CANDET ANDREI GABRIEL

932

LAB 1b

Alphabet:

a. Upper (A-Z) and lower case letters (a-z) of the English alphabet
b. Underline character "_"
c. Decimal digits (0-9)

1. Lexic
a. Special symbols, representing:
- Operators + - * / := < <= = >= === ~
- Separators [ ] { } : ; *space*
- Reserved words: **array, map, const, do, else, if, int, elif while, for, range, class, struct, string, float, char, boolean, input, print, return, fun, key, value, main, entry**

b. Identifiers
- A sequence of letters and digits, such that the first character is a letter; the rule is:

Identifier ::= letter | letter{letter}{digit}
Letter ::= "A" | "B" | … | "Z" | "a" | "b" | … | "z"
Digit ::= "0" | "1" | … | "9"

c. Constants
1. Integer - rule:

integer_const = "0" | [" + " | " - "] nonzerodigit { "0" | nonzerodigit }
nonzero_digit = "1" | … | "9"

2. Character
character:= 'letter | digit | SPACE '

3. String
Std ::= " char{str} "
2. Syntax

Predefined tokens are emphasized.

Program ::= **entry** cmpdstmt **;**

Type ::= **bool** | **int** | **char** | **string**
Assignstmt ::= IDENTIFIER **=** expression
decl ::= declstmt | declasgnstmt
Declstmt ::= type IDENTIFIER
Declasgnstmt ::= type IDENTIFIER **=** expression

Cmpdstmt ::= **{** stmtlist **}**
Stmtlist ::= stmt | stmt **;** stmtlist
Stmt ::= simplstmt | structstmt
Simplstmt ::= assignstmt | decl

Expression ::= expression **+** term | term
Term ::= term ***** factor | factor | arrayAccess
Factor ::= **(** expression **)** | IDENTIFIER

Structstmt ::= cmpdstmt | ifstmt | whilestmt
Whilestmt ::= **while (** condition **)** cmpdstmt
Ifstmt**::= if(** CONDITION **)** cmpdstmt **else** cmpdstmt
Condition ::= expression RELATION expression
Relation ::= **<** | **<=** | **==** | **>=** | **===** | **>**

Arraydecl ::= type IDENTIFIER **[** number **]**
arrayAccess::= IDENTIFIER[IDENTIFIER]
Mapdecl ::= **map{** type : type **}**
mapAccess ::= IDENTIFIER[IDENTIFIER]

TOKEN LIST:

+ - * / := < <= = >= === ~ % & ^ **array, map, const, do, else, if, int, elif while, for, range, class, struct, string, float, char, boolean, input, print, return, fun, key, value, main, entry**