# Practical Test 2

## Instructions

1. This test is covers inheritance and polymorphism.
2. You will need the Processing project PracTest2 for this test.
3. You should save your work regularly as you progress through the test.
4. By the end of the test, you should have *one* Processing project to submit. Its name should be **PracTest2**.

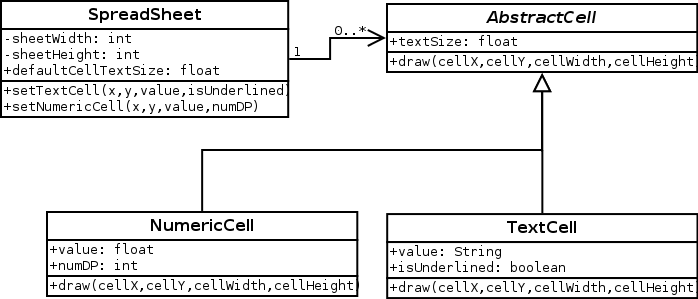
## Task Description

Consider a *spreadsheet* program. A spreadsheet is a 2D array of cells. Each cell can have different properties (e.g. formatting) and different behaviour (e.g. one cell may contain a string, another a formula).

A very simple spreadsheet has only two types of cell:

* numeric, being cells containing numbers that are formatted to a specified number of decimal places; and
* textual, being cells containing a string with optional underlining.

A nearly complete UML diagram for this program is given below:



Note the following:

* In the diagram, *x* and *y* always refer to spreadsheet cell coordinates and are integers, whereas *cellX*, *cellY*, *cellWidth*, and *cellHeight* refer to screen coordinates/sizes and are floats.
* To save space, the diagram does not show the types of each method’s parameters. The types are as follows: *x, y* and *numDP* are always ints, *isUnderlined* is a boolean, the *value* parameter of the setTextCell() method is a String, and everything else are floats.

**Your task is to convert this UML diagram into Processing code as well as you can. Every class, method and property, and class relationship must be correct. Correct programs that deviate from the UML diagram will lose marks.**

## Test Case

One test case is provided for this program in the first tab of the starter program. The code is reproduced here (with extra formatting for readability!):

SpreadSheet sheet = new SpreadSheet(2,6);

sheet.defaultCellTextSize = 50;

sheet.setTextCell( 0,0, "Name", true);

sheet.setTextCell( 1,0, "Score", true);

sheet.setTextCell( 0,1, "Amy", false);

sheet.setTextCell( 0,2, "Jack", false);

sheet.setTextCell( 0,3, "Joe", false);

sheet.setTextCell( 0,4, "Pam", false);

sheet.setTextCell( 0,5, "AVERAGE", true);

sheet.setNumericCell(1,1, 45.67, 1);

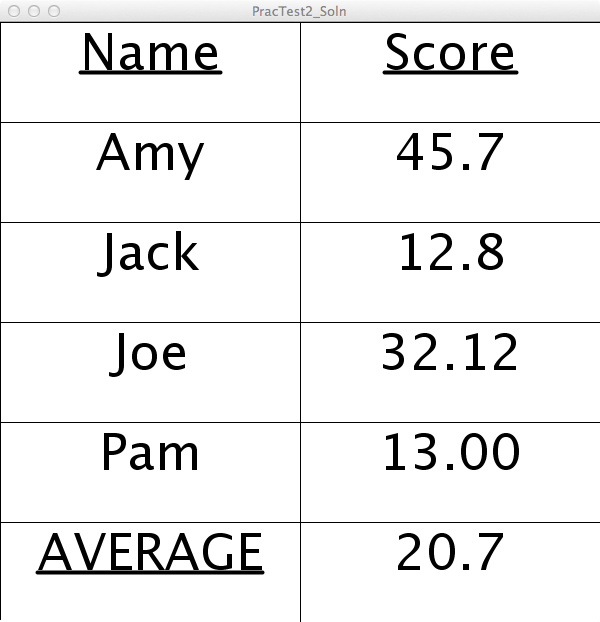
sheet.setNumericCell(1,2, 12.82222, 1);

sheet.setNumericCell(1,3, 32.12345, 2);

sheet.setNumericCell(1,4, 13.0, 2);

sheet.setNumericCell(1,5, 20.723134, 1);

When running this test code, your program should generate this output:



## Notes

None of the class relationships (inheritance or association) have been implemented, so you will need to add these, along with all the properties and methods for each class so that your program passes test case.

You do not need to use any fonts for this test other than the default; the underlining format can be simply achieved using Processing’s line() function.

Converting a number into a string with a specified number of decimal places can be achieved using Processing’s nf() function.

## Submission

Save your file as PracTest2 at the conclusion of the test.

Marking Scheme

|  |  |
| --- | --- |
| **Item** | **Marks** |