**Lexic**

**Alphabet**

* uppercase (A-Z) and lowercase (a-z) letters of the English alphabet;
* decimal digits (0-9)
* special characters
  + [space]
  + ;
  + “
  + +
  + –
  + \*
  + /
  + =
  + %
  + &
  + |

**Lexic**

* special symbols, representing:
  + operators
    - +
    - -
    - \*
    - /
    - =
  + separators
    - [space]
  + reserved words
    - main
    - read
    - write
    - int
    - string
    - equals
    - lessEqual
    - lessThan
    - greaterEqual
    - greaterThan
    - if
    - for
    - return
* identifiers
  + a sequence of letters and digits, such that the first character is a letter
  + the rule is:

identifier = letter { digit | letter }

digit = "0"|"1"|...|"9"

letter = "A" | ... | "Z" | “a” | … | “z”

* constants
  + integer (int):

constInt := “0” | "+"int | "-" int | int

int := nonZeroDigit { digit }

nonZeroDigit = “1” | … | “9”

* + string

str := ""{ letter | digit |otherCharacter }""

otherCharacter = [space] | \_ | . | , | : | ; | ! | ? | ‘ | #

**Syntax**

The words are predefined tokens are specified between two double quotes (“).

**Syntactical rules**

program ::= "int main() {" statementList "; return 0; }"

type ::= "int" | "string"

relation = “equals” | “lessEqual” | “lessThan” | “greaterEqual” | “greaterThan”

operand = IDENTIFIER | CONSTANT

operator = “+” | “-“ | “\*” | “/” | “%” | “=”

size = 1 | 2 | … | 1000

statementList ::= statement | statement “;” statementList

statement ::= simpleStatement | structuredStatement

simpleStatement ::= declaration | io | assignment | condition

declaration ::= type IDENTIFIER | type IDENTIFIER “[“ size “]”

io ::= “read” IDENTIFIER | “write” IDENTIFIER | “write” CONSTANT

assignment ::= IDENTIFIER “=” expression

expression = operand operator operand | IDENTIFIER | CONSTANT

condition ::= operand relation operand

structuredStatement ::= if | for

if::= “if (“ condition “) {“ statementList “}”

for ::= “for (” assignment “;” condition “;” assignment “) {“ statementList “}”

**Tokens**

|  |  |
| --- | --- |
| **Token** | **Code** |
| main | 2 |
| return | 3 |
| read | 4 |
| write | 5 |
| equals | 10 |
| lessEqual | 11 |
| lessThan | 12 |
| greaterEqual | 13 |
| greaterThan | 14 |
| if | 20 |
| for | 22 |
| int | 30 |
| string | 31 |
| ; | 40 |
| { | 41 |
| } | 42 |
| ( | 43 |
| ) | 44 |
| [ | 45 |
| ] | 46 |
| + | 50 |
| - | 51 |
| \* | 52 |
| / | 53 |
| % | 54 |
| = | 55 |