Introduction

The goal of this assignment is to write a Java program that uses the visitor pattern. Your program will perform various operations (visits) of an expression tree.

Big picture the class AbstractSyntaxElement is an abstraction of the expressions found in multiple programming languages (e.g., \(2+3\times4\)). Class AbstractSyntaxElement will have at least three subclasses: Number (e.g., 42), ParnExpr (e.g., \((42)\)), and ArithExpr (e.g., \(2+3\)), where you must support at least addition, subtraction, and multiplication. While pure object orientation is not the goal of this assignment, you might consider having separate classes for each ArithExpr.

The remaining classes are all visitors, which each perform an operation on the tree. For example, class EvalVisitor evaluates the arithmetic expression represented by the tree, while InorderVisitor, PreOrderVisitor, and PostOrderVisitor produce a string representation of the expression based on the particular visit order.

For the assignment first implement an expression tree with two non-visitor methods: one to evaluate the expression and the other that dumps out the tree for debugging. Finally add five visitors: the four listed above and one of your own devising.

What to hand in

(1) Upload to moodle your single Java source file or a zip if you have multiple source files. In either case include your program’s output on some well chosen tests as a comment at the bottom of the code.