Project 2. Algorithm Expression Evaluation Demo

[Problem Description]

The expression evaluation is one of the basic program language. And it is a typical example of stack. Please design a program to evaluate an algorithm expression by using operator precedence method (运算符优先法).

[Requirement]

(1) Input the integer expression of correct grammar and without variables by character sequence. Using the relationship of operator precedence in the following table to implement the mixture expression evaluation, and stimulate the transformation of the operator stack, the operand stack, the input and major operations in the evaluation procedure.

$\theta 2$ $\theta 1$	+	-	*	/	(	)	#
+	>	>	<	<	<	>	>
-	>	>	<	<	<	>	>
*	>	>	>	>	<	>	>
/	>	>	>	>	<	>	>
(	<	<	<	<	<	=	---
)	>	>	>	>	---	>	>
#	<	<	<	<	<	---	=

Notice: $\theta 1<\theta 2$ indicate the $\theta 1$ priority is lower than $\theta 2$.

(2) Expand the operator set, for example involution, unary minus, Assignment.

(3) The calculator function and simulation interface (Optional).

[Test Cases]

Expression：

3*(7-2); 8; 1+2+3+4; 88-1*5; 1024/4*8; 1024/(4*8); (20+2)*(6/2);3-3-3; 8/(9-9); 2*(6+2*(3+6*(6+6))); (((6+6)*6+3)*2+6)*2;

[Hints]

(1) Set the operator stack and calculation stack to help to analysis the relation of operators.

(2) When reading expression sequences of characters, it should complete the operator and the operand (integer) recognition and the corresponding operations.

(3) In the operand recognition, it requires to convert the character sequence into the integer form.

(4) Output the operators, the operand stack, input characters and content of the main operations in a proper form.

[Grading]

Implementation: 50% Interface: 30% Coding Style: 20%

Notice: This project will be checked on the experimental lesson in the 9th week (2016.10.25).