```
//import section

import java_cup.runtime.*;
import java.io.*;
import java.util.*;

import SymbolTable.*;

parser code {:

   SymbolTable symbolTable;

:};

init with {:
    symbolTable = new SymbolTable();
:}

//Terminal symbols
terminal PROGRAM, VAR, INTEGER, CHAR, BEGIN,
END, READ, WRITE, IF, THEN, ELSE;
terminal PLUS, MUL, LEFTPAR, RIGHTPAR, DOTDOT,
ASSIGN, SEMI, COMMA, DOT;
terminal String ID;
terminal Integer INTCONST;
terminal Character CHARCONST;


//Nonterminal symbols
non terminal Block Program, Blok, NizNar;
non terminal DeklProm, NizDekl, Deklaracija;
non terminal Statement Naredba;
non terminal Assignment Dodela;
non terminal IfStatement IfNar;
non terminal Expression Izraz, PIzraz, FIzraz;
non terminal Type Tip;
non terminal ArrayList NizImena;
non terminal Constant  Konstanta;

//Grammar

Program ::= PROGRAM DeklProm Blok:b DOT
            {:
             RESULT = b;
            :};

DeklProm ::= VAR NizDekl;

NizDekl ::= NizDekl Deklaracija
          | Deklaracija
          ;

Deklaracija ::= NizImena:niz DOTDOT Tip:t
                {:
                   for (int i=0;
                       i<niz.size(); i++)
                    {
                      String ime;
                      ime=(String)niz.get(i);
                      parser.symbolTable.addVar(
                        ime, t );
                    }
```

```
                 :}
               ;

NizImena ::= NizImena:niz COMMA ID:ime
           {:
             RESULT = niz;
             RESULT.add( ime );
           :}
           | ID:ime
             {:
                RESULT = new ArrayList();
                RESULT.add( ime );
             :}
           ;

Tip ::= INTEGER
        {:
           RESULT = parser.symbolTable.getType(
             "integer" );
        :}
      | CHAR
        {:
           RESULT = parser.symbolTable.getType(
             "char" );
        :}
      ;

Blok ::= BEGIN NizNar:nn END
         {:
             RESULT = nn;
         :};
```

```
NizNar ::= NizNar:nn SEMI Naredba:n
          {:
             RESULT = nn;
             RESULT.addStatement( n );
          :}
         | Naredba:n
           {:
             RESULT = new Block();
             RESULT.addStatement( n );
           :}
         ;

Naredba ::= Dodela:d
            {:
              RESULT = d;
            :}
          | Blok:b
            {:
              RESULT = b;
            :}
          | IfNar:in
            {:
              RESULT = in;
            :}
          ;

Dodela ::= ID:ime ASSIGN Izraz:i
           {:
             Variable var =
               parser.symbolTable.getVar( ime );
             RESULT = new Assignment( var, i );
           :}
         ;

Izraz ::= Izraz:i1 PLUS PIzraz:i2
          {:
             RESULT = new Sum( i1, i2 );
          :}
        | PIzraz:i
          {:
             RESULT = i;
          :}
        ;

PIzraz ::= PIzraz:i1 MUL FIzraz:i2
           {:
              RESULT = new Product( i1, i2 );
           :}
         | FIzraz:i
           {:
              RESULT = i;
           :}
         ;

FIzraz ::= ID:ime
          {:
            Variable var =
              parser.symbolTable.getVar( ime );
            RESULT =
              new VariableExpression(var);
          :}
         | Konstanta:k
           {:
             RESULT = new ConstantExpression(k);
           :}
         | LEFTPAR Izraz:i RIGHTPAR
           {:
             RESULT = i;
           :}
         ;

Konstanta ::= INTCONST:c
              {:
                RESULT = new Constant(
                  parser.symbolTable.getType(
                    "integer" ), c );
              :}
            | CHARCONST:c
              {:
                 RESULT = new Constant(
                    parser.symbolTable.getType(
                       "char" ), c );
              :}
            ;

IfNar ::= IF Izraz:i THEN Naredba:n1 ELSE
          Naredba:n2
          {:
              RESULT = new IfStatement(i,n1,n2);
          :}
        ;
```