

2 Πλήρης γραμματική της **Pazcal**^b

Η σύνταξη της γλώσσας **Pazcal**^b δίνεται παρακάτω σε μορφή EBNF. Η γραμματική που ακολουθεί είναι *διφορούμενη*, οι αμφισημίες όμως μπορούν να ξεπεραστούν αν λάβει κανείς υπόψη τους κανόνες προτεραιότητας και προσεταιριστικότητας των τελεστών, όπως περιγράφονται στον πίνακα 2. Τα σύμβολα $\langle \text{id} \rangle$, $\langle \text{int-const} \rangle$, $\langle \text{float-const} \rangle$, $\langle \text{char-const} \rangle$ και $\langle \text{string-literal} \rangle$ είναι τερματικά σύμβολα της γραμματικής και αντιστοιχούν στις λεκτικές μονάδες που περιγράφονται στη ενότητα 1.1.

```

1   $\langle \text{module} \rangle$           ::= (  $\langle \text{declaration} \rangle$  ) *
2   $\langle \text{declaration} \rangle$     ::=  $\langle \text{const\_def} \rangle$  |  $\langle \text{var\_def} \rangle$  |  $\langle \text{routine} \rangle$  |  $\langle \text{program} \rangle$ 
3   $\langle \text{const\_def} \rangle$       ::= “const”  $\langle \text{type} \rangle$   $\langle \text{id} \rangle$  “=”  $\langle \text{const\_expr} \rangle$  ( “,”  $\langle \text{id} \rangle$  “=”  $\langle \text{const\_expr} \rangle$  ) * “;”
4   $\langle \text{var\_def} \rangle$        ::=  $\langle \text{type} \rangle$   $\langle \text{var\_init} \rangle$  ( “,”  $\langle \text{var\_init} \rangle$  ) * “;”
5   $\langle \text{var\_init} \rangle$       ::=  $\langle \text{id} \rangle$  [ “=”  $\langle \text{expr} \rangle$  ] |  $\langle \text{id} \rangle$  ( “[”  $\langle \text{const\_expr} \rangle$  “]” ) +
6   $\langle \text{routine\_header} \rangle$  ::= ( “PROC” | “FUNC”  $\langle \text{type} \rangle$  )  $\langle \text{id} \rangle$  “(” [  $\langle \text{type} \rangle$   $\langle \text{formal} \rangle$  ( “,”  $\langle \text{type} \rangle$   $\langle \text{formal} \rangle$  ) * ] “)”
7   $\langle \text{formal} \rangle$          ::= [ “&” ]  $\langle \text{id} \rangle$  |  $\langle \text{id} \rangle$  “[” [  $\langle \text{const\_expr} \rangle$  ] “]” ( “[”  $\langle \text{const\_expr} \rangle$  “]” ) *
8   $\langle \text{routine} \rangle$         ::=  $\langle \text{routine\_header} \rangle$  ( “;” |  $\langle \text{block} \rangle$  )
9   $\langle \text{program\_header} \rangle$  ::= “PROGRAM”  $\langle \text{id} \rangle$  “(” “)”
10  $\langle \text{program} \rangle$        ::=  $\langle \text{program\_header} \rangle$   $\langle \text{block} \rangle$ 
11  $\langle \text{type} \rangle$            ::= “int” | “bool” | “char” | “REAL”
12  $\langle \text{const\_expr} \rangle$    ::=  $\langle \text{expr} \rangle$ 
13  $\langle \text{expr} \rangle$           ::=  $\langle \text{int-const} \rangle$  |  $\langle \text{float-const} \rangle$  |  $\langle \text{char-const} \rangle$  |  $\langle \text{string-literal} \rangle$  | “true” | “false”
14                        | “(”  $\langle \text{expr} \rangle$  “)” |  $\langle \text{l\_value} \rangle$  |  $\langle \text{call} \rangle$  |  $\langle \text{unop} \rangle$   $\langle \text{expr} \rangle$  |  $\langle \text{expr} \rangle$   $\langle \text{binop} \rangle$   $\langle \text{expr} \rangle$ 
15  $\langle \text{l\_value} \rangle$       ::=  $\langle \text{id} \rangle$  ( “[”  $\langle \text{expr} \rangle$  “]” ) *
16  $\langle \text{unop} \rangle$          ::= “+” | “-” | “!” | “not”
17  $\langle \text{binop} \rangle$         ::= “+” | “-” | “*” | “/” | “%” | “MOD” | “==” | “!=” | “<” | “>” | “<=” | “>=”
18                        | “&&” | “and” | “||” | “or”
19  $\langle \text{call} \rangle$          ::=  $\langle \text{id} \rangle$  “(” [  $\langle \text{expr} \rangle$  ( “,”  $\langle \text{expr} \rangle$  ) * ] “)”
20  $\langle \text{block} \rangle$          ::= “{” (  $\langle \text{local\_def} \rangle$  |  $\langle \text{stmt} \rangle$  ) * “}”
21  $\langle \text{local\_def} \rangle$     ::=  $\langle \text{const\_def} \rangle$  |  $\langle \text{var\_def} \rangle$ 
22  $\langle \text{stmt} \rangle$           ::= “;” |  $\langle \text{l\_value} \rangle$   $\langle \text{assign} \rangle$   $\langle \text{expr} \rangle$  “;” |  $\langle \text{l\_value} \rangle$  ( “++” | “--” ) “;” |  $\langle \text{call} \rangle$  “;”
23                        | “if” “(”  $\langle \text{expr} \rangle$  “)”  $\langle \text{stmt} \rangle$  [ “else”  $\langle \text{stmt} \rangle$  ] | “while” “(”  $\langle \text{expr} \rangle$  “)”  $\langle \text{stmt} \rangle$ 
24                        | “FOR” “(”  $\langle \text{id} \rangle$  “,”  $\langle \text{range} \rangle$  “)”  $\langle \text{stmt} \rangle$  | “do”  $\langle \text{stmt} \rangle$  “while” “(”  $\langle \text{expr} \rangle$  “)” “;”
25                        | “switch” “(”  $\langle \text{expr} \rangle$  “)” “{” ( ( “case”  $\langle \text{const\_expr} \rangle$  “:” ) +  $\langle \text{clause} \rangle$  ) *
26                        [ “default” “:”  $\langle \text{clause} \rangle$  ] “}”
27                        | “break” “;” | “continue” “;” | “return” [  $\langle \text{expr} \rangle$  ] “;” |  $\langle \text{block} \rangle$ 
28                        |  $\langle \text{write} \rangle$  “(” [  $\langle \text{format} \rangle$  ( “,”  $\langle \text{format} \rangle$  ) * ] “)” “;”
29  $\langle \text{assign} \rangle$         ::= “=” | “+=” | “-=” | “*=” | “/=” | “%=”
30  $\langle \text{range} \rangle$         ::=  $\langle \text{expr} \rangle$  ( “TO” | “DOWNTO” )  $\langle \text{expr} \rangle$  [ “STEP”  $\langle \text{expr} \rangle$  ]
31  $\langle \text{clause} \rangle$        ::= (  $\langle \text{stmt} \rangle$  ) * ( “break” “;” | “NEXT” “;” )
32  $\langle \text{write} \rangle$         ::= “WRITE” | “WRITELN” | “WRITESP” | “WRITESPLN”
33  $\langle \text{format} \rangle$        ::=  $\langle \text{expr} \rangle$  | “FORM” “(”  $\langle \text{expr} \rangle$  “,”  $\langle \text{expr} \rangle$  [ “,”  $\langle \text{expr} \rangle$  ] “)”
34
```