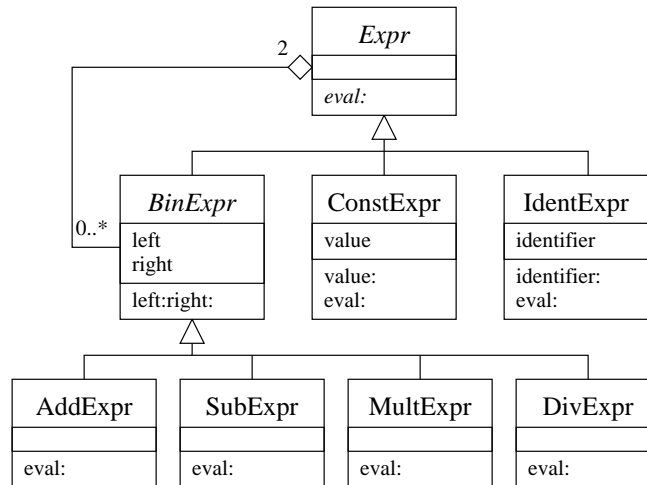


CSci 360, Fall 2004, Assignment 4

This assignment, worth 35 points, is due Friday, October 1, at 5:00pm. To submit, attach your final solution (as described below) to an e-mail sent to cburch@cburch.com.

Your job in this project is to implement an expression tree evaluator, based on the following UML diagram. (Notice, incidentally, how BinExpr is an instance of the Composite design pattern.)



Using these classes, you should for example be able to evaluate the following Squeak code.

```
tree := AddExpr new
    left: (ConstExpr new value: 4)
    right: (MultExpr new
        left: (IdentExpr new identifier: 'x')
        right: (IdentExpr new identifier: 'y')).

context := Dictionary new.
context at: 'x' put: 10.
context at: 'y' put: 2.
tree eval: context
```

The `tree` variable assigned is an expression tree representing the expression $4 + (x \cdot y)$, and `context` is a Dictionary object representing a world in which an `x` variable has the value 10 and a `y` variable has the value 2. (Dictionary is a standard Smalltalk class, similar to HashMap in Java.) When we print `tree eval: context`, the evaluated value should be 24.

You can find instructions about using the Blox GUI on the class Web site. To generate the files that you send to me, for each class you have defined, select it and choose “File out...” from the Class menu. Then attach these files to your e-mail. I do *not* want your image file. I suggest briefly reviewing these files before you send them to me, simply to assure that they contain the appropriate code.

WARNING: Save the image file early and often! The program has a tendency to crash, and when this happens you will lose all of your modifications since the program started! All you need to do to save is to choose “Save image” under the Smalltalk menu. (The image file, `gst.im`, is placed in your current directory, and it will automatically be loaded when you start BLOX later in the same directory.)