Assessed Exercise 2

Tim Storer
February 24, 2010

- Due at 10:00 am on the day after your lab during the week of 15th March.
- Reminder: credit for OOSE2 will be refused unless you submit at least one of the two assessed exercises and attend at least three OOSE2 lab sessions.

This handout describes the second assessed exercise for OOSE2. Read it over carefully. If you do not fully understand what you need to do, please discuss with your tutor during your lab the week of 1st March.

You are responsible for modifying an existing, working application to meet a set of additional requirements. You have been provided with the requirements specification, the design specification, and a set of JUnit test drivers that are used to validate the classes that make up the application. You have been given java files for all of the classes that constitute the application and the test applications. Additionally, you have been provided with the html files generated by javadoc.

1 Task

For this exercise, you are required to implement a Scientific calculator that provides the following additional capabilities over and above that of the Basic calculator (where S refers to the stack, S_y refers to the top element of the stack and S_x refers to the second element on the stack):

Operation	LUI Symbol	Category	Type	Effect
e	e	Constant		S_y:= e
Natural log	ln	Arithmetic	unary	$\log_{\mathrm{e}} S_y$
Exponential	exp	Arithmetic	unary	e^{S_y}
Power	pow	Arithmetic	binary	$S_y^{S_x}$
Radian	rad	Arithmetic	unary	$\pi * S_y/180$
Sine	sin	Arithmetic	unary	$\sin(S_y)$
Cosine	cos	Arithmetic	unary	$\cos(S_y)$
Tangent	tan	Arithmetic	unary	$\tan(S_y)$

NOTE THAT sine, cosine and tangent functions operate on degree radians. So, for example, the sequence

```
90 rad sin
```

Should output

```
1
```

You need to provide both line-oriented and graphical user interfaces to your extended calculator. The design document describes how this is supported by the design, and gives you guidance in doing so. An `ExtendedCalculator` class that has been extended to support temperature conversion is provided for you to study/emulate.

For the line-oriented user interface (LUI), you are to use the names in the 'LUI Symbol' column in the table above for the operations. For the graphical user interface, you are to arrange the buttons in the following order on the calculator keypad and use the labels shown:

rad	sin	cos	tan
e	log	exp	pow

2 Approach

The most important first step is to thoroughly read the requirements and design documents to make sure you understand them. Then study the java classes; it is critical that you understand what each of the methods are doing AND how they are performing their tasks.

Next you should study the "Extended" calculator provided in the assignment to be sure you understand how to extend the "Basic" classes when providing a new calculator in the framework.

Then you must create and implement the classes:

- `ScientificCalculator`,
- `LuiScientific`
- `GuiScientific`
- `ScientificCalculatorTest`

in the correct packages as described in the design document.

You also need to create the files:

- GlasgowCalculatorFramework/docs/ScientificCalculatorRequirements.doc
- GlasgowCalculatorFramework/docs/ScientificCalculatorImplementation.doc

to document the additional requirements that your calculator meets and how you implemented these requirements while conforming to the design document. Finally, you need to generate Javadoc files for your new classes.

3 Set Up

When you set up Exercise2 using AMS you will obtain an Exercise2 folder, and within it a folder entitled GlasgowCalculatorFramework. For this assignment, you will have to switch your Eclipse workspace to the Exercise2 folder.

Create a new Java project, named GlasgowCalculatorFramework, creating it from the existing source in the GlasgowCalculatorFramework folder in Exercise2. At this point, all of the files in the folder tree descending from GlasgowCalculatorFramework will become part of your project in Eclipse. These files include the source code and javadoc-generated documentation for the classes and applications. The requirements documents and design document, as Word files, are available in the Exercise2 folder.

You should NOT modify or rename any of the files you receive, with the exception of `CalculatorFactory`.

4 What you will be submitting

When you submit your solution to Exercise 2, we will copy the following files into the AMS system:

- Exercise2/GlasgowCalculatorFramework/docs/ScientificCalculatorRequirements.doc
- Exercise2/GlasgowCalculatorFramework/docs/ScientificCalculatorImplementation.doc
- Exercise2/GlasgowCalculatorFramework/src/ex2/oose2/model/ScientificCalculator.java
- Exercise2/GlasgowCalculatorFramework/src/ex2/oose2/view/lui/LuiScientific.java
- Exercise2/GlasgowCalculatorFramework/src/ex2/oose2/view/gui/GuiScientific.java
- Exercise2/GlasgowCalculatorFramework/src/ex2/oose2/view/CalculatorFactory.java
- Exercise2/GlasgowCalculatorFramework/src/ex2/oose2/model/tests/ScientificCalculatorTest.java

Any other files that you create anywhere in the Exercise2 tree will not be copied.

At the time of submission, the line-oriented version of the calculator will be executed against a set of legal and faulty RPN expressions to make sure that the `ScientificCalculator` and `LuiScientific` classes work correctly. Your tutor will also test drive the graphical version of the calculator to see how it works.

5 Assessment

Your submission will be marked on a basis of 10 marks for the assignment.

Section	Marks
Documentation (requirements,implementation,source)	2
ScientificCalculatorTest tests	2
ScientificCalculator	1
LuiScientific	2
GuiScientific	2
Unseen submission test	1