```
1  // package the generated Java class lives in; no need to change this
2  %package "parser";
3
4  // name of the generated Java class; no need to change this
5  %class "Parser";
6
7  // no need to change this either
8  %embed {:
9      // turn off automated error recovery
10     @Override
11     protected void recoverFromError(Symbol token, TokenStream in)
   throws java.io.IOException, Exception {
12         super.recoverFromError(new Symbol(0), in);
13     }
14  :};
15
16  // the list of all terminals; no need to change this
17  %terminals MODULE, INT, VOID, IF, ELSE, WHILE, RETURN, BREAK, BOOLEAN,
    PUBLIC, TRUE, FALSE,
18             INT_LITERAL, STRING_LITERAL, ID, TYPE, IMPORT,
19             PLUS, MINUS, TIMES, DIV, MOD, EQEQ, NEQ, LEQ, GEQ, LT, GT,
20             LBRACKET, RPAREN, COMMA, RCURLY, LCURLY, SEMICOLON,
   RBRACKET, LPAREN, EQL;
21
22  // declaration of start symbol; no need to change this
23  %goal Module;
24
25  Module = MODULE ID LCURLY ImportStmt DeclarationStmt RCURLY;
26  ImportStmt = IMPORT ID SEMICOLON ImportStmt
27             |
28             ;
29  DeclarationStmt = FunctionDeclarationStmt DeclarationStmt
30                 | FieldDeclarationStmt DeclarationStmt
31                 | TypeDeclarationStmt DeclarationStmt
32                 |
33                 ;
34  FunctionDeclarationStmt = Accessibility TypeName ID LPAREN
   ParameterlistStmt RPAREN LCURLY StatementList RCURLY;
35  Accessibility = PUBLIC
36                 |
37                 ;
38  ParameterStmt = TypeName ID;
39  ParameterlistStmt = ParameterStmt ParameterlistStmtTail
40                     |
41                     ;
42  ParameterlistStmtTail = COMMA ParameterStmt ParameterlistStmtTail
43                         |
44                         ;
45  Statement = LocalVariableDeclaration | StatementBlock | IfStmt |
   WhileStmt | BreakStmt | ReturnStmt | ExprStmt;
```

```
46 StatementList = Statement StatementList
47                   |
48                   ;
49 LocalVariableDeclaration = TypeName ID SEMICOLON;
50 StatementBlock = LCURLY StatementList RCURLY;
51 IfStmt = IF LPAREN Expr RPAREN Statement ELSE Statement;
52 WhileStmt = WHILE LPAREN Expr RPAREN Statement;
53 BreakStmt = BREAK SEMICOLON;
54 ReturnStmt = RETURN Expr SEMICOLON
55              | RETURN SEMICOLON
56              ;
57 ExprStmt = Expr SEMICOLON;
58 Expr = Assignment | RHSExpr;
59 ExprList = Expr ExprListTail
60          |
61          ;
62 ExprListTail = COMMA Expr ExprListTail
63                |
64                ;
65 NonEmptyExprList = Expr ExprListTail;
66 Assignment = LHSExpr EQL Expr;
67 RHSExpr = ArithmeticExpr
68         | ArithmeticExpr ComparisionOperator ArithmeticExpr;
69 ArithmeticExpr = ArithmeticExpr AdditiveOperator Term
70                  | Term
71                  ;
72 Term = Term MultiplicativeOperator Factor
73      | Factor
74      ;
75 MultiplicativeOperator = TIMES | DIV | MOD;
76 Factor = MINUS Factor | PrimaryExpr;
77 PrimaryExpr = LHSExpr
78             | FunctionCall
79             | ArrayExpr
80             | STRING_LITERAL
81             | INT_LITERAL
82             | BooleanLiteral
83             | ParenthesisedExpr
84             ;
85 FunctionCall = ID LPAREN ExprList RPAREN;
86 ArrayExpr = LBRACKET NonEmptyExprList RBRACKET;
87 BooleanLiteral = TRUE | FALSE;
88 ParenthesisedExpr = LPAREN Expr RPAREN;
89 AdditiveOperator = PLUS | MINUS;
90 ComparisionOperator = EQEQ | NEQ | LEQ | GEQ | LT | GT;
91 FieldDeclarationStmt = Accessibility TypeName ID SEMICOLON;
92 TypeDeclarationStmt = Accessibility TYPE ID EQL STRING_LITERAL
   SEMICOLON;
93 PrimitiveType = VOID | INT | BOOLEAN;
94
```

```
 95 // Fixing reduce-reduce + shift-reduce conflicts
 96 TypeName = PrimitiveType | ArrayType | ID;
 97 ArrayType = ID LBRACKET RBRACKET
 98             | ArrayType LBRACKET RBRACKET;
 99 LHSExpr = ID
100           | IndexExpr;
101 IndexExpr = ID LBRACKET Expr RBRACKET
102             | IndexExpr LBRACKET Expr RBRACKET;
103 /**/
104
```