```java
import components.simplereader.SimpleReader;

/**
 * Program to evaluate XMLTree expressions of {@code int}.
 *
 * @author Vaishnavi Kasabwala
 *
 */

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";

        int solution = 0;

        // digit
        if (exp.label().equals("number")) {
            solution = Integer.parseInt(exp.attributeValue("value"));
        }
        // +
        if (exp.label().equals("plus")) {
            solution = evaluate(exp.child(0)) + evaluate(exp.child(1));
        }
        // -
        if (exp.label().equals("minus")) {
            solution = evaluate(exp.child(0)) - evaluate(exp.child(1));
        }
        // *
        if (exp.label().equals("times")) {
            solution = evaluate(exp.child(0)) * evaluate(exp.child(1));
        }
        // "/"
        if (exp.label().equals("divide")) {
            int dividend = evaluate(exp.child(1));
            if (dividend == 0) {
                Reporter.fatalErrorToConsole("Error: Dividing by zero");
            }
            solution = evaluate(exp.child(0)) / dividend;
```

```
64          }
65
66          return solution;
67      }
68
69      /**
70       * Main method.
71       *
72       * @param args
73       *            the command line arguments
74       */
75      public static void main String[] args) {
76          SimpleReader in = new SimpleReader1L();
77          SimpleWriter out = new SimpleWriter1L();
78
79          out.print("Enter the name of an expression XML file: ");
80          String file = in.nextLine();
81          while (!file.equals("")) {
82              XMLTree exp = new XMLTree1(file);
83              out.println(evaluate(exp.child(0)));
84              out.print("Enter the name of an expression XML file: ");
85              file = in.nextLine();
86          }
87
88          in.close();
89          out.close();
90      }
91
92  }
```