```java
import components.simplereader.SimpleReader;

/**
 * Program to evaluate XMLTree expressions of {@code int}.
 *
 * @author Shyam Sai Bethina
 *
 */
public final class XMLTreeIntExpressionEvaluator {

    /**
     * Private constructor so this utility class cannot be
instantiated.
     */
    private XMLTreeIntExpressionEvaluator() {
    }

    /**
     * Evaluate the given expression.
     *
     * @param exp
     *            the {@code XMLTree} representing the expression
     * @return the value of the expression
     * @requires <pe>  [exp is a subtree of a
     *           well-formed XML arithmetic expression] and [the
label of the
     *           root of exp is not "expression"]  </pre>
     * @ensures evaluate = [the value of the expression]
     */
    private static int evaluate(XMLTree exp) {
        assert exp != null : "Violation of: exp is not null";

        int result = 0;

        if (!exp.label().equals("number")) {
            //Each node only has two child nodes
            int firstNum = evaluate(exp.child(0));
            int secondNum = evaluate(exp.child(1));

            //does the operation depending on what type of label
the node is
            if (exp.label().equals("plus")) {
                result = firstNum + secondNum;
            } else if (exp.label().equals("minus")) {
```

```
47                result = firstNum - secondNum;
48            } else if (exp.label().equals("times")) {
49                result = firstNum * secondNum;
50            } else {
51                result = firstNum / secondNum;
52            }
53
54        } else {
55            //if the label does equal number, then returns the
  value of the number instead
56            result = Integer.parseInt(exp.attributeValue("value"));
57        }
58
59        return result;
60    }
61
62    /**
63     * Main method.
64     *
65     * @param args
66     *            the command line arguments
67     */
68    public static void main String[] args) {
69        SimpleReader in = new SimpleReader1L();
70        SimpleWriter out = new SimpleWriter1L();
71
72        out.print("Enter the name of an expression XML file: ");
73        String file = in.nextLine();
74        while (!file.equals("")) {
75            XMLTree exp = new XMLTree1(file);
76            out.println(evaluate(exp.child(0)));
77            out.print("Enter the name of an expression XML file:
  ");
78            file = in.nextLine();
79        }
80
81        in.close();
82        out.close();
83    }
84
85}
86
```