Design of Project 4

Jeremiah Moss

Professor Duane Jarc

CMSC 420

For the final project, I have decided to use the MVC architecture, with different models for the formula and for the directed graph that is to be used for the formula dependencies. I have decided to borrow existing classes from previous projects where applicable.

The UI is a simple UI built using the UI editor in NetBeans, which has a label for displaying status messages and a JTable for displaying the spreadsheet itself. Since the spreadsheet is 10x10 and not required to be dynamic, it is stored using 10x10 arrays in the SheetArray class, which has arrays for the original input, the computed output, and the nodes for the directed graph.

The directed graph uses an adjacency list with a list of nodes, and each node having a list of incoming nodes. Each node is also places in the SheetArray class, so that any algorithm can look at the current node from its position in the spreadsheet.

Each cell on the spreadsheet is represented by a cell as defined by the Cell interface, and they are stores in the SheetArray class. If the type of cell is a Formula, it uses an expression tree. Expression trees are created using a slightly modified version of the ExpressionTree class from project 2. After the expression trees are created, they are passed to the Formula class, which processes them and computes the results.

There are still some bugs with the Formula cells involving the expression tree, the directed graph, and computing formulas. While the formulas work, they do not update dynamically and it is unknown whether complex expressions will always give proper results. It is also unknown whether the topological sort is working properly. If I had the time, I would write some unit tests and redesign that portion of the program. The interactions between the formula computations and the directed graph turned out to be complex with the current design. There is a unit testing framework in place, with some working unit tests, but I was unable to write all of the tests by the due date of the assignment.

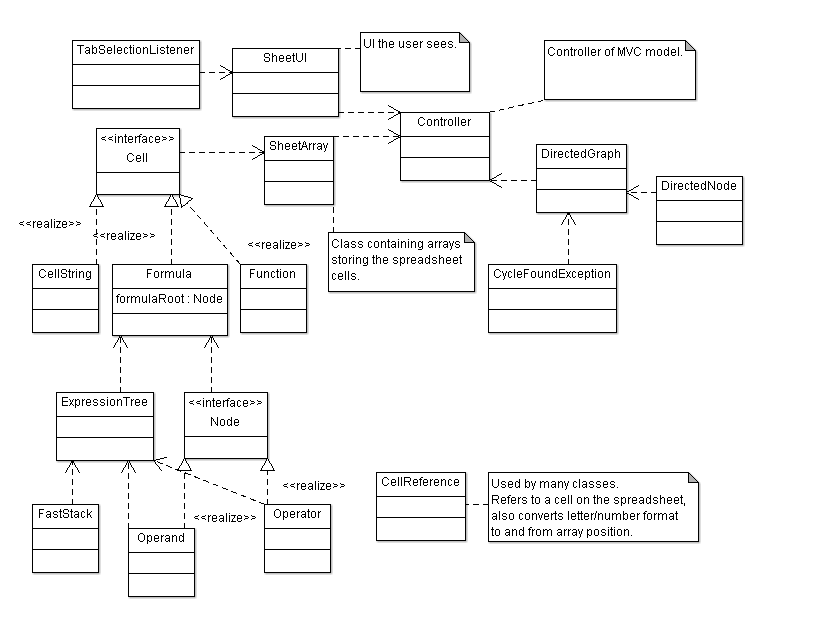


Figure 1: Class diagram of finished project. Created using ArgoUML.