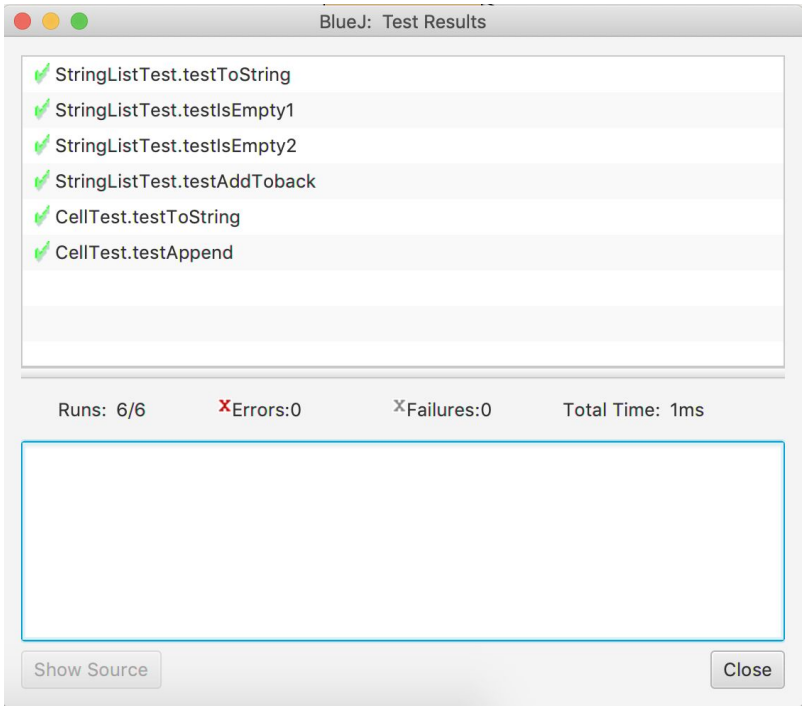
Dylan Maloy
CS150 Lab
Lab #3 write-up
09/15/19

Introduction:
The goal of this lab was to create a linked-list that implemented recursion in its append and toString methods. It was controlled through the experimentController which pointed to the StringListADT and included the methods from the StringList class that extended this ADT. The StringList class included 3 methods (append, toString, and isEmpty). These methods point to the Cell class which acts as a “node”. Via the cell class, the methods (append, toString) recursively call each other and “loop” through the cells to complete the process starting at the head cell and ending at the end of the list when this.next == null.

Unit Tests:

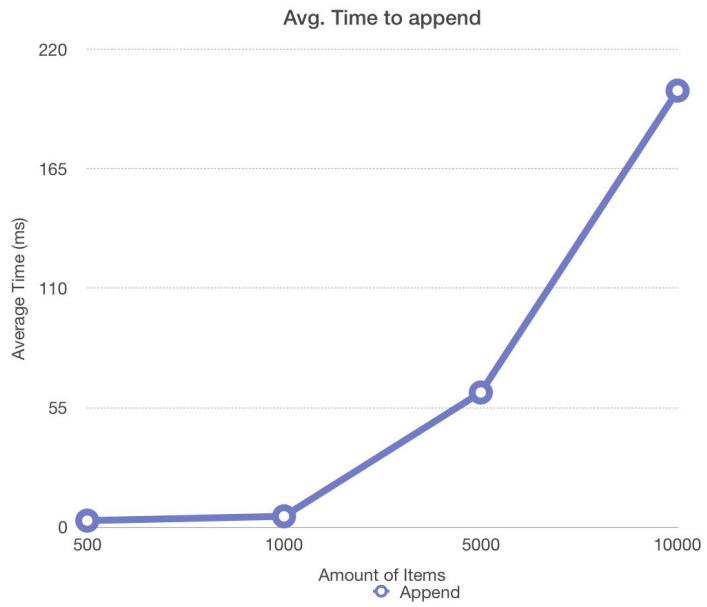


- BlueJ test results window

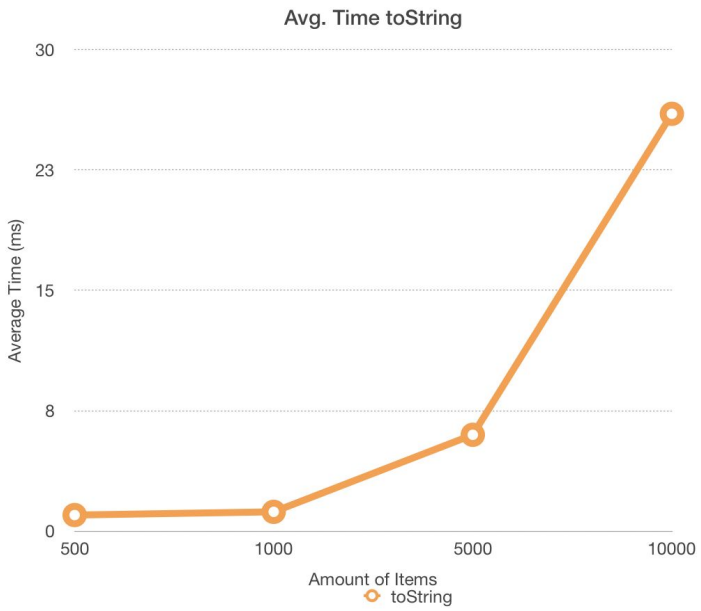
Output Required:

```
 500, 2030203
Execute time (timeAddToback): 4ms
Execute time (toString): 1ms
500, 2030204
Execute time (timeAddToback): 2ms
Execute time (toString): 0ms
1000, 2030203
Execute time (timeAddToback): 6ms
Execute time (toString): 0ms
1000, 2030204
Execute time (timeAddToback): 3ms
Execute time (toString): 0ms
5000, 2030203
Execute time (timeAddToback): 53ms
Execute time (toString): 8ms
5000, 2030204
Execute time (timeAddToback): 53ms
Execute time (toString): 6ms
```

- BlueJ output (amount of items, seed)



- Avg. time to append to list (500 - 10000 items)



- Avg. time toString (500 - 10000 items)

Trouble Report:
This section is not applicable because I did not run into any trouble during the completion of the lab.

References:

This section is not applicable because I didn’t reference any outside sources.