

Assignment 1: Calculator Component

ETH Zurich

28 March 2007

Hand-out: 28 March 2007

Due: 3 April 2007

1 Description

The aim of this exercise is to write a Reverse Polish notation (RPN) based calculator. In the RPN, also called postfix notation, the operands precede the operator. The notation has no need for parentheses. For example the expression $3 * (4 + 7)$ would be written as

```
3
4
7
+
*
```

You can learn more about RPN calculators on its wikipedia page at http://en.wikipedia.org/wiki/Reverse_Polish_notation.

Make sure to properly use the object-oriented facilities such as encapsulation, inheritance, design patterns, ... to implement the calculator and try to implement it as cleanly as possible. Goal of this exercise is not to try create a very complex application. The simplicity of the domain (you do know how a basic calculator works, do you) allows you to focus on the technical aspects of software development.

2 Task 1: Code

For this assignment please implement the calculator as a re-usable component. It can consist of one or several classes. It should not have input or output facilities, instead it should be usable by other classes who provide their own I/O mechanism. You do not need to provide these facilities. Try to design your calculator's interface so that it is likely to be usable by potential front-ends with no or little change.

The calculator must provide the following operations:

- + 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.

stack Access to elements on stack.

evaluate Evaluates expressions on stack.

Note that the calculator should be implemented with lazy evaluation. This means that operations should not be evaluated as soon as they are entered, but rather stored on the stack too. Only when the user enters the evaluation command should the expressions on the stack be evaluated and then the result be pushed on the stack again.

3 Task 2: Design

Describe why you chose the design you have chosen. Why do you think that your design makes for a good reusable component. What did you do to make it easy for potential developers who need to use your calculator?

4 Submission

Submit the source code and textual answer to http://wiki.se.inf.ethz.ch/jp_07/.