

May 8, 2014

Jani Viherväs  
jani.vihervas@cs.helsinki.fi

# MINI-JAVA COMPILER

582648 CODE GENERATION

[http://www.cs.helsinki.fi/u/vihavain/k12/compiler\\_project/project/compiler\\_project\\_2012.html](http://www.cs.helsinki.fi/u/vihavain/k12/compiler_project/project/compiler_project_2012.html)

# 1 Overview

## 2 Grammar

<i>&lt; program &gt;</i>	→ <i>&lt; main class &gt;</i> <i>&lt; class declaration &gt;</i> *
<i>&lt; main class &gt;</i>	→ <b>class</b> <i>&lt; new identifier &gt;</i> { <b>public static void main()</b> { <i>&lt; statement &gt;</i> * } }
<i>&lt; class declaration &gt;</i>	→ <b>class</b> <i>&lt; new identifier &gt;</i> [ <b>extends</b> <i>&lt; identifier &gt;</i> ] { <i>&lt; declaration &gt;</i> * }
<i>&lt; declaration &gt;</i>	→ <i>&lt; variable declaration &gt;</i>   <i>&lt; method declaration &gt;</i>
<i>&lt; method declaration &gt;</i>	→ <b>public</b> <i>&lt; type &gt;</i> <i>&lt; new identifier &gt;</i> ( [ <i>&lt; formals &gt;</i> ] ) { <i>&lt; statement &gt;</i> * }
<i>&lt; variable declaration &gt;</i>	→ <i>&lt; type &gt;</i> <i>&lt; new identifier &gt;</i> <i>&lt; variable assignment &gt;</i> ;
<i>&lt; variable assignment &gt;</i>	→ <i>ε</i>   = <i>&lt; expr &gt;</i>
<i>&lt; formals &gt;</i>	→ <i>&lt; type &gt;</i> <i>&lt; new identifier &gt;</i> ( , <i>&lt; type &gt;</i> <i>&lt; new identifier &gt;</i> ) *
<i>&lt; type &gt;</i>	→ <i>&lt; simple type &gt;</i> <i>&lt; array type &gt;</i>
<i>&lt; simple type &gt;</i>	→ <b>int</b>   <b>boolean</b>   <b>void</b>   <i>&lt; type identifier &gt;</i>
<i>&lt; array type &gt;</i>	→ <i>ε</i>   [ ]
<i>&lt; type identifier &gt;</i>	→ <i>&lt; identifier &gt;</i>
<i>&lt; statement &gt;</i>	→ <b>assert</b> ( <i>&lt; expr &gt;</i> );   <i>&lt; local variable declaration &gt;</i>   { <i>&lt; statement &gt;</i> * }   <b>if</b> ( <i>&lt; expr &gt;</i> ) <i>&lt; statement &gt;</i> <i>&lt; else &gt;</i>   <b>while</b> ( <i>&lt; expr &gt;</i> ) <i>&lt; statement &gt;</i>   <b>System.out.println</b> ( <i>&lt; expr &gt;</i> );   <i>&lt; identifier &gt;</i> [ [ <i>&lt; epxr &gt;</i> ] ] = <i>&lt; expr &gt;</i> ;   <b>return</b> <i>&lt; expr &gt;</i> ;   <i>&lt; method invocation &gt;</i> ;
<i>&lt; else &gt;</i>	→ <i>ε</i>   <b>else</b> <i>&lt; statement &gt;</i>
<i>&lt; local variable declaration &gt;</i>	→ <i>&lt; variable declaration &gt;</i>
<i>&lt; method invocation &gt;</i>	→ <i>&lt; expr1 &gt;</i> . <i>&lt; method tail &gt;</i>
<i>&lt; method tail &gt;</i>	→ <b>length</b>   <i>&lt; identifier &gt;</i> ( [ <i>&lt; expr &gt;</i> ( , <i>&lt; expr &gt;</i> ) * ] )
<i>&lt; expr &gt;</i>	→ <i>&lt; expr1 &gt;</i> <i>&lt; expr2 &gt;</i>
<i>&lt; expr1 &gt;</i>	→ <b>new</b> <i>&lt; new &gt;</i>   ! <i>&lt; expr &gt;</i>   - <i>&lt; expr &gt;</i>   ( <i>&lt; expr &gt;</i> )   <i>&lt; identifier &gt;</i>   <i>&lt; integer literal &gt;</i>   <b>this</b>   <b>true</b>   <b>false</b>
<i>&lt; expr2 &gt;</i>	→ <i>ε</i>   [ <i>&lt; expr &gt;</i> ]   . <i>&lt; method tail &gt;</i>   <i>&lt; binary operator &gt;</i> <i>&lt; expr &gt;</i>
<i>&lt; new &gt;</i>	→ <i>&lt; simple type &gt;</i> [ <i>&lt; expr &gt;</i> ]   <i>&lt; type identifier &gt;</i> ( )
<i>&lt; binary operator &gt;</i>	→ &&        <   >   ==   +   -   *   /   %