```
Package
        minijava;

Helpers
        letter = ['a'..'z'] | ['A'..'Z'] | '_';
        digit  = ['0'..'9'];
        all    = [0..0xffff];
        tab    = 9;
        cr     = 13;
        lf     = 10;
        eol    = cr | lf | cr lf;
        schar  = [ all - [['\' + '"'] + [cr + lf]]];
        sitem  = schar  | '\\' | '\n' | '\t' | '\"';

States
        normal,
        comment;

Tokens
        {normal} iconst = digit+;
        {normal} sconst = '"' sitem* '"';
        {normal} classtok = 'class';
        {normal} public = 'public';
        {normal} static = 'static';
        {normal} return = 'return';
        {normal} if = 'if';
        {normal} else = 'else';
        {normal} while = 'while';
        {normal} true = 'true';
        {normal} false = 'false';
        {normal} this = 'this';
        {normal} new = 'new';
        {normal} null = 'null';
        {normal} length = 'length';
        {normal} print = 'System.out.print';

        {normal} id = letter (letter | digit)*;
        {normal} whitespace = (' ' | eol | tab)+;
        {normal->comment} comment_start = '/*';
        {normal} lparen = '(';
        {normal} rparen = ')';
        {normal} and = '&&';
        {normal} or = '||';
        {normal} lt = '<';
        {normal} le = '<=';
        {normal} gt = '>';
        {normal} ge = '>=';
        {normal} eq = '==';
        {normal} ne = '!=';
        {normal} plus = '+';
        {normal} minus = '-';
        {normal} times = '*';
        {normal} div = '/';
        {normal} mod = '%';
        {normal} lbrack = '[';
        {normal} rbrack = ']';
        {normal} dot = '.';
        {normal} assign = '=';
        {normal} semi = ';';
        {normal} lbrace = '{';
        {normal} rbrace = '}';
        {normal} comma = ',';
        {comment->normal} comment_end = '*/';
        {comment} comment_body = [all- ['*' + '/']]+;
        {comment} comment_star = '*';
        {comment} comment_slash = '/';
        {normal} line_comment = '//' [all - [cr + lf]]* eol;

Ignored Tokens
        whitespace,
```

```
        comment_body,
        comment_star,
        comment_slash,
        comment_start,
        comment_end,
        line_comment;

Productions
        program = public classtok id lbrace maindecl* rbrace
                ;

        maindecl = {var}    privacy static type id semi
                 | {method} privacy static type id lparen paramlist rparen lbrace stmt* r
brace
                 ;

        paramlist = {list} type id param*
                  | {empty}
                  ;

        param = comma type id
              ;

        privacy = {public} public
                | {blank}
                ;

        type = id emptydim*
             ;

        stmt = {while} while lparen expr rparen stmt
             | {decl}  type id semi
             | {block} lbrace stmt* rbrace
             | {if} if lparen expr rparen [thenclause]:stmt else [elseclause]:stmt
             | {expr} expr semi
             | {return} return expr? semi
             | {print} print lparen expr rparen semi
             | {empty} semi
             ;

        expr = {assign} lhs assign expr
             | {expr} expr10
             ;

        expr10 = {or} [left]:expr10 or [right]:expr20
               | {expr} expr20
               ;

        expr20 = {and} [left]:expr20 and [right]:expr30
               | {expr} expr30
               ;

        expr30 = {eq} [left]:expr30 eq [right]:expr40
               | {ne} [left]:expr30 ne [right]:expr40
               | {expr} expr40
               ;

        expr40 = {lt} [left]:expr40 lt [right]:expr50
               | {le} [left]:expr40 le [right]:expr50
               | {ge} [left]:expr40 ge [right]:expr50
               | {gt} [left]:expr40 gt [right]:expr50
               | {expr} expr50
               ;

        expr50 = {plus}  [left]:expr50 plus [right]:term
             | {minus} [left]:expr50 minus [right]: term
             | {term}  term
             ;

        term = {times}  [left]:term times [right]:factor
```

```
             | {div}    [left]:term div [right]:factor
             | {mod}    [left]:term mod [right]:factor
             | {factor} factor
             ;

        factor = {primary} primary
               | {id} id
               | {length}  id dot length
               | {length2}  id dot length lparen rparen
               ;

        primary =  {newarray} new id lbrack expr rbrack emptydim*
                |  {primary2} primary2
                ;

        primary2 = {iconst} iconst
               | {sconst} sconst
               | {null}   null
               | {true}   true
               | {false}  false
               | {parens} lparen expr rparen
               | {call}   id lparen arglist? rparen
               | {arrayref} arrayref
               ;

        arrayref = {name} id lbrack expr rbrack
                 | {primary} primary2 lbrack expr rbrack
                 ;

        lhs = {id} id
            | {arrayref} arrayref
            ;

        arglist = {list} expr arg*
                ;

        arg = comma expr
            ;

        emptydim = lbrack rbrack
                 ;
```