

## ADS1 Assignment 1

### *Postfix expression calculator*

The task is to make a postfix expression calculator as described in the book chapter 3.6.3 and the slides for lesson 1.

Requirements:

- The postfix expression calculator is implemented using a stack which is an abstract data structure. To implement the stack, use a generic linked list as the underlying data structure. In the linked list it is only necessary to implement the operations required for this assignment.
- The program must provide an API with just one operation:  

```
int evaluateExpression( ArrayList<Token> tokenList )
```

so no GUI is necessary.
- As can be seen, `evaluateExpression` takes an `ArrayList` of `Token`s as input. The `Token`s are either numbers wrapped in an `Operand` class or operators wrapped in an `Operator` class. Both classes are implementations of the `Token` interface.
- When all `Token`s in the list have been evaluated, `getResult` in the calculator will return the result of the calculation which is now top of the stack.
- Finally, `evaluateExpression` returns the result as an integer.
- All public methods must be tested with unit tests. Remember to test that the right exceptions are thrown.
- The assignment must be handed in no later than February 28 at 23.55.

Feedback:

- Feedback will be peer assessment. In practice, this means that you will give feedback on two of your fellow students' work and receive feedback from two of your fellow students.
- Remember to be constructive.
- Feedback must be given no later than March 7 at 23.55.