- all programs must consist of two parts: `DECLARATION` and `MAIN`; and must end with the keyword `END`. Thus, a program has the following structure:
  `DECLARATION … MAIN … END`
- all variable declarations must be done in the `DECLARATION` part, no assignments can be done during declarations
- all other actions must be done in the `MAIN` part
- comments can be written in the following format in one line:
  - `# I'm a valid comment #`
- comments can't be written before `DECLARATION` or after `END`
- comments can't be written in the middle of a statement
- all statements must be ended by `"!"`
- all programs can be terminated anywhere with statement `EXIT!`
- command `IN` indicates that an input is expected; must be used only in assignment
- command `OUT` indicates that an output is expected; must be used as a separate statement
- there are 3 types: `"Z", "Q", "CHAR"`
- variable name may contain lowercase letters, digits and underscores, and must start with a lowercase letter or underscore
- only one declaration can be done within one statement
- all variables must be declared in the following format:
  - `CHAR char_var!`
  - `Z int198!`
  - `Q _float!`
- only one assignment can be done within one statement
- assignment must be done in the following way:
  - `char_var := 'k'!`
  - `_float := (-275.6 + 23) : 9!`
- there are 4 arithmetic operators: `+, -, *, :`
- there are 2 logical operators: `/\, \/`
- there are 6 comparison operators: `<, >, <=, >=, =, X=`
- an assignment may include any arithmetic, logical or comparison operators
- the number of variables or constants in an expression is unlimited
- operator precedence:

Operator	Description
`()`	Parentheses
`+, -`	Unary +, unary -
`*, :`	Multiplication, division
`+, -`	Addition, subtraction
`<, >, <=, >=, =, X=`	Comparison
`/\`	Logical AND
`\/`	Logical OR
`:=`	Assignment

- conditional statement can be written in the following format:

```
# lines 3 and 4 are optional #
[a > b] ->
      max := a!
--
      max := b!
DONE
```

- loop structure `WHENEVER` can be written in the following format:

```
# assuming that i was declared in the DECLARATION part #
i := 0!
WHENEVER [i < 10]
      OUT i!
      i := i + 1!
DONE
```

- any logical or arithmetic expression can be written in square brackets
- any kind of statements can be written in the body part of conditional statement and loop structure