C:\Users\Rich\Documents\NetBeansProjects\Lab06\src\StackTest.java

```
1
2 import java.util.Random;
3 /**
4  *Testing the various runtime of each stack implementation by trying to push
5  * 100,000 elements onto the stack and pop those 100,000 element out the stack
6  * @author Richelin Metellus
7  * @version 02/24/2017
8  */
9 public class StackTest {
10
11
12    public static void main(String[] args) {
13
14        ArrayStack<Integer> goodArrayStack = new ArrayStack<>(100000);
15        LinkedStack<Integer> goodLinkedStack = new LinkedStack<>();
16        ArrayListStack<Integer> ArrListStack = new ArrayListStack<>();
17        ArrayStackBad<Integer> badArrayStack = new ArrayStackBad<>(100000);
18        LinkedStackBad<Integer> badLinkedStack = new LinkedStackBad<>();
19
20        Random rand = new Random();
21        // running time check for ArrayStack
22        long startTime1 = System.currentTimeMillis();
23        for ( int i = 0; i < 100000; ++i)
24        {
25          int randNumber = rand.nextInt(100);
26          goodArrayStack.push(randNumber);
27        }
28        for ( int i = 0; i < 100000; ++i)
29            goodArrayStack.pop();
30        long endTime1 = System.currentTimeMillis();
31        long elapsedTime1 = endTime1 - startTime1;
32
33 //************************************************************
34        // runing time for LinkedStack
35        long startTime2 = System.currentTimeMillis();
36        for ( int i = 0; i < 100000; ++i)
37        {
38          int randNumber = rand.nextInt(100);
39          goodLinkedStack.push(randNumber);
40        }
41        for ( int i = 0; i < 100000; ++i)
42            goodLinkedStack.pop();
43        long endTime2 = System.currentTimeMillis();
44        long elapsedTime2 = endTime2 - startTime2;
45 //************************************************************
46        // runtime for ArrayListStack
47        long startTime3 = System.currentTimeMillis();
48        for ( int i = 0; i < 100000; ++i)
49        {
50          int randNumber = rand.nextInt(100);
51          ArrListStack.push(randNumber);
52        }
53        for ( int i = 0; i < 100000; ++i)
54            ArrListStack.pop();
55        long endTime3 = System.currentTimeMillis();
56        long elapsedTime3 = endTime3 - startTime3;
```

```
57 //**********************************************************************
58       // running time for ArrayStackBad
59       long startTime4 = System.currentTimeMillis();
60       for ( int i = 0; i < 100000; ++i)
61       {
62         int randNumber = rand.nextInt(100);
63         badArrayStack.push(randNumber);
64       }
65       for ( int i = 0; i < 100000; ++i)
66          badArrayStack.pop();
67       long endTime4 = System.currentTimeMillis();
68       long elapsedTime4 = endTime4 - startTime4;
69 //**********************************************************************
70       // running time for LinkedStackBad
71       long startTime5 = System.currentTimeMillis();
72       for ( int i = 0; i < 100000; ++i)
73       {
74         int randNumber = rand.nextInt(100);
75         badLinkedStack.push(randNumber);
76       }
77       for ( int i = 0; i < 100000; ++i)
78          badLinkedStack.pop();
79       long endTime5 = System.currentTimeMillis();
80       long elapsedTime5 = endTime5 - startTime5;
81 //**********************************************************************
82       System.out.printf("push/pop ArrayStack \tfor N\t= 100,000 \ttime \t=  %,10d miliseconds \n", elapsedTime1);
83       System.out.printf("push/pop LinkedStack \tfor N\t= 100,000 \ttime \t=  %,10d miliseconds \n", elapsedTime2);
84       System.out.printf("push/pop ArrayListStack for N\t= 100,000 \ttime \t=  %,10d miliseconds \n", elapsedTime3);
85       System.out.printf("push/pop ArrayStackBad \tfor N\t= 100,000 \ttime \t=  %,10d miliseconds \n", elapsedTime4);
86       System.out.printf("push/pop LinkedStackBad for N\t= 100,000 \ttime \t=  %,10d miliseconds \n", elapsedTime5);
87
88    }
89
90 }
91
```