Refactoring exercise

You are given a “working” calculator program. It is a quick and dirty programming job.
Make it as pretty as possible!

What is an RPN calculator?

HP produced physical ones when I was young, coolest thing to have…
RPN = reverse Polish notation.
Instead of writing complex expressions with parentheses, let the operator come last

Normal notation	RPN
1 + 2	1 2 +
3 * 5 + 1	3 5 * 1 +
3 * (5 + 1)	3 5 1 + *

The calculator keeps the numbers on a stack.
An entered number is pushed on top of the calculator stack.
An entered operator pops the two top numbers, performs the operation and pushes back the result. Play with it, and you will understand.

Refactoring

Code smells
Find as many smells as you can find, e.g.

- Substandard naming
- Duplicated code
- Long methods with all kinds of responsibilities
- Creating an own DoubleStack ?
- Etc.

Refactor to beauty!

Separation of concerns
The program has a simple GUI written in Swing. Since we have covered that very little, you do not have to delve too much into the Swing code.
However, what if we wanted to have another user interface, like Scanner/System.out.println or maybe even a web user interface.
Can you separate the calculator logic from the presentation code with a Java interface?
And then write a new UI (simplest would be a Scanner/System.out.println one).

Fragile code
Handling of incorrect input is fragile, only works in sunshine. Fix!
Operators invoked with only 1 value on the stack behaves badly. Fix!

This is an exercise. There are many different beautiful solutions!
(Next week is about testing, is the calculator testable?)