Expression Trees

Design of a spreadsheet program

Each cell contains an expression that may reference other cell values

How do we represent these expressions so we can re-evaluate them?
```
Expression Tree Code

public abstract class Exp {
    public double value() ;
} // Exp

public class NumberNode extends Exp {
    private double value;

    public NumberNode( double value ) {
        this.value = value;
    }

    public double value() {
        return value;
    }
} // NumberNode

public class CellNode extends Exp {
    private SpreadsheetCell cell;

    public CellNode( SpreadsheetCell cell ) {
        this.cell = cell;
    }

    public double value() {
        return cell.value();
    }
} // CellNode
```
public static class AddNode extends Exp {

    private Exp node1, node2;

    public AddNode( Exp node1, Exp node2 ) {
        this.node1 = node1;
        this.node2 = node2;
    }

    public double value() {
        return node1.value() + node2.value();
    }
} // AddNode

public static class SubNode extends Exp {

    private Exp node1, node2;

    public SubNode( Exp node1, Exp node2 ) {
        this.node1 = node1;
        this.node2 = node2;
    }

    public double value() {
        return node1.value() - node2.value();
    }
} // SubNode
This scheme can be extended

More operators
if statements

Three-way node with
  condition, consequent, alternative
  if condition then consequent else alternative

Whole programs can be encoded in a parse tree
Game-playing programs

Two person games of perfect information

tic-tac-toe
Nim
Chess
Go
How do we analyze a position?

Game tree

- Each node is a position of the game
- Children are resulting positions after a legal move
- Nodes are valued as the best the player can do assuming children nodes are valued properly
- Base case is won or lost games