
THE JAVA PROGRAMMING LANGUAGE

REVERSE POLISH NOTATION CALCULATOR

University of Wrocław
Institute of Computer Science

Paweł Rzechonek

Subexercise 1

Design an interface `Queue`, with methods to **add** and **remove** elements (tokens). Furthermore, a method to check whether the queue is empty or not should exist.

Design an interface `Stack`, with methods to **push** and **pop** elements (real numbers). Furthermore, a method to check whether the stack is empty or not should exist.

Subexercise 2

Implement the queue and stack using arrays. If the array is completely filled and you try insert new element, throw an exception. If the structure is empty and you try extract an element, throw an exception too.

You should design your own hierarchy of exceptions.

Subexercise 3

Finally write a simple program, which will read RPN expressions from standard input (split each input string into tokens), calculate them, and print results on standard output. Use in your program mentioned data structures.

Note

Use exceptions and assertions in your program. Create your own hierarchy of exception. Put the `RPNException` class on the top of the hierarchy:

```
public class RPNException extends Exception {  
    // ...  
}
```

Prompt

Implement the following interface into the operators and functions:

```
public interface Calculable {  
    public int arity (); // the number of arguments that a function need take  
    public int missingArg (); // the number of missing arguments  
    public void setArg (int x) throws RPNException; // input the next argument into a function  
    public double calculate () throws RPNException; // calculate the value of a function  
}
```

Hint

Some information about RPN can be obtained on the website:

http://en.wikipedia.org/wiki/Reverse_Polish_notation