CS 635 Advanced OO Design and Programming
Spring Semester, 2013
Assignment 3

San Diego State University -- This page last updated 2/12/13

SpreadSheet

Due March 7

1. Use the Interpreter pattern to evaluate postfix expressions. The expression (1967 + 21) * sin(3) will be 1967 21 + 3 sin * in postfix. For more information about postfix see the Wikipedia entry entry. You should support at least the operations +, -, *, / , lg (base 2) and cos. You may restrict the input to integer values. However the result may be a floating point number since we are including lg and cos. Postfix is used here to simplify the parsing. If you prefer to use normal arithmetic expressions you may.

2. Implement a spreadsheet with 9 cells with a GUI. The cells are labeled $A, $B, $C, $D, $E, $F, $G, $H, $I. (If you prefer you can use a 2-dimensional layout with the cells labeled AA, AB, AC, BA, BB, etc) A cell can contain either a formula, a number or be empty. A formula can contain numbers, reference to cells and the operations. In the first example below cell $A contains 1, $B contains the number 2, $C contains the formula $A $B +, and $E contains the formula $A $C +. We have an equation and a value view of the cells. The two tables below shows both the equation and the value views of the cells. Provide the user with a button to switch between views. That is you display the spreadsheet once and the user can toggle the view between the two views by clicking on a button. You need to provide a way for the user to enter values and equations. You either allow the user to enter the values directly in a cell or provide them with a separate input field to enter values and/or equations.

Value View

$A	$B	$C	$D	$E	$F	$G	$H	$I
2	3	5		8				

Equation View

$A	$B	$C	$D	$E	$F	$G	$H	$I
2	3	$A $B +		$B $C +				

While in the value view if a user changes the value of a cell (for example cell $B) then all cells dependent on that cell need to be updated automatically. Note that more than one cell may require updating. **One may be tempted to update all the cells whenever a user modifies one cell. However this will not scale to a really spreadsheet, so do not do it. When a user modifies one cell only update the cells that are effected by the change.**

In this example above three cells depend on $A. If the user changes the value in cell $A all the values in the other cells need to be updated. Note that which cells depend on which cell is completely determined by the user. You have to be careful with circular dependancies. The following is just one example of a circular dependancy. This is an error condition which you have to detect.

View	$A	$B	$C	$D	$E	$F	$G	$H	$I
Equation	1	$A $C +	$D 1 +	$B 2 *					
Value	1	Error	Error	Error					

You might find the observer and state patterns useful here.

3. Add an undo mechanism to your spreadsheet. The undo mechanism should be unlimited. However the undo history does not have to span multiple invocations of the program. That is when you start the program your undo history can start empty. Each change the user makes in either a value or an equations is to be undoable.

Grading

Item	Percent of Grade
Working Code	20%
Unit Tests	10%
Proper implementation of Patterns	60%
Quality of Code	10%