ITI 1121. Introduction to Computer Science II

Laboratory 11

Winter 2016

Objectives

- Further understanding of recursive list processing

Part I

Recursive list processing

For this part of the laboratory, you must write recursive implementations of the methods **inside** the class **BitList**.

- Iterator.java
- BitList.java
- BitListTest.java

1 void complement()

Write the instance method **complement()** for/into the class BitList. It **transforms** this list changing each bit to its complement.

```
a = new BitList( "10001" );
System.out.println( a );
-> 10001

a.complement();
System.out.println( a );
-> 01110
```

2 BitList or(BitList other)

Write an instance method that returns a new BitList that is the **or** of this list and the argument **other**. This list and **other** remain unchanged.

```
a = new BitList( "10001" );
b = new BitList( "00011" );

c = a.or( b );

System.out.println( a );
-> 10001
```

```
System.out.println( b );
-> 00011
System.out.println( c );
-> 10011
```

3 boolean equals(Object other)

Implement the method **boolean equals(Object other)** that returns **true** if **other** designates a **BitList**, of the same size, and such that the bits at the same position in both lists are the same, and **false** otherwise. The implementation must be recursive. Here are examples:

```
a = 11001
b = 11001
a.equals( b ) -> true
a = 11001
b = 11000
a.equals( b ) -> false
a = 01001
b = 11001
a.equals( b ) -> false
a = 1100
b = 11001
a.equals( b ) -> false
a = 11001
b = 110
a.equals( b ) -> false
a = 11001
b = 0
a.equals( b ) -> false
a = 0
b = 11001
a.equals( b ) -> false
a = 0
b = 0
a.equals( b ) -> true
```

Part II

Quiz (1 mark)

If you need to implement a stack (respectively a queue) using queues (respectively stacks), what is the minimum number of required queues (respectively stacks)?

A. 1 queue for a stack, 1 stack for a queue

B. 1 queue for a stack, 2 stacks for a queue

C. 2 queues for a stack, 3 stacks for a queue

D. 3 queues for a stack, 1 stack for a queue

E. 2 queues for a stack, 2 stacks for a queue

F. 3 queues for a stack, 3 stacks for a queue

- Write your answer to the above question directly in the **Submission** text field of the submission Web page:
- https://uottawa.blackboard.com/.

Last Modified: March 30, 2016