PROGRAM -> BLOCK

BLOCK -> **block** DECLARATIONS **begin** STATEMENTS **end**

DECLARATIONS -> DECLARATION DECLARATIONS\_TAG

DECLARATIONS\_TAG -> ϵ | ; DECLARATIONS

DECLARATION -> VAR\_DECLARATION | TYPE\_DECLARATION

VAR\_DECLARATION -> id: VAR\_DECLARATION\_TAG   
VAR\_DECLARATION\_TAG -> SIMPLE\_TYPE | **array** [SIZE] **of** SIMPLE\_TYPE | type\_name

SIZE -> int\_num

SIMPLE\_TYPE -> **integer** | **real**

TYPE\_DECLARATION -> **type** type\_name **is** TYPE\_INDICATOR

TYPE\_INDICATOR -> ENUM\_TYPE | STRUCTURE\_TYPE

ENUM\_TYPE -> **enum** { ID\_LIST }

ID\_LIST -> id ID\_LIST\_TAG   
ID\_LIST\_TAG -> ϵ | , ID\_LIST

STRUCTURE\_TYPE -> **struct** { FIELDS }

FIELDS -> FIELD FIELDS\_TAG

FIELDS\_TAG -> ; FIELD FIELDS\_TAG | ϵ

FIELD -> VAR\_DECLARATION

STATEMENTS -> STATEMENT STATEMENTS\_TAG   
STATEMENTS\_TAG -> ; STATEMENT STATEMENTS\_TAG | ϵ

STATEMENT ->

VAR\_ELEMENT = EXPRESSION |

**switch** (KEY) { CASE\_LIST; **default** : STATEMENTS } |

**break** |

BLOCK

VAR\_ELEMENT -> id | id[EXPRESSION] | FIELD\_ACCESS

KEY -> VAR\_ELEMENT

CASE\_LIST -> CASE CASE\_LIST\_TAG   
CASE\_LIST\_TAG -> CASE CASE\_LIST\_TAG | ϵ

CASE -> **case** KEY\_VALUE : { STATEMENTS }

KEY\_VALUE -> int\_num | id

FIELD\_ACCESS -> id. FIELD\_ACCESS\_TAG

FIELD\_ACCESS\_TAG -> id | FIELD\_ACCESS

EXPRESSION -> SIMPLE\_EXPRESSION EXPRESSION\_TAG

EXPRESSION\_TAG -> ar\_op EXPRESSION | ϵ

SIMPLE\_EXPRESSION -> int\_num | real\_num | VAR\_ELEMENT

DECLARATIONS -> DECLARATION | DECLARATION ; DECLARATIONS

DECLARATIONS -> DECLARATION DECLARATIONS\_TAG

DECLARATIONS\_TAG -> ϵ | ; DECLARATIONS

VAR\_DECLARATION -> id : SIMPLE\_TYPE |

id : array [SIZE] of SIMPLE\_TYPE |

id : type\_name

VAR\_DECLARATION -> id: VAR\_DECLARATION\_TAG   
VAR\_DECLARATION\_TAG -> SIMPLE\_TYPE | array [SIZE] of SIMPLE\_TYPE | type\_name

ID\_LIST -> id | id , ID\_LIST

ID\_LIST -> id ID\_LIST\_TAG   
ID\_LIST\_TAG -> ϵ | , ID\_LIST

FIELDS -> FIELDS; FIELD | FIELD  
FIELDS -> FIELD FIELDS\_TAG

FIELDS\_TAG -> ; FIELD FIELDS\_TAG | ϵ

STATEMENTS -> STATEMENT | STATEMENTS; STATEMENT

STATEMENTS -> STATEMENT STATEMENTS\_TAG   
STATEMENTS\_TAG -> ; STATEMENT STATEMENTS\_TAG | ϵ

CASE\_LIST -> CASE | CASE\_LIST CASE

CASE\_LIST -> CASE CASE\_LIST\_TAG   
CASE\_LIST\_TAG -> CASE CASE\_LIST\_TAG | ϵ

FIELD\_ACCESS -> id.id | id.FIELD\_ACCESS  
FIELD\_ACCESS -> id. FIELD\_ACCESS\_TAG

FIELD\_ACCESS\_TAG -> id | FIELD\_ACCESS

EXPRESSION -> SIMPLE\_EXPRESSION | SIMPLE\_EXPRESSION ar\_op EXPRESSION

EXPRESSION -> SIMPLE\_EXPRESSION EXPRESSION\_TAG

EXPRESSION\_TAG -> ar\_op EXPRESSION | ϵ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | **First** | **Follow** | **Nullable** |
| 1 | **PROGRAM** | block | EOF |  |
| 2 | **BLOCK** | block | EOF | end | } |  |
| 3 | **DECLARATIONS** | Id | type | begin |  |
| 5 | **DECLARATIONS\_TAG** | ; | begin | Yes |
| 6 | **DECLARATION** | Id | type | ; | begin |  |
| 7 | **VAR\_DECLARATION** | Id | ; | begin| } |  |
| 8 | **VAR\_DECLARATION\_TAG** | array | type\_name | integer | real | ;| begin| } |  |
| 9 | **SIZE** | int\_num | [ |  |
| 10 | **SIMPLE\_TYPE** | integer | real | ; |  |
| 11 | **TYPE\_DECLARATION** | type | ; |  |
| 12 | **TYPE\_INDICATOR** | enum | struct | ; |  |
| 13 | **ENUM\_TYPE** | enum | ; |  |
| 14 | **ID\_LIST** | Id | } |  |
| 15 | **ID\_LIST\_TAG** | , | { | Yes |
| 16 | **STRUCTURE\_TYPE** | struct | ; |  |
| 17 | **FIELDS** | Id | } |  |
| 18 | **FIELDS\_TAG** | ; | { | Yes |
| 19 | **FIELD** | Id | ; | } |  |
| 20 | **STATEMENTS** | switch | break | id | end | } |  |
| 21 | **STATEMENTS\_TAG** | ; | end | } | Yes |
| 22 | **STATEMENT** | switch | break | id |block | ; | end | } |  |
| 23 | **VAR\_ELEMENT** | Id | = | ar\_op | ] | ; | ) |  |
| 24 | **KEY** | Id | ) |  |
| 25 | **CASE\_LIST** | case | ; |  |
| 26 | **CASE\_LIST\_TAG** | case | ; | Yes |
| 27 | **CASE** | case | case | Int\_num | id |  |
| 28 | **KEY\_VALUE** | Int\_num | id | : |  |
| 29 | **FIELD\_ACCESS** | Id | = | ar\_op | ] | ; |) |  |
| 30 | **FIELD\_ACCESS\_TAG** | id | = | ar\_op | ] | ; |) |  |
| 31 | **EXPRESSION** | Int\_num | real\_num | Id | ] | ; | end | } |  |
| 32 | **EXPRESSION\_TAG** | ar\_op | ] | ; | end | } | Yes |
| 33 | **SIMPLE\_EXPRESSION** | Int\_num | real\_num | Id | ar\_op| ] | ; |  |