

Automatically marking your project (1).

1. Introduction

This document contains information about a set of Java files which, once properly inserted in your spreadsheet project, shall give you a part of the marks corresponding to the “FeaturesTested” sheet of the feedback spreadsheet that you receive every time you make a delivery of the project. More specifically, this set of java files shall automatically mark the features listed in the table below.

NOTE: these weights are valid only for **GROUPS of TWO PERSONS**.

Feature	Weight
Assignment of TEXT CONTENT to one cell	0,15
Assignment of NUMERIC CONTENT to one cell	0,15
FORMULA with Numbers	0,3
FORMULA with Numbers with parenthesis (one level)	0,4
FORMULA with:Numbers with parenthesis (two levels)	0,5
FORMULA with: Numbers, . Cells references	0,5
FORMULA with: Numbers, Cells references with parenthesis (one level)	0,55
FORMULA with: Numbers, Cells references with parenthesis (two levels)	0,65
FORMULA with: Numbers, Cells references, Functions with the following arguments: Numbers	0,75
FORMULA with: Numbers, Cells references, Functions with the following arguments: Numbers, Cell references	0,85
FORMULA with: Numbers, Cells references, Functions with the following arguments: Numbers, Cell references, Ranges	0,95

FORMULA with: Numbers, Cells references, Functions with the following arguments: Numbers, Cell references, Ranges, and other functions	1
IDENTIFIES CIRCULAR DEPENDENCIES	0,75
PROPERLY UPDATES THE VALUES OF THE ALL CELLS THAT ARE DIRECTLY OR INDIRECTLY IMPACTED BY CHANGING THE CONTENT OF ONE CELL.	1,25
PROPERLY UPDATES THE VALUES OF THE ALL CELLS THAT ARE DIRECTLY OR INDIRECTLY IMPACTED BY CHANGING THE CONTENT OF ONE CELL.	1,25
PROPERLY LOADS A SPREADSHEET FROM A FILE.	0,5
PROPERLY SAVES A SPREADSHEET IN A FILE.	0,5

IMPORTANT NOTE: The automatic marker does not mark the following feature, which shall be manually marked during the final review of your projects: "PROPERLY SHOWS THE CONTENT OF A SPREADSHEET", which has a weight of 0,25.

You must take into account that I will use this file to mark the part of the project related with the features provided.

2. Material provided

The zip file uploaded contains:

1. This explanatory document
2. The zipped NetBeans project spreadsheetMarkerForStudents.zip, which contains a number of packages which you will have to integrate into your projects
3. The **BaseCorrectorDACJunit4-2022_2023on.jar** JAR file, which contains the superclass of all the markers.
4. The apidocs.zip zipped Javadoc of the classes included within the src packages of the zipped NetBeans project spreadsheetMarkerForStudents.zip.

3. Joining your code with the markers

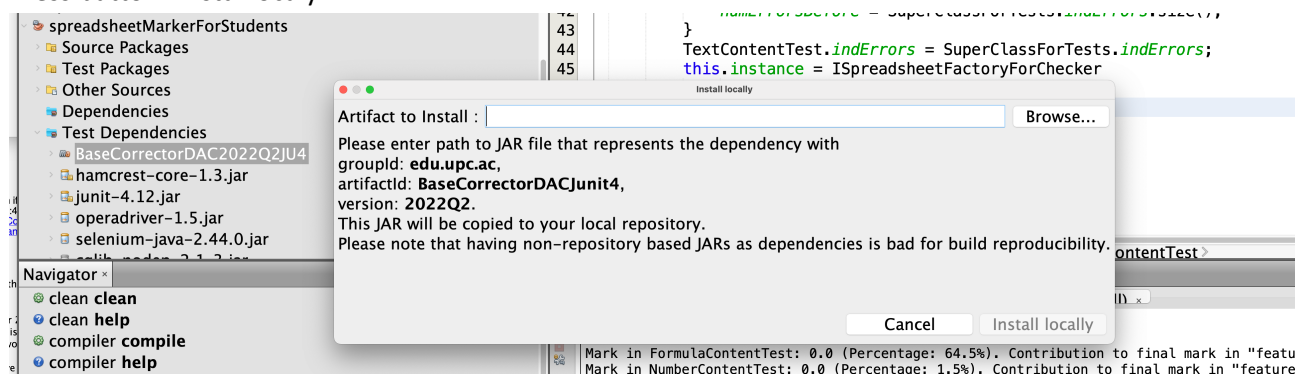
You may proceed as you prefer. You either move the code of your project into the spreadsheetMarkerForStudents or you move the code of this last project into your NetBeans project.

The advantage of moving your code into the spreadsheetMarkerForStudents is that the project has a pom file which contains all the dependencies required for successfully running the marker. Although, in fact, if

you decide to integrate the marker code into your project, you may copy these dependencies in the pom file of your own project.

Regardless the way you choose for integrating the marker into your project you must install the **BaseCorrectorDACJunit4-2022_2023on.jar** JAR file into your local maven repository so that maven is able to integrate this jar into your project. For doing that, you can use NetBeans IDE.

1. Go to "Test Dependencies" in the Projects tab.
2. Click right button on **BaseCorrectorDACJunit4-2022_2023on**.
3. Select "Manually install artifact" as indicated in the figure below.
4. Navigate to the folder where you have downloaded **BaseCorrectorDACJunit4-2022_2023on.jar** and select this file.
5. Press button "Instal locally".



IMPORTANT NOTE: The figure above shows an old dependency. Whenever you see in the figure BaseCorrectorDAC2022Q2JU4, you should see **BaseCorrectorDACJunit4-2022_2023on**.

When you complete this installation, the warning sign in BaseCorrectorDAC2022Q2JU4 shall disappear and will appear as in the figure above.

4. Before start running the markers.

Before running the markers you must read the Javadoc of the spreadsheetMarkerForStudents project delivered.

The source packages of the spreadsheetMarkerForStudents project, contain:

1. A number of exception classes which you do not have to modify, but you have to understand their purposes reading the Javadoc
2. The ISpreadsheetControllerForChecker.java Java interface. This interface is used by the markers for changing the contents of selected cells in your spreadsheet (invoking the method *public void setCellContent(String cellCoord, String strContent)*), getting the value of the content of a certain cell as a double (invoking the method *public double getCellContentAsDouble(String coord) throws BadCoordinateException, NoNumberException*), getting the value of the content of a certain cell as a String double (invoking the method *public double getCellContentAsDouble(String coord) throws BadCoordinateException*), getting the textual representation of a formula *public String getCellFormulaExpression(String coord) throws BadCoordinateException*; and others.
3. The ISpreadsheetFactoryForChecker Java interface with a static method (since Java 1.8, Java interfaces may include static methods which are invoked as any other static method in a class: <Name of the Java interface>.<invocation of the method>. In this case it would be: ISpreadsheetFactoryForChecker.createSpreadsheetController()).
4. An excel spreadsheet (HojaDeEvaluacion.xlsx) with several sheets, each one supporting the different groups of features for processing formulas.

For preparing the project to mark your code, you have to:

1. Either create a new class or modify some of your already existing class (likely your SpreadsheetController) so that it implements the ISpreadsheetControllerForChecker java interface. Once this is done, the markers shall use objects of this (new or modified) class for marking your code.
2. Implement the method of ISpreadsheetFactoryForChecker:: createSpreadsheetController() as specified in the javadoc.

In the method ISpreadsheetFactoryForChecker:: createSpreadsheetController() you must insert the code that creates all the objects that your code needs for starting the normal execution of your program. The code must return an object that implements the ISpreadsheetControllerForChecker java interface.

5. Running the markers.

The code includes 7 markers, namely:

- TextContentTest: in charge of marking the feature "Assignment of TEXT CONTENT to one cell".
- NumberContentTest: in charge of marking the feature "Assignment of NUMERIC CONTENT to one cell".
- FormulaContentTest: in charge of marking the features from "FORMULA with: Numbers" to "FORMULA with: Numbers, Cells references, Functions with the following arguments: Numbers, Cell references, Ranges, and other functions". Therefore this marker does not test and mark the update of the values of the contents of dependent cells, nor the inclusion of circular dependencies.
- DependentCellsTest: in charge of marking the feature "PROPERLY UPDATES THE VALUES OF THE ALL CELLS THAT ARE DIRECTLY OR INDIRECTLY IMPACTED BY CHANGING THE CONTENT OF ONE CELL".
- CircularDependenciesTest: in charge of marking the feature "IDENTIFIES CIRCULAR DEPENDENCIES".
- SaveTest: in charge of marking the feature "PROPERLY SAVES A SPREADSHEET IN A FILE".
- LoadTest: in charge of marking the feature "PROPERLY LOADS A SPREADSHEET FROM A FILE".

NOTE 1: it would be a good exercise to take a look to the contents of the markers. You would then be aware of the kind of tests that they perform and how they compute your marks.

In the package edu.upc.arqsoft.marker you will find the following classes:

- TestSuite: in charge of defining the suite of classes that JUnit will automatically run. In our case it contains the 7 already mentioned marker classes.
- TestAll: the class that contains the main() method. It launches the execution of the marker classes, and presents the final results.

In order to run the markers, you only have to run the class edu.upc.arqsoft.marker.TestAll.

Below follows an example of an execution of the markers that signal that some things are OK and others are not OK.

Note that if some check results in an error, a message providing details of the error shall appear immediately after the message that notifies the check.

Note also that at the end of the report of each marker, if there are errors, a summary of the errors (message starting "*Resumen de errores*") shall be given.

Finally, note that the final part of the report is a list of the marks obtained by the markers. Note that this report mentions does not contain any specific report for LoadTest and SaveTest: this is because the code used did not actually loaded or saved any spreadsheet. When you integrate this with your code, the reports shall appear.

-----< edu.upc.etsetb.arqsoft:spreadsheetMarkerForProfessor >-----

Building spreadsheetMarkerForProfessor 1.0-SNAPSHOT

-----[jar]-----

--- exec-maven-plugin:3.0.0:exec (default-cli) @ spreadsheetMarkerForProfessor ---

SpreadsheetControllerForChecker::testSetCellContent() with text content. Values: 10.0

Puntos obtenidos: 10.0. Puntos acumulados: 10.0

SpreadsheetControllerForChecker::testSetCellContent() with numerical content. Value: 10.0

Puntos obtenidos: 10.0. Puntos acumulados: 10.0

Setting a formula with only numbers as operands in a cell. Total value (over 10): 0.465116279069767

Case 1: a sum of two numbers. Value: 0.11627906976744175

Case 2: a subtraction of two numbers. Value: 0.11627906976744175

Case 3: a multiplication of two numbers. Value: 0.11627906976744175

Case 4: a division of two numbers. Value: 0.11627906976744175

Puntos obtenidos: 0.465. Puntos acumulados: 0.465

Setting a formula with only numbers and one level of parenthesis in a cell

Puntos obtenidos: 0.62. Puntos acumulados: 1.085

Setting a formula with only numbers and two levels of parenthesis in a cell

Puntos obtenidos: 0.775. Puntos acumulados: 1.86

Setting a formula with numbers and references to cells as operands

Puntos obtenidos: 0.775. Puntos acumulados: 2.636

Setting a formula with numbers, references to cells, and one level of parenthesis

Puntos obtenidos: 0.853. Puntos acumulados: 3.488

Setting a formula with numbers, references to cells, and one level of parenthesis

Puntos obtenidos: 1.008. Puntos acumulados: 4.496

Setting a formula with numbers, references to cells, and functions that have only numbers as args

Puntos obtenidos: 1.163. Puntos acumulados: 5.659

Setting a formula with numbers, references to cells, and functions that have numbers and references to cells as args

Puntos obtenidos: 1.318. Puntos acumulados: 6.977

Setting a formula with numbers, references to cells, and functions that have numbers, references to cells, and ranges as args

Puntos obtenidos: 1.473. Puntos acumulados: 8.45

Setting a formula with numbers, references to cells, and functions that have numbers, references to cells, ranges, and other functions as args

Puntos obtenidos: 1.55. Puntos acumulados: 10.0

Checking the proper update of cells that contain formulas that contain an operand that is a reference to the cell whose content is modified. Total value (over 10): 2.5

Case 1: modifying one cell that is directly referenced as an operand in the formula: 1.25

The cell should contain the number: 2 -result of formula =A1+2-A2, when A1=2 and A2=2- with a margin of 0.00001. Instead, it contains the value 1.0; expected:<2.0> but was:<1.0>

Case 2: modifying a second cell that is directly referenced as an operand in the formula: 1.25

The cell should contain the number: 0 -result of formula =A1+2-A2, when A1=2 and A2=4- with a margin of 0.00001. Instead, it contains the value 1.0; expected:<0.0> but was:<1.0>

Puntos obtenidos: 0.0. Puntos acumulados: 0.0

Checking the proper update of cells that contain formulas that contain a function with an argument that is a reference to the cell whose content is modified. Total value (over 10): 3.5

Case 1: modifying one cell whose reference is one of the arguments of a function in another cell: 1.75

The cell should contain the number: 14 -result of formula =1+SUMA(A3;A4;A5), when A3=4, A4=4, and A5=5- with a margin of 0.00001. Instead, it contains the value 13.0; expected:<14.0> but was:<13.0>

Case 2: modifying a second cell whose reference is one of the arguments of a function in another cell: 1.75

The cell should contain the number: 15 -result of formula =1+SUMA(A3;A4;A5), when A3=4, A4=5, and A5=5- with a margin of 0.00001. Instead, it contains the value 13.0; expected:<15.0> but was:<13.0>

Puntos obtenidos: 0.0. Puntos acumulados: 0.0

Checking the proper update of cells that contain formulas that contain a function with an argument that is a range that contains the cell whose content is modified. Total value (over 10): 4.0

Case 1: modifying one cell that appears in the range that is one of the arguments of a function in another cell: 2.0

The cell should contain the number: 43 -result of formula =2+SUMA(A6:A10), when A6=7, A7=7, A8=8, A9=9, and A10=10- with a margin of 0.00001. Instead, it contains the value 42.0; expected:<43.0> but was:<42.0>

Case 2: modifying a second cell that appears in the range that is one of the arguments of a function in another cell: 2.0

The cell should contain the number: 44 -result of formula =2+SUMA(A6:A10), when A6=7, A7=8, A8=8, A9=9, and A10=10- with a margin of 0.00001. Instead, it contains the value 42.0; expected:<44.0> but was:<42.0>

Puntos obtenidos: 0.0. Puntos acumulados: 0.0

Resumen de errores en DependentCellsTest

The cell should contain the number: 2 -result of formula =A1+2-A2, when A1=2 and A2=2- with a margin of 0.00001. Instead, it contains the value 1.0; expected:<2.0> but was:<1.0>

The cell should contain the number: 0 -result of formula =A1+2-A2, when A1=2 and A2=4- with a margin of 0.00001. Instead, it contains the value 1.0; expected:<0.0> but was:<1.0>

The cell should contain the number: 14 -result of formula =1+SUMA(A3;A4;A5), when A3=4, A4=4, and A5=5- with a margin of 0.00001. Instead, it contains the value 13.0; expected:<14.0> but was:<13.0>

The cell should contain the number: 15 -result of formula =1+SUMA(A3;A4;A5), when A3=4, A4=5, and A5=5- with a margin of 0.00001. Instead, it contains the value 13.0; expected:<15.0> but was:<13.0>

The cell should contain the number: 43 -result of formula =2+SUMA(A6:A10), when A6=7, A7=7, A8=8, A9=9, and A10=10- with a margin of 0.00001. Instead, it contains the value 42.0; expected:<43.0> but was:<42.0>

The cell should contain the number: 44 -result of formula =2+SUMA(A6:A10), when A6=7, A7=8, A8=8, A9=9, and A10=10- with a margin of 0.00001. Instead, it contains the value 42.0; expected:<44.0> but was:<42.0>

Checking that the program detects a direct circular dependency: 4.0

Cell A1 contains the formula =A2+A3+A4+A5, and now a try has been done to set cell A2 to =A1+A7+A8. This introduces a direct circular dependency that your program should have detected and the corresponding type of exception should have been thrown. Instead, a java.lang.StackOverflowError has been thrown. You should review your code;

Puntos obtenidos: 0.0. Puntos acumulados: 0.0

Checking that the program properly detects an indirect circular dependency: 6.0

A change is introduced in a cell that does not introduce a circular dependency

*Case 2: modifying a cell in such a way that it introduces an indirect circular dependency:
2.0999999999999996*

*Case 3: modifying another cell in such a way that it introduces an indirect circular dependency:
2.0999999999999996*

Cell A1 contains the formula =A2+A3+A4+A5, cell A2 contains the formula =A6+A7+A8, and now a try has been done to set cell A6 to =A1+5. This introduces a circular dependency, BUT your program has not thrown an exception notifying this circular dependency. Instead it has thrown a `java.lang.StackOverflowError`. Review your code;

Puntos obtenidos: 1.8. Puntos acumulados: 1.8

Resumen de errores en CircularDependenciesTest

Cell A1 contains the formula =A2+A3+A4+A5, and now a try has been done to set cell A2 to =A1+A7+A8. This introduces a direct circular dependency that your program should have detected and the corresponding type of exception should have been thrown. Instead, a `java.lang.StackOverflowError` has been thrown. You should review your code;

Cell A1 contains the formula =A2+A3+A4+A5, cell A2 contains the formula =A6+A7+A8, and now a try has been done to set cell A6 to =A1+5. This introduces a circular dependency, BUT your program has not thrown an exception notifying this circular dependency. Instead it has thrown a `java.lang.StackOverflowError`. Review your code;

Testing save spreadsheet. Total value (over 10): 10.0

Testing load spreadsheet. Total value (over 10): 10.0

Resumen de notas obtenidas en corrección automática:

*Nota en clase FormulaContentTest: 10.0 (Porcentaje en nota final: 64.5%). Contribución a nota final:
6.4499999999999997*

Nota en clase LoadTest: 0.0 (Porcentaje en nota final: 5.0%). Contribución a nota final: 0.0

Nota en clase NumberContentTest: 10.0 (Porcentaje en nota final: 1.5%). Contribución a nota final: 0.15

Nota en clase DependentCellsTest: 0.0 (Porcentaje en nota final: 12.5%). Contribución a nota final: 0.0

Nota en clase CircularDependenciesTest: 1.8 (Porcentaje en nota final: 7.5%). Contribución a nota final: 0.13499999999999998

Nota en clase TextContentTest: 10.0 (Porcentaje en nota final: 1.5%). Contribución a nota final: 0.15

Nota en clase SaveTest: 0.0 (Porcentaje en nota final: 5.0%). Contribución a nota final: 0.0

Nota final de corrección automática: 6.885

BUILD SUCCESS

Total time: 0.711 s

Finished at: 2022-12-28T11:55:28+01:00
