Joshua Newhouse

SE6329

Homework 3

Design

The design follows the provided state diagram to the letter using the State design pattern. The ExpressionEvaluator class acts as the main entry point. It creates a context instance for each expression string processed and for each character of the string calls the context.getNext method.

The EvaluatorContext class maintains the current value, the current operand, the current operator, the current state, and a reference to the previous state.

The states all implement the EvaluatorState interface and provide the actions associated with each transition out of the current state.

There are a couple of slight differences in the way the state diagram looks and how the program implements it.

1. There is no actual end state; when a newline character is encountered in a state that accepts it then the current value is returned and the evaluation terminates.
2. There is no actual error state; since the error state has no transitions then there is no reason to create a state class for it. It is handled by throwing an EvaluatorStateException with an appropriate message.
3. There are two separate space states, one after an operand and one after the operator. They have mostly the same functionality but slightly different transitions. Rather than have two separate classes for each of these states it is implemented with a single class which consumes the space characters and then handles the separate transitions by creating two separate handlers which implement the same interface. This is basically just another smaller example of the entire State diagram pattern.

Finally, the operations performed by the program are implemented as an enumeration which stores them as lambda functions. New operations can simply be added by adding the symbol and lambda function to the enumeration.

Testing

There is a TestDriver program which has a handful of test cases and verifies that the ExpressionEvaluator provides the expected output.

Based on the requirements originally specified by the Professor in class leading 0’s are ignored at the beginning of the expression but are treated as errors if they are on operands anywhere else in the expression. The requirements did not seem too solid and so this may differ in how others have implemented the evaluator.