

```
PROGRAM ::= [VARSEQ] [PROCSEQ] main { INSTRSEQ }
VARSEQ ::= ( VARDECL ; ) *
VARDECL ::= var IDLIST : TYPEIDENT
TYPEIDENT ::= integer | float | boolean | string | ATYPE
ATYPE ::= arrayof TYPEIDENT
IDLIST ::= IDENT ( , IDENT ) *
PROCSEQ ::= ( PROCDECL ; ) *
PROCDECL ::= sub IDENT [ ( FPARAMS ) ] [ : TYPEIDENT ] ; { [ VARSEQ ] INSTRSEQ }
FPARAMS ::= IDENT : TYPEIDENT ( , IDENT : TYPEIDENT ) *
EXPR ::= [ + | - ] SIMPEXPR ( RELOP SIMPEXPR ) *
SIMPEXPR ::= TERM ( ADDOP TERM ) *
TERM ::= FACTOR ( MULTOP FACTOR ) *
FACTOR ::= IDENT [ ( LINT | IDENT ) ] | LSTRING | LINT | LBOOL | LREAL | SCALL
RELOP ::= == | < | > | != | >= | <=
ADDOP ::= + | - | or
MULTOP ::= * | / | and
INSTRSEQ ::= ( INSTR ; ) *
INSTR ::= ASSIGN | COND | LOOP | IN | OUT | BLOCK | SCALL | ARRALLOC
ASSIGN ::= IDENT [ ( LINT | IDENT ) ] = EXPR
OUT ::= print ( EXPR )
IN ::= read ( IDENT )
COND ::= if ( EXPR ) { INSTRSEQ }
LOOP ::= do ( EXPR ) { INSTRSEQ }
BLOCK ::= block { [ VARSEQ ] INSTRSEQ }
SCALL ::= call IDENT [ ( APARAMS ) ]
APARAMS ::= EXPR ( , EXPR ) *
ARRALLOC ::= alloc ( IDENT , LINT )
```

Oznaczenia:

- Czcionką pogrubioną zaznaczono symbole terminalne (proszę zwrócić uwagę na pojedyncze znaki, niektóre też są terminalami)
- [] oznacza "opcjonalnie"
- () * oznacza "zero lub więcej razy"
- | oznacza "lub" (czyli kilka produkcji w jednym wierszu)
- IDENT - identyfikator, LSTRING - literał napisowy, LINT - literał całkowity itd.

Uwagi:

- Dodano parametry w podprogramach
- Dodano dynamiczny typ tablicowy o indeksach całkowitych, przykład zastosowania

```
var x:arrayof float;
alloc(x,10); // alokacja tablicy 10-elementowej
x(1) = 2.5;
```
- Dodatkowe założenia leksykalne, składniowe, semantyczne mogą być określone przez studenta **po uprzedniej konsultacji z prowadzącym pracownię**.
- Generacja kodu wynikowego - do wyboru: pseudokod lub kod maszynowy wybranego procesora. Pseudokod wymaga implementacji maszyny wirtualnej
- Gramatyka może wymagać przekształceń lub uzupełnień - wszelkie wątpliwości należy wyjaśniać z prowadzącymi