- Pazcal Grammar

The following rules are depicting the grammar of Pazcal language in EBNF-form :

```
⟨module⟩          ::= ( ⟨declaration⟩ )*
⟨declaration⟩     ::= ⟨const_def⟩ | ⟨var_def⟩ | ⟨routine⟩ | ⟨program⟩
⟨const_def⟩       ::= “const” ⟨type⟩ ⟨id⟩ “=” ⟨const_expr⟩ ( “,” ⟨id⟩ “=” ⟨const_expr⟩ )* “;”
⟨var_def⟩         ::= ⟨type⟩ ⟨var_init⟩ ( “,” ⟨var_init⟩ )* “;”
⟨var_init⟩        ::= ⟨id⟩ [ “=” ⟨expr⟩ ] | ⟨id⟩ ( “[” ⟨const_expr⟩ “]” )+
⟨routine_header⟩  ::= ( “PROC” | “FUNC” ⟨type⟩ ) ⟨id⟩ “(” [ ⟨type⟩ ⟨formal⟩ ( “,” ⟨type⟩ ⟨formal⟩ )* ] “)”
⟨formal⟩          ::= [ “&” ] ⟨id⟩ | ⟨id⟩ “[” [ ⟨const_expr⟩ ] “]” ( “[” ⟨const_expr⟩ “]” )*
⟨routine⟩         ::= ⟨routine_header⟩ ( “;” | ⟨block⟩ )
⟨program_header⟩  ::= “PROGRAM” ⟨id⟩ “(” “)”
⟨program⟩         ::= ⟨program_header⟩ ⟨block⟩
⟨type⟩            ::= “int” | “bool” | “char” | “REAL”
⟨const_expr⟩      ::= ⟨expr⟩
⟨expr⟩            ::= ⟨int-const⟩ | ⟨float-const⟩ | ⟨char-const⟩ | ⟨string-literal⟩ | “true” | “false”
                    | “(” ⟨expr⟩ “)” | ⟨l_value⟩ | ⟨call⟩ | ⟨unop⟩ ⟨expr⟩ | ⟨expr⟩ ⟨binop⟩ ⟨expr⟩
⟨l_value⟩         ::= ⟨id⟩ ( “[” ⟨expr⟩ “]” )*
⟨unop⟩            ::= “+” | “-” | “!” | “not”
⟨binop⟩           ::= “+” | “-” | “*” | “/” | “%” | “MOD” | “==” | “!=” | “<” | “>” | “<=” | “>=”
                    | “&&” | “and” | “||” | “or”
⟨call⟩            ::= ⟨id⟩ “(” [ ⟨expr⟩ ( “,” ⟨expr⟩ )* ] “)”
⟨block⟩           ::= “{” ( ⟨local_def⟩ | ⟨stmt⟩ )* “}”
⟨local_def⟩       ::= ⟨const_def⟩ | ⟨var_def⟩
⟨stmt⟩            ::= “;” | ⟨l_value⟩ ⟨assign⟩ ⟨expr⟩ “;” | ⟨l_value⟩ ( “++” | “--” ) “;” | ⟨call⟩ “;”
                    | “if” “(” ⟨expr⟩ “)” ⟨stmt⟩ [ “else” ⟨stmt⟩ ] | “while” “(” ⟨expr⟩ “)” ⟨stmt⟩
                    | “FOR” “(” ⟨id⟩ “,” ⟨range⟩ “)” ⟨stmt⟩ | “do” ⟨stmt⟩ “while” “(” ⟨expr⟩ “)” “;”
                    | “switch” “(” ⟨expr⟩ “)” “{” ( ( “case” ⟨const_expr⟩ “:” )+ ⟨clause⟩ )*
                          [ “default” “:” ⟨clause⟩ ] “}”
                    | “break” “;” | “continue” “;” | “return” [ ⟨expr⟩ ] “;”
                    | ⟨write⟩ “(” [ ⟨format⟩ ( “,” ⟨format⟩ )* ] “)” “;”
⟨assign⟩          ::= “=” | “+=” | “-=” | “*=” | “/=” | “%=”
⟨range⟩           ::= ⟨expr⟩ ( “TO” | “DOWNTO” ) ⟨expr⟩ [ “STEP” ⟨expr⟩ ]
⟨clause⟩          ::= ( ⟨stmt⟩ )* ( “break” “;” | “NEXT” “;” )
⟨write⟩           ::= “WRITE” | “WRITELN” | “WRITESP” | “WRITESPLN”
⟨format⟩          ::= ⟨expr⟩ | “FORM” “(” ⟨expr⟩ “,” ⟨expr⟩ [ “,” ⟨expr⟩ ] “)”
```