

```
<Program>           : <Body>
<Function>          : <Head> <Body> <Tail>
<Head>              : func id ( <Par_decl_list> ) : <Type>
<Tail>              : end id ;
<Type>              : id
| int
| bool
| array of <Type>
| record of { <Var_decl_list> }
<Par_decl_list>    : <Var_decl_list>
|  $\epsilon$ 
<Var_decl_list>   : <Var_decl_list> , <Var_type>
| <Var_type>
<Var_type>          : id : <Type>
<Body>              : <Decl_list> <Statement_list>
<Decl_list>         : <Decl_list> <Declaration> ;
|  $\epsilon$ 
<Declaration>       : type id = <Type>
| <Function>
| var <Var_decl_list>
<Statement_list>   : <Statement_list> <Statement>
| <Statement>
<Statement>          : return <Expression> ;
| write <Expression> ;
| new <Variable> of length <Expression> ;
| new <Variable> ;
| <Variable> = <Expression> ;
| <Variable> += <Expression> ;
| <Variable> -= <Expression> ;
| if <Expression> then <Statement> else <Statement>
| if <Expression> then <Statement>
| while <Expression> do <Statement>
| for <Variable> = <Expression> to <Expression> do <Statement>
| for <Variable> = <Expression> downto <Expression> do <Statement>
| { <Statement_list> }
<Variable>          : id
| <Variable> [ <Expression> ]
| <Variable> . id
<Expression>         : <Term>
| <Expression> * <Expression>
| <Expression> / <Expression>
| <Expression> + <Expression>
| <Expression> - <Expression>
| <Expression> < <Expression>
| <Expression> > <Expression>
| <Expression> <= <Expression>
| <Expression> >= <Expression>
| <Expression> == <Expression>
| <Expression> != <Expression>
| <Expression> && <Expression>
| <Expression> || <Expression>
<Term>              : <Variable>
| - <Variable>
| id ( <Act_list> )
| ( <Expression> )
| ! <Term>
| <Expression> |
| - num
| num
| true
| false
| null
<Act_list>          : <Exp_list>
|  $\epsilon$ 
<Exp_list>          : <Expression>
| <Exp_list> , <Expression>
```