CS 240 Homework 3

Alex Vondrak

Fall 2011

For this Homework, we'll be exploring the following code, which is a copy of the Java file you can download from `http://www.csupomona.edu/~ajvondrak/cs/240/11/fall/lecture/Parser.java`.

```java
import java.util.Scanner;
import java.util.Stack;

interface Expr {}

class Atom implements Expr {
   String value;

   public Atom(String value) { this.value = value; }

   public String toString() { return value; }
}

class List<E> implements Expr {
   private Node<E> head;

   public List(E... elements) {
      head = null;

      for(int i = elements.length - 1; i >= 0; i--) {
         head = new Node<E>(elements[i], head);
      }
   }

   public void append(E element) {
      if (head == null) {
         head = new Node<E>(element, null);
      }
      else {
         Node<E> last = head;
         while (last.link != null) last = last.link;
         last.link = new Node<E>(element, null);
      }
   }

   public String toString() {
      String s = "( ";
      Node<E> elements = head;
      while (elements != null) {
         s += elements.data + " ";
         elements = elements.link;
      }
      return s + ")";
   }

   private class Node<E> {
      E data;
      Node<E> link;

      public Node (E data, Node<E> link) {
         this.data = data;
         this.link = link;
      }
   }
}

class UnclosedListException extends Exception { }

public class Parser {
   public static List<Expr> parse(String[] tokens)
                                                    throws Exception {
      Stack<List<Expr>> stack = new Stack<List<Expr>>();

      stack.push(new List<Expr>());

      for (String token : tokens) {
         if (token.equals("(")) {
            stack.push(new List<Expr>());
         }

         else if (token.equals(")")) {
            List<Expr> toAppend = stack.pop();
            stack.peek().append(toAppend);
         }

         else {
            stack.peek().append(new Atom(token));
         }
      }

      if (stack.size() > 1)
         throw new UnclosedListException();

      return stack.pop();
   }

   public static void main(String[] args) throws Exception {
      Scanner in = new Scanner(System.in);
      while(true) {
         System.out.print("lisp> ");
         String line = in.nextLine();
         try {
            System.out.println(parse(line.split("\\s+")));
         }
         catch (UnclosedListException ule) {
            System.out.println("Missing right paren.");
         }
      }
   }
}
```

1. Download the code and test it out by compiling and running it. What does it output for the following inputs?

   (a) `( 1 2 3 )`

   (b) `(1 2 3)`

   (c) `( 8 6 7 ( 5 3 0 9 ) )`

   (d) `( 3 1 4 ( 1 5 ( 9 2 ) 6`

2. Come up with your own examples to test out the code. Give at least three different inputs, and their corresponding outputs. What Java structures do the outputs respectively represent?

3. Explain (in sufficiently detailed English) what the code does: how each line/part works, why they're written in such a way, the algorithms at use, etc.