Fixed size stack of strings

Monday, August 29, 2022 7:26 PM

fixed_size_stack_of_strings.java tobe.txt

```java
import edu.princeton.cs.algs4.StdIn;

public class fixed_size_stack_of_strings {

    public static void main( String[] args)
    {
        var s = new fixed_size_stack(100);
        while (!StdIn.isEmpty())
        {
            String item = StdIn.readString();
            if (!item.equals("-")) s.push(item);
            else if (!s.is_empty()) System.out.print(s.pop() + " ");
        }

        System.out.println(" (" + s.size() + " left on stack) ");
        s.everything_in_a();


    }
}

class fixed_size_stack
{
    private String[] a;
    private int N;

    public fixed_size_stack (int size) { a = new String[size];}

    public boolean is_empty () { return N == 0;}
    public int size () { return N; }
    // N++ means the value of N is returned and then the value of N is incremented by 1
    public void push (String item) { a[N++] = item; }
    // --N means the value of N is decremented by 1, and then it returns the value
    public String pop () { return a[--N]; }

    public void everything_in_a ()
    {
        for ( String s : a) { if (s != null ) System.out.println(s); }
    }
}
```

tobe.txt | fixed_size_stack_of_strings.java

```
to be or not To - Be - - that - - - is
```

```
~/IdeaProjects/Algorithms_4th_edition/src> java fixed_size_stack_of_strings < tobe.txt
To Be not that or be  (2 left on stack)
to
is
or
that
Be
```

StdIn (*push*)	StdOut (*pop*)	N	a[]				
			0	1	2	3	4
		0					
to		1	to				
be		2	to	be			
or		3	to	be	or		
not		4	to	be	or	not	
to		5	to	be	or	not	to
-	to	4	to	be	or	not	to
be		5	to	be	or	not	be
-	be	4	to	be	or	not	be
-	not	3	to	be	or	not	be
that		4	to	be	or	that	be
-	that	3	to	be	or	that	be
-	or	2	to	be	or	that	be
-	be	1	to	be	or	that	be
is		2	to	is	or	not	to

Trace of FixedCapacityStackOfStrings test client