Assignment #6: ArrayBasedSpreadSheet

Due Dates:

6A: Sunday, Nov. 28, 11:59 PM ET

6B: Thursday, Dec. 2, 11:59 PM ET

Overview

As we said on assignment #5, in the real-world programs are not written once and then forgotten, rather they are maintained, features are added, bugs are fixed, etc., over time. In that spirit, in this assignment you will write an entirely new version of a spreadsheet program.

In this Assignment You Will Practice:

1) Creating and manipulating arrays
2) Writing an Object Oriented class; no more static!
3) Recursion (albeit possibly indirect recursion): cell formulas can reference other cells which in turn have formulas…just like in Excel

Important Details:

1. The java class you submit must be called ArrayBasedSpreadSheet.
2. Below is the API for the class you must write, a sample program which uses it, and the output from the sample program.
3. Read all the comments above each method signature very carefully and implement the method as described therein!
4. Do not to change anything at all about the signature in your code or it will fail all tests!
5. In this assignment you need not worry about invalid input, but you do have to deal with boundary cases.
6. In your 2D Array, the first dimension must be the columns and the second dimension must be the rows
7. From the user's perspective, the first column is labeled "A", and the first row is numbered 1

Note that in the comments above each method, *@param* means that the following text describes the parameter with the given name, and *@return* means that the following text explains what the method should return

API

```
/**
 * Initializes the spreadsheet to be a given height and width
 * @param rows
 * @param columns
 */
public ArrayBasedSpreadSheet(int rows, int columns)


/**
 * @param showFormulas if true, show the raw cell formulas. If false, show the value generated by
calculating the formulas.
 * @return a String, comma separated values representation of the entire spreadsheet. Each line ends with
"\n"
 */
public String getSpreadSheetAsCSV(boolean showFormulas)


/**
 *
 * Sets a cell to given value, where the value is either a double or a cell formula.
 * Formulas are written as two cell references separated by a basic math operation. There must be a space
 * between each of the 3 elements, i.e. a space between the first cell reference and operation, and then
another
```

```
 * space between the operation and the second cell reference. Example formulas:
  * Addition: A1 + C3
  * Subtraction:  D7 - F6
  * Multiplication: F15 * Z9
  * Division: B6 / C2
  *
  * If the column or row is beyond the current bounds of the 2D array storing the spreadsheet, the size of
the 2D array must be expanded as needed to store the value at column, row.
  *
  * @param column column whose value to set, between 'A' and 'Z'
  * @param row row whose value to set. An integer >= 1
  * @param value must be either a string representation of a double or a cell formula
  */
public void setValue(char column, int row, String value)


/**
 * Returns a complete copy of the spreadsheet data. Since it is a COPY, any edits to the copy will have
no effect on the spreadsheet itself.
  * @return a complete copy of the spreadsheet data
  */
public Object[][] getCopyOfData()


/**
  * Expand the spreadsheet to extend it to the given column.
  * @param column
  */
public void expandColumnRange(char column)


/**
 * Returns a copy of a given column up to and including the given row.
 * If throughRow is larger than the current height of the column, all the values in the rows past the
current column height must be returned as null.
  * @param c the column to get, between 'A' and 'Z'
  * @param throughRow the row to get a copy through.
  * @return
  */
public Object[] getCopyOfColumnThroughRow(char c, int throughRow


/**
  *
  * @param column (first column is A)
  * @param row (first row is 1)
  * @return value stored in that cell. If it holds a double, return the double. If it holds a formula,
return the result of calculating the formula stored in the cell. If the cell is empty, it returns 0.
  */
public double getValue(char column, int row)


/**
  *
  * @param cell the cell whose value should be returned. E.g. "A1" or "F9", etc.
  * @return
  */
public double getValue(String cell)

/**
  * Evaluate the formula held in the given cell
  * @param column 'A' through 'Z'
  * @param row row number, 1 or greater
  * @return the numeric value produced by evaluating the formulae stored in the given cell
  * @throws if the cell does not contain a formula, throw an IllegalStateException
  */
public double evaluateFormula(char column, int row)
```

Sample Program Which Uses the ArrayBasedSpreadSheet

```java
import edu.yu.cs.intro.spreadsheet.ArrayBasedSpreadSheet;

public class Assignment6Demo {
    public static void main(String[] args) {
        ArrayBasedSpreadSheet abss = new ArrayBasedSpreadSheet(6,6);
        //set a few values, print it out
        abss.setValue('A',1,"100");
        abss.setValue('F',5,"55");
        System.out.println(abss.getSpreadSheetAsCSV(true));
        //get copy of data
        Object[][] copy = abss.getCopyOfData();
        copy[0][0] = "99";
        //print out again, showing was just a copy and actual spreadsheet has not been affected
        System.out.println(abss.getSpreadSheetAsCSV(true));
        //fill in some data on column C
        abss.setValue('C',1,"10");
        abss.setValue('C',2,"11");
        abss.setValue('C',3,"12");
        abss.setValue('C',4,"13");
        abss.setValue('C',5,"14");
        abss.setValue('C',6,"15");
        //print out again, showing new values
        System.out.println(abss.getSpreadSheetAsCSV(true));
        abss.expandColumnRange('H');
        //print out again, showing expanded spreadsheet
        System.out.println(abss.getSpreadSheetAsCSV(true));
        //print out some values
        System.out.println("Value of C3 is: " + abss.getValue('C',3));
        System.out.println("Value of F5 is: " + abss.getValue('F',5));
        System.out.println("Value of B3 is: " + abss.getValue('B',3) + "\n");
        //set some cells to formulas
        abss.setValue('D',1,"C1 * F5");
        abss.setValue('E',1,"D1 / C3");
        abss.setValue('A',2,"A1 + C3");
        abss.setValue('B',3,"C6 - C1");
        System.out.println("The value of the formula stored in D1 is: " + abss.evaluateFormula('D',1) +
"\n");
        System.out.println(abss.getSpreadSheetAsCSV(true));
        System.out.println(abss.getSpreadSheetAsCSV(false));

    }
}
```

Output of Sample Program

```
A,B,C,D,E,F
100.0,0.0,0.0,0.0,0.0,0.0
0.0,0.0,0.0,0.0,0.0,0.0
0.0,0.0,0.0,0.0,0.0,0.0
0.0,0.0,0.0,0.0,0.0,0.0
0.0,0.0,0.0,0.0,0.0,55.0
0.0,0.0,0.0,0.0,0.0,0.0

A,B,C,D,E,F
100.0,0.0,0.0,0.0,0.0,0.0
0.0,0.0,0.0,0.0,0.0,0.0
0.0,0.0,0.0,0.0,0.0,0.0
0.0,0.0,0.0,0.0,0.0,0.0
0.0,0.0,0.0,0.0,0.0,55.0
0.0,0.0,0.0,0.0,0.0,0.0

A,B,C,D,E,F
100.0,0.0,10.0,0.0,0.0,0.0
```

```
0.0,0.0,11.0,0.0,0.0,0.0
0.0,0.0,12.0,0.0,0.0,0.0
0.0,0.0,13.0,0.0,0.0,0.0
0.0,0.0,14.0,0.0,0.0,55.0
0.0,0.0,15.0,0.0,0.0,0.0

A,B,C,D,E,F,G,H
100.0,0.0,10.0,0.0,0.0,0.0,0.0,0.0
0.0,0.0,11.0,0.0,0.0,0.0,0.0,0.0
0.0,0.0,12.0,0.0,0.0,0.0,0.0,0.0
0.0,0.0,13.0,0.0,0.0,0.0,0.0,0.0
0.0,0.0,14.0,0.0,0.0,55.0,0.0,0.0
0.0,0.0,15.0,0.0,0.0,0.0,0.0,0.0

Value of C3 is: 12.0
Value of F5 is: 55.0
Value of B3 is: 0.0

The value of the formula stored in D1 is: 550.0

A,B,C,D,E,F,G,H
100.0,0.0,10.0,C1 * F5,D1 / C3,0.0,0.0,0.0
A1 + C3,0.0,11.0,0.0,0.0,0.0,0.0,0.0
0.0,C6 - C1,12.0,0.0,0.0,0.0,0.0,0.0
0.0,0.0,13.0,0.0,0.0,0.0,0.0,0.0
0.0,0.0,14.0,0.0,0.0,55.0,0.0,0.0
0.0,0.0,15.0,0.0,0.0,0.0,0.0,0.0

A,B,C,D,E,F,G,H
100.0,0.0,10.0,550.0,45.833333333333336,0.0,0.0,0.0
112.0,0.0,11.0,0.0,0.0,0.0,0.0,0.0
0.0,5.0,12.0,0.0,0.0,0.0,0.0,0.0
0.0,0.0,13.0,0.0,0.0,0.0,0.0,0.0
0.0,0.0,14.0,0.0,0.0,55.0,0.0,0.0
0.0,0.0,15.0,0.0,0.0,0.0,0.0,0.0
```