

---

# Pascal

## ISO 7185:1990

---

This online copy of the unextended Pascal standard is provided only as an aid to standardization. In the case of differences between this online version and the printed version, the printed version takes precedence.

Do not modify this document. Do not include this document in another software product. You may print this document for personal use only. Do not sell this document.

Use this information only for good; never for evil. Do not expose to fire. Do not operate heavy equipment after reading, may cause drowsiness. Do not read under the influence of alcohol (although there have been several unconfirmed reports that alcohol actually improves the readability). The standard is written in English. If you have trouble understanding a particular section, read it again and again and again... Sit up straight. Eat your vegetables. Do not mumble.

©ISO/IEC 1991

## Annex A

(Informative)

### Collected syntax

The nonterminal symbols pointer-type, program, signed-number, simple-type, special-symbol, and structured-type are only referenced by the semantics and are not used in the right-hand-side of any production. The nonterminal symbol program is the start symbol of the grammar.

- 6.7.3 actual-parameter = expression | variable-access | procedure-identifier  
| function-identifier .
- 6.7.3 actual-parameter-list = '(' actual-parameter { ',' actual-parameter } ')'
- 6.7.2.1 adding-operator = '+' | '-' | 'or' .
- 6.1.7 apostrophe-image = "'" .
- 6.4.3.2 array-type = 'array' '[' index-type { ',' index-type } ']' 'of' component-type .
- 6.5.3.2 array-variable = variable-access .
- 6.8.2.2 assignment-statement = ( variable-access | function-identifier ) ':=' expression .
- 6.4.3.4 base-type = ordinal-type .
- 6.2.1 block = label-declaration-part constant-definition-part type-definition-part  
variable-declaration-part procedure-and-function-declaration-part  
statement-part .
- 6.7.2.3 Boolean-expression = expression .
- 6.6.3.7.1 bound-identifier = identifier .
- 6.5.5 buffer-variable = file-variable '↑' .
- 6.4.3.3 case-constant = constant .
- 6.4.3.3 case-constant-list = case-constant { ',' case-constant } .
- 6.8.3.5 case-index = expression .
- 6.8.3.5 case-list-element = case-constant-list ':' statement .
- 6.8.3.5 case-statement = 'case' case-index 'of' case-list-element  
{ ',' case-list-element } [ ';' ] 'end' .
- 6.1.7 character-string = "" string-element { string-element } "" .
- 6.4.3.2 component-type = type-denoter .
- 6.5.3.1 component-variable = indexed-variable | field-designator .
- 6.8.3.2 compound-statement = 'begin' statement-sequence 'end' .
- 6.8.3.3 conditional-statement = if-statement | case-statement .

## ISO/IEC 7185:1990(E)

- 6.6.3.7.1 conformant-array-parameter-specification = value-conformant-array-specification  
| variable-conformant-array-specification .
- 6.6.3.7.1 conformant-array-schema = packed-conformant-array-schema  
| unpacked-conformant-array-schema .
- 6.3 constant = [ sign ] ( unsigned-number | constant-identifier )  
| character-string .
- 6.3 constant-definition = identifier '=' constant .
- 6.2.1 constant-definition-part = [ 'const' constant-definition ';' { constant-definition ';' } ] .
- 6.3 constant-identifier = identifier .
- 6.8.3.9 control-variable = entire-variable .
- 6.1.1 digit = '0' | '1' | '2' | '3' | '4' | '5' | '6' | '7' | '8' | '9' .
- 6.1.5 digit-sequence = digit { digit } .
- 6.1.4 directive = letter { letter | digit } .
- 6.4.4 domain-type = type-identifier .
- 6.8.3.4 else-part = 'else' statement .
- 6.8.2.1 empty-statement = .
- 6.5.2 entire-variable = variable-identifier .
- 6.4.2.3 enumerated-type = '(' identifier-list ')' .
- 6.7.1 expression = simple-expression [ relational-operator simple-expression ] .
- 6.6.3.7.1 factor > bound-identifier .
- 6.7.1 factor > variable-access | unsigned-constant | function-designator  
| set-constructor | '(' expression ')' | 'not' factor .
- 6.5.3.3 field-designator = record-variable '.' field-specifier | field-designator-identifier .
- 6.8.3.10 field-designator-identifier = identifier .
- 6.5.3.3 field-identifier = identifier .
- 6.4.3.3 field-list = [ ( fixed-part [ ';' variant-part ] | variant-part ) [ ';' ] ] .
- 6.5.3.3 field-specifier = field-identifier .
- 6.4.3.5 file-type = 'file' 'of' component-type .
- 6.5.5 file-variable = variable-access .
- 6.8.3.9 final-value = expression .
- 6.4.3.3 fixed-part = record-section { ';' record-section } .

- 6.8.3.9 for-statement = 'for' control-variable ':=' initial-value ( 'to' | 'downto' ) final-value  
'do' statement .
- 6.6.3.1 formal-parameter-list = '(' formal-parameter-section { ';' formal-parameter-section } ')'
- 6.6.3.1 formal-parameter-section > value-parameter-specification  
| variable-parameter-specification  
| procedural-parameter-specification  
| functional-parameter-specification .
- 6.6.3.7.1 formal-parameter-section > conformant-array-parameter-specification .
- 6.1.5 fractional-part = digit-sequence .
- 6.6.2 function-block = block .
- 6.6.2 function-declaration = function-heading ';' directive  
| function-identification ';' function-block  
| function-heading ';' function-block .
- 6.7.3 function-designator = function-identifier [ actual-parameter-list ] .
- 6.6.2 function-heading = 'function' identifier [ formal-parameter-list ] ':' result-type .
- 6.6.2 function-identification = 'function' function-identifier .
- 6.6.2 function-identifier = identifier .
- 6.6.3.1 functional-parameter-specification = function-heading .
- 6.8.2.4 goto-statement = 'goto' label .
- 6.5.4 identified-variable = pointer-variable '↑' .
- 6.1.3 identifier = letter { letter | digit } .
- 6.4.2.3 identifier-list = identifier { ';' identifier } .
- 6.8.3.4 if-statement = 'if' Boolean-expression 'then' statement [ else-part ] .
- 6.5.3.2 index-expression = expression .
- 6.4.3.2 index-type = ordinal-type .
- 6.6.3.7.1 index-type-specification = identifier '..' identifier ':' ordinal-type-identifier .
- 6.5.3.2 indexed-variable = array-variable '[' index-expression, { ';' index-expression } ']' .
- 6.8.3.9 initial-value = expression .
- 6.1.6 label = digit-sequence .
- 6.2.1 label-declaration-part = [ 'label' label { ',' label } ';' ] .
- 6.1.1 letter = 'a' | 'b' | 'c' | 'd' | 'e' | 'f' | 'g' | 'h' | 'i' | 'j'  
| 'k' | 'l' | 'm' | 'n' | 'o' | 'p' | 'q' | 'r' | 's' | 't'  
| 'u' | 'v' | 'w' | 'x' | 'y' | 'z' .

## ISO/IEC 7185:1990(E)

- 6.7.1 member-designator = expression [ `‘.’` expression ] .
- 6.7.2.1 multiplying-operator = `‘*’` | `‘/’` | `‘div’` | `‘mod’` | `‘and’` .
- 6.4.2.1 new-ordinal-type = enumerated-type | subrange-type .
- 6.4.4 new-pointer-type = `‘↑’` domain-type .
- 6.4.3.1 new-structured-type = [ `‘packed’` ] unpacked-structured-type .
- 6.4.1 new-type = new-ordinal-type | new-structured-type | new-pointer-type .
- 6.4.2.1 ordinal-type = new-ordinal-type | ordinal-type-identifier .
- 6.4.2.1 ordinal-type-identifier = type-identifier .
- 6.6.3.7.1 packed-conformant-array-schema = `‘packed’` `‘array’` [ `‘[’` index-type-specification `‘]’`  
`‘of’` type-identifier .
- 6.4.4 pointer-type = new-pointer-type | pointer-type-identifier .
- 6.4.1 pointer-type-identifier = type-identifier .
- 6.5.4 pointer-variable = variable-access .
- 6.6.3.1 procedural-parameter-specification = procedure-heading .
- 6.2.1 procedure-and-function-declaration-part = { ( procedure-declaration  
| function-declaration ) `‘;’` } .
- 6.6.1 procedure-block = block .
- 6.6.1 procedure-declaration = procedure-heading `‘;’` directive  
| procedure-identification `‘;’` procedure-block  
| procedure-heading `‘;’` procedure-block .
- 6.6.1 procedure-heading = `‘procedure’` identifier [ formal-parameter-list ] .
- 6.6.1 procedure-identification = `‘procedure’` procedure-identifier .
- 6.6.1 procedure-identifier = identifier .
- 6.8.2.3 procedure-statement = procedure-identifier ( [ actual-parameter-list  
| read-parameter-list | readln-parameter-list  
| write-parameter-list | writeln-parameter-list ] ) .
- 6.10 program = program-heading `‘;’` program-block `‘.’` .
- 6.10 program-block = block .
- 6.10 program-heading = `‘program’` identifier [ `‘(’` program-parameter-list `‘)’` ] .
- 6.10 program-parameter-list = identifier-list .
- 6.9.1 read-parameter-list = `‘(’` [ file-variable `‘,’` ] variable-access { `‘,’` variable-access } `‘)’` .

- 6.9.2 readln-parameter-list = [ '(' ( file-variable | variable-access )  
   { ',' variable-access } ')' ] .
- 6.4.2.1 real-type-identifier = type-identifier .
- 6.4.3.3 record-section = identifier-list ':' type-denoter .
- 6.4.3.3 record-type = 'record' field-list 'end' .
- 6.5.3.3 record-variable = variable-access .
- 6.8.3.10 record-variable-list = record-variable { ',' record-variable } .
- 6.7.2.1 relational-operator = '=' | '<>' | '<' | '>' | '<=' | '>=' | 'in' .
- 6.8.3.7 repeat-statement = 'repeat' statement-sequence 'until' Boolean-expression .
- 6.8.3.6 repetitive-statement = repeat-statement | while-statement | for-statement .
- 6.6.2 result-type = simple-type-identifier | pointer-type-identifier .
- 6.1.5 scale-factor = [ sign ] digit-sequence .
- 6.7.1 set-constructor = '[' [ member-designator { ',' member-designator } ] ']' .
- 6.4.3.4 set-type = 'set' 'of' base-type .
- 6.1.5 sign = '+' | '-' .
- 6.1.5 signed-integer = [ sign ] unsigned-integer .
- 6.1.5 signed-number = signed-integer | signed-real .
- 6.1.5 signed-real = [ sign ] unsigned-real .
- 6.7.1 simple-expression = [ sign ] term { adding-operator term } .
- 6.8.2.1 simple-statement = empty-statement | assignment-statement  
   | procedure-statement | goto-statement .
- 6.4.2.1 simple-type = ordinal-type | real-type-identifier .
- 6.4.1 simple-type-identifier = type-identifier .
- 6.1.2 special-symbol = '+' | '-' | '\*' | '/' | '=' | '<' | '>' | '[' | ']'  
   | ':' | ';' | ':' | ',' | '↑' | '(' | ')'  
   | '<>' | '<=' | '>=' | ':=' | '..' | word-symbol .
- 6.8.1 statement = [ label ':' ] ( simple-statement | structured-statement ) .
- 6.2.1 statement-part = compound-statement .
- 6.8.3.1 statement-sequence = statement { ',' statement } .
- 6.1.7 string-character = one-of-a-set-of-implementation-defined-characters .
- 6.1.7 string-element = apostrophe-image | string-character .

## ISO/IEC 7185:1990(E)

- 6.8.3.1 structured-statement = compound-statement | conditional-statement  
| repetitive-statement | with-statement .
- 6.4.3.1 structured-type = new-structured-type | structured-type-identifier .
- 6.4.1 structured-type-identifier = type-identifier .
- 6.4.2.4 subrange-type = constant ‘.’ constant .
- 6.4.3.3 tag-field = identifier .
- 6.4.3.3 tag-type = ordinal-type-identifier .
- 6.7.1 term = factor { multiplying-operator factor } .
- 6.4.1 type-definition = identifier ‘=’ type-denoter .
- 6.2.1 type-definition-part = [ ‘type’ type-definition ‘;’ { type-definition ‘;’ } ] .
- 6.4.1 type-denoter = type-identifier | new-type .
- 6.4.1 type-identifier = identifier .
- 6.6.3.7.1 unpacked-conformant-array-schema =  
‘array’ ‘[’ index-type-specification { ‘;’ index-type-specification } ‘]’  
‘of’ ( type-identifier | conformant-array-schema ) .
- 6.4.3.1 unpacked-structured-type = array-type | record-type | set-type | file-type .
- 6.7.1 unsigned-constant = unsigned-number | character-string | constant-identifier | ‘nil’ .
- 6.1.5 unsigned-integer = digit-sequence .
- 6.1.5 unsigned-number = unsigned-integer | unsigned-real .
- 6.1.5 unsigned-real = digit-sequence ‘.’ fractional-part [ ‘e’ scale-factor ]  
| digit-sequence ‘e’ scale-factor .
- 6.6.3.7.1 value-conformant-array-specification = identifier-list ‘:’ conformant-array-schema .
- 6.6.3.1 value-parameter-specification = identifier-list ‘:’ type-identifier .
- 6.5.1 variable-access = entire-variable | component-variable | identified-variable  
| buffer-variable .
- 6.6.3.7.1 variable-conformant-array-specification = ‘var’ identifier-list ‘:’ conformant-array-schema .
- 6.5.1 variable-declaration = identifier-list ‘:’ type-denoter .
- 6.2.1 variable-declaration-part = [ ‘var’ variable-declaration ‘;’ { variable-declaration ‘;’ } ] .
- 6.5.2 variable-identifier = identifier .
- 6.6.3.1 variable-parameter-specification = ‘var’ identifier-list ‘:’ type-identifier .
- 6.4.3.3 variant = case-constant-list ‘:’ ‘(’ field-list ‘)’ .

- 6.4.3.3 variant-part = 'case' variant-selector 'of' variant { ';' variant } .
- 6.4.3.3 variant-selector = [ tag-field ':' ] tag-type .
- 6.8.3.8 while-statement = 'while' Boolean-expression 'do' statement .
- 6.8.3.10 with-statement = 'with' record-variable-list 'do' statement .
- 6.1.2 word-symbol = 'and' | 'array' | 'begin' | 'case' | 'const' | 'div'  
 | 'do' | 'downto' | 'else' | 'end' | 'file' | 'for'  
 | 'function' | 'goto' | 'if' | 'in' | 'label' | 'mod'  
 | 'nil' | 'not' | 'of' | 'or' | 'packed' | 'procedure'  
 | 'program' | 'record' | 'repeat' | 'set' | 'then'  
 | 'to' | 'type' | 'until' | 'var' | 'while' | 'with' .
- 6.9.3 write-parameter = expression [ ':' expression [ ':' expression ] ] .
- 6.9.3 write-parameter-list = '(' [ file-variable ';' ] write-parameter  
 { ',' write-parameter } ')' .
- 6.9.4 writeln-parameter-list = [ '(' ( file-variable | write-parameter )  
 { ',' write-parameter } ')' ] .