```java
import components.simplereader.SimpleReader;
import components.simplereader.SimpleReader1L;
import components.simplewriter.SimpleWriter;
import components.simplewriter.SimpleWriter1L;
import components.xmltree.XMLTree;
import components.xmltree.XMLTree1;

/**
 * Program to evaluate XMLTree expressions of {@code int}.
 *
 * @author Kevin Haller
 *
 */
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 <pre>
     * [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";

        //Result integer
        int result = 0;

        if (exp.label().equals("plus")) {
            //Plus Operator
            result += evaluate(exp.child(0)) + evaluate(exp.child(1));

        } else if (exp.label().equals("minus")) {
            //Subtraction Operator
            result += evaluate(exp.child(0)) - evaluate(exp.child(1));

        } else if (exp.label().equals("times")) {
            //Multiplication Operator
            result += evaluate(exp.child(0)) * evaluate(exp.child(1));

        } else if (exp.label().equals("divide")) {
            //Division Operator
            result += evaluate(exp.child(0)) / evaluate(exp.child(1));

        } else {
            //Number
```

```
58              result = Integer.parseInt(exp.attributeValue("value"));
59          }
60
61          return result;
62
63      }
64
65      /**
66       * Main method.
67       *
68       * @param args
69       *            the command line arguments
70       */
71      public static void main String[] args) {
72          SimpleReader in = new SimpleReader1L();
73          SimpleWriter out = new SimpleWriter1L();
74
75          out.print("Enter the name of an expression XML file: ");
76          String file = in.nextLine();
77          while (!file.equals("")) {
78              XMLTree exp = new XMLTree1(file);
79              out.println(evaluate(exp.child(0)));
80              out.print("Enter the name of an expression XML file: ");
81              file = in.nextLine();
82          }
83
84          in.close();
85          out.close();
86      }
87
88  }
89
```