

Context-free grammar for Minijava variant (komp11)

v 1.1

Reserved words are **bold face**. Terminal and non-terminal symbols are *italics*. Literal strings which are not reserved words are in **typewriter face**.

Please see the course project pages for the latest version of this grammar.

<i>Program</i>	→	<i>MainClass</i> <i>ClassDecl</i> *
<i>MainClass</i>	→	class <i>id</i> { public static void main (String [] <i>id</i>) { <i>Statement</i> * } }
<i>ClassDecl</i>	→	class <i>id</i> { <i>VarDecl</i> * <i>MethodDecl</i> * }
<i>VarDecl</i>	→	<i>Type</i> <i>id</i> ;
<i>MethodDecl</i>	→	public <i>Type</i> <i>id</i> (<i>FormalList</i>) { <i>VarDecl</i> * <i>Statement</i> * return <i>Exp</i> ; }
<i>FormalList</i>	→	<i>Type</i> <i>id</i> <i>FormalRest</i> *
	→	
<i>Formalrest</i>	→	, <i>Type</i> <i>id</i>
<i>Type</i>	→	int []
	→	boolean
	→	int
	→	<i>id</i>
<i>Statement</i>	→	{ <i>Statement</i> * }
	→	if (<i>Exp</i>) <i>Statement</i> else <i>Statement</i>
	→	while (<i>Exp</i>) <i>Statement</i>
	→	System.out.println (<i>Exp</i>) ;
	→	<i>id</i> = <i>Exp</i> ;
	→	<i>id</i> [<i>Exp</i>] = <i>Exp</i> ;
<i>Exp</i>	→	<i>Exp</i> <i>op</i> <i>Exp</i>
	→	<i>Exp</i> [<i>Exp</i>]
	→	<i>Exp</i> . length
	→	<i>Exp</i> . <i>id</i> (<i>ExpList</i>)
	→	integer
	→	true
	→	false
	→	<i>id</i>
	→	this
	→	new int [<i>Exp</i>]
	→	new <i>id</i> ()
	→	! <i>Exp</i>
	→	(<i>Exp</i>)
<i>op</i>	→	&&
	→	<
	→	+
	→	-
	→	*
<i>ExpList</i>	→	<i>Exp</i> <i>ExpRest</i> *
	→	
<i>ExpRest</i>	→	, <i>Exp</i>

Extensions

These are *grammar* extensions. For a full list of extensions, please see project web pages.

Extension 15p

$Statement \rightarrow \mathbf{if} \ (\ Exp \) \ Statement$

Extension 5p

$Type \rightarrow \mathbf{long} \ [\]$
 $\rightarrow \mathbf{long}$

Extension 30p (syntax checks) + 30p (code generation)

$ClassDecl \rightarrow \mathbf{class} \ id \ \mathbf{extends} \ id \ \{ \ VarDecl^* \ MethodDecl^* \}$

Extension 20p (replace first *Statement* production)

$Statement \rightarrow \{ \ VarDecl^* \ Statement^* \}$

Extension 1p per operator

$op \rightarrow \leq$
 $\rightarrow >$
 $\rightarrow \geq$
 $\rightarrow =$
 $\rightarrow \neq$

Extension 2p

$op \rightarrow ||$

Extension *X*p (suggest your own extension!)