

Assignment 2: Console Input/Output

ETH Zurich

March 4, 2013

Hand-out: March 4, 2013
Due: March 11, 2013 (Java)
March 18, 2013 (C#)

1 Task 1: Code

Write a console-based interface for the calculator component you have constructed in the first assignment. The interface should provide at least the following commands:

- + Adds top two elements of the stack.
- Subtracts top-most entry from second top-most entry of the stack.
- * Multiplies top two elements of the stack.
- / Divides second top-most entry by top-most entry of the stack.

clear Wipes out stack.

print Prints stack.

evaluate Evaluates expressions on stack.

help Prints explanatory text on how to use the calculator.

quit Terminates program execution.

The following shows an example session of such a calculator:

```
#10
#15
#+
#print
10.0
15.0
+
#evaluate
#print
25.0
#4
#/
#2
#+
#print
25.0
4.0
```

```
/
2.0
+
#evaluate
#print
8.25
#quit
Console calculator quitting...
```

Parametrize your console UI code with input and output streams, so that you can run the same code against the JUnit/NUnit test provided with the assignment or with the console input and output (for example using a `BufferedReader` and `PrintWriter` in Java or a `TextReader` and `TextWriter` in C#). See the test-classes for the console UI tests to see how a possible interface for this looks like.

You might want to use the *Model-View-Controller* pattern to implement the console UI.

2 Task 2: National Language Support (NLS)

Use class `ResourceBundle` (Java) or `ResourceManager` (C#) to provide the calculator with more languages. Add commands “deutsch” to switch the interface to German and “english” to switch back to English. For example, the following shows a possible session of interaction.

```
#deutsch
Aktuelle Sprache: deutsch
Befehle: loeschen, drucken, beenden, hilfe, auswerten
#4
#5
#+
#auswerten
#drucken
9.0
#english
Current language: english
Available commands: clear, print, quit, help, evaluate
#quit
Console calculator quitting...
```

3 Task 3: Design Description

Describe if and what you had to change in the design of your component to complete this assignment. Did you have to put the user interface I/O code into the component itself or were you able to avoid it?

4 Submission

Zip your code into a single file and send it to your assistant via email. Include the whole project structure in the zip file. Make your email subject informative by including the course id, the assignment number, the implementation language, and your name. For example, for Tom Cruise to hand in his Java implementation for assignment 2, the email subject could be

[JCD]-2-Java-Tom Cruise

where “JCD” stands for “Java and C# in Depth”.

5 Contacts

Christian Estler	<code>christian.estler@inf.ethz.ch</code>
Alexey Kolesnichenko	<code>alexey.kolesnichenko@inf.ethz.ch</code>
Max (Yu) Pei	<code>yu.pei@inf.ethz.ch</code>
Julian Tschannen	<code>julian.tschannen@inf.ethz.ch</code>

Note:

- Language-specific strings necessary to the program are available in the language property files coming with this discription.
- The test code for this assignment depends on the design of the application. You may change the relevant part of the test case to suit your implementation, but not the input/output behavior specified in the test.