

Expression Tree Code Demo (cont.)

Cpt S 321

Washington State University

Problems you identified last time

- Switch statement in the evaluate method: hardcoded and difficult to evolve
- The algorithm for building the expression tree is not intuitive
- Things are defined in one single file
- The design is very coupled and confusing
- Not sure if it actually works (precedence etc.)
- Hard to read and understand
- All nodes inherit from an empty abstract class
- Don't need nested classes
- Exceptions are all general
- Limited commenting
- No tests
- Duplicated code
- The evaluate method should be defined for class Node

THINGS (NOT PROBLEMS) you identified last time

i.e., Things I disagree with

- Having a class for each node is an overkill
- “cannot handle negative numbers”: Not a problem at this point – this is a feature 😊
- “when a number is divided by 0, we will get ∞ ”: Not a problem – this is a feature 😊: Dividing a **floating-point** value by zero results in positive infinity, negative infinity, or not a number (NaN) according to the rules of IEEE 754 arithmetic.

Summary of solutions

- Throw more descriptive exceptions
- Get rid of the hardcoded operators
- Allow support for new operators without needing to change the logic in every method
- Extract classes into separate files
- Consider operator precedence/associativity
- Parse the expression string and build the expression tree more elegantly
- Get rid of the redundant code

As we go, improve documentation, naming, style

Where do we start?

And in what order do we refactor?

Why is the order important?

STOP!

Before you start improving
the design make sure ALL
your test cases are PASSING!

Pointers for solutions ExpressionTreeCodeDemo

- **Extract classes into separate files**

- I.e., Node, ConstantNode, VariableNode, and OperatorNode should be in their own files.
- How do we implement them?
 - How are they related?
 - What should each of those contain?
 - **Operator precedence/associativity** ([useful](#)):
 - How to define them (instance vs class properties)?
 - Where to define them Base class (i.e., OperatorNode) versus child classes (e.g., PlusOperatorNode)?

STOP!

Don't forget to adapt your test cases and run them – the same number should pass before you continue!