

PLC Laboratory 8

Obsah

Type checking of Arithmetic Expressions Using ANTLR II

Input specification

Output specification

Example

Solution

Type checking of Arithmetic Expressions Using ANTLR II

Extend the interpreter of arithmetic expressions from previous [Laboratory 7](#) with the usage of variables.

The basic description is still valid. Expressions contain `+`, `-`, `*`, `/` operators (with common priorities and left associativity) and parentheses. To simplify the task, consider we have only binary operators. There are no unary operators in our language.

Now, we have variables. Their identifiers compose from letters and they have two types: `float` and `int`. Before variables are used, they need to be declared (use the same syntax as in C). The language is extended with `=` operator (lowest priority and right associativity).

Moreover, our expressions have a type. If it contains only integer numbers and variables its type is `int`. In all other cases it is `float`. In other words, if there is a floating point number in our expression, all other integer values will be converted to `float` and the resulting type is `float`.

Also now, in our language we have also a modulo: `%` that works only for integers.

- You can start with following grammar: ANTLR input file (http://linedu.vsb.cz/~beh01/wiki_data/PLC_Lab8_expr.g4)
- Or you can start with following C# codes (containing symbol table and class solving printing of errors): [PLC_Lab8.zip](http://linedu.vsb.cz/~beh01/wiki_data/PLC_Lab8.zip) (http://linedu.vsb.cz/~beh01/wiki_data/PLC_Lab8.zip)

Input specification

In the input, there are expressions and variable declarations, they are written in free formatting. Each expression and declaration ends with semicolon. Numbers can be written similarly to C language constants. It can be either: integer number or a floating point number. Variable declaration starts with the type and follows with a list of declared variables separated by `,`.

Output specification

Go through all expression, if there is any syntactic error in the input, you can stop the computation, if these expressions contain some other (type) error, write these errors.

Example

■ Input

```
int a,b;  
a = b = 15;  
a + b;  
a % b;  
float c;  
c = a + b;  
c + a;  
a = c;  
c + 1.1;  
c % a;
```

■ Output

Your output may be different. Type error's messages are not precisely defined.

```
8:0 - Variable 'a' type is int, but the assigned value is float.  
10:2 - Module can be used only with integers.
```

Solution

- You can download the solution: [PLC_Lab8_solution.zip](http://linedu.vsb.cz/~beh01/wiki_data/PLC_Lab8_solution.zip) (http://linedu.vsb.cz/~beh01/wiki_data/PLC_Lab8_solution.zip)

Citováno z „http://behalek.cs.vsb.cz/wiki/index.php?title=PLC_Laboratory_8&oldid=3798“

Stránka byla naposledy editována 16. 4. 2024 v 11:38.