

Lab Tasks 2

No submission! Use this lab to check your understanding of queues and stacks.

CheckBalanced.java
MyArrayQueue.java
MyArrayListWithReverse.java
MyTwoStacksArray.java

During testing, do not modify the interfaces provided in the given Eclipse project!

1. Exercise 3.16, page 97: Download the assignment project from http://www.comp.nus.edu.sg/~cs1102s/java/labtasks_01.zip.
Complete the class `reverseIterator` in `MyArrayListWithReverse.java`.
The given program `MyArrayList.java` is taken from the textbook.
The iterator function of `MyArrayListWithReverse` should handle the following exceptions:
 - `next()` throws the exception `java.util.NoSuchElementException`,
 - `remove()` throws the exception `IllegalStateException` if `remove()` is called without an immediately preceding `next()`.

(You may ignore exceptions arising from concurrent modification `java.util.ConcurrentModificationException`.)
2. Exercise 3.21 (b), page 98: Implement class `balancing` in `CheckBalanced.java`.
Note the following facts about Java comments:
 - When the compiler reads `/*`, it skips any text until the next character sequence `*/` (ignore all brackets between these two “tokens”).
 - When the compiler reads `//`, it skips any text until the next newline character (ignore all brackets in between).

You may assume that the given Java program has no strings.

3. Exercise 3.24, page 98: Implement the class `twoStacks` in `MyTwoStacksArray.java`.
4. Implement a queue data structure as described in the textbook, using arrays, where the front and back pointers wrap around. When `enqueue(..)` is attempted on a queue whose array is full with queue elements, resize the array as with `ArrayList`.
Implement the class `queues` in `MyArrayQueue.java`.