomes that integer value +1
  + For each regular queue:
    - Maintain the first “needed queue length” number of data in its original sequence.
    - Move the remaining data (from front to back), one by one, to the special queue.
      * If the special queue has length equal to “needed queue length”, then stop moving data to it.
        + Make sure the regular queue, after all these moves, starts with its original first data.
    - If the above process ends, but the special queue still has no data, move the last data of the last regular queue to the special queue.
* Your code must use DeQ methods from interface DeQ. A queue can be implemented using Array or Linked list. Your code must work on both.
* You must not modify any file except BankQueue. Otherwise, you get 0 mark.
* You must not create a new class. Otherwise, you get 0 mark.
* There is no performance requirement for this question.

Counters[0]

1

2

3

4

Counters[1]

5

6

7

8

Counters[2]

9

10

11

12

special

Counters[0]

1

2

3

Counters[1]

5

6

7

Counters[2]

9

10

11

13

special

4

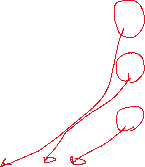
8

12

13

distribute()

“needed queue length = 13/4 = 3.25 -> 3”



Not moved because special queue has the needed length.

Example:



4

“needed queue length = 23/7 = 3.285 -> 3”

Counters[2]

9

10

11

12

Counters[3]

13

14

15

16

Counters[4]

17

18

19

20

Counters[5]

21

22

23

special

Counters[0]

1

2

3

4

Counters[1]

5

6

7

8



distribute()

Counters[2]

9

10

11

Counters[3]

13

14

15

16

Counters[4]

17

18

19

20

Counters[5]

21

22

23

special

Counters[0]

1

2

3

Counters[1]

5

6

7

4

8

12

“needed queue length = 10/6 = 1.667 -> 2”

Counters[2]

5

6

Counters[3]

7

8

Counters[4]

9

10

special

Counters[0]

1

2

Counters[1]

3

4

distribute()

Counters[2]

5

6

Counters[3]

7

8

Counters[4]

9

special

Counters[0]

1

2

Counters[1]

3

4

10



no queue is adjusted so the special queue is empty. Therefore the last data of the last regular queue is taken.

**Score Total 12:**

**3 marks for each test case.**