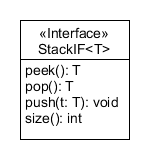
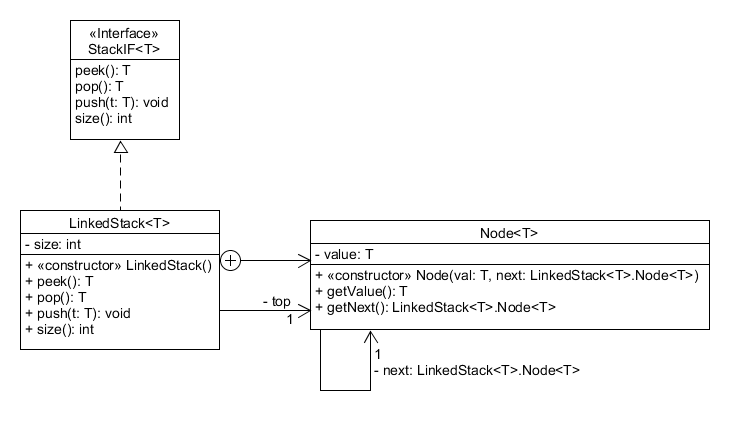
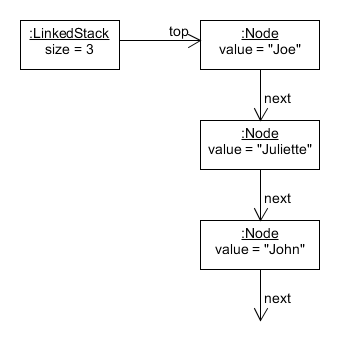
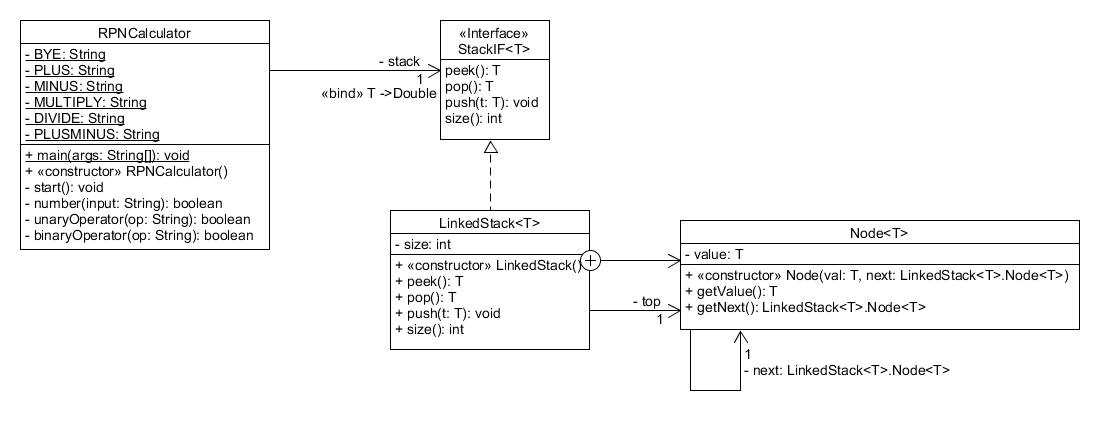
**Implement a LinkedStack and an RPN calculator in Java.**

1. Start by defining the interface: StackIF. (We do not use “Stack” as it is already used in Java.)



1. Implement implement the StackIF as LinkedStack. The provided UML **class diagram** shows an implementation with an internal class, Node<T>. Observe the **object diagram** that shows the object structure of a LinkedStack object of Strings with three elements for a better understanding (**next page**).



1. **Challenge exercise.** Use the LinkedStack implementation to implement an RPN calculator. We implement addition, subtraction, multiplication, division, and sign flip. An RPN calculator accepts numbers and pushes them onto the stack. If it receives an operation, it pops one (for sign flip) or two (for binary operations) numbers from the stack in performs the operation on the value(s) in opposite order, and pushes the result back on top of the stack. Now it’s ready to receive additional operations or numbers. We don’t need to worry about operator precedence or parentheses, as the order of operations performed must observe these rules. To display the top of the stack, use peek().  
   4 + 5 \* 6 = becomes 4 5 6 \* + if performed correctly.  
   Alternatively, one may do 5 6 \* 4 +  
   The provided class diagram shows a possible solution of the task.