

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ
СІКОРСЬКОГО»**

**Факультет інформатики та обчислювальної техніки
Кафедра інформатики та програмної інженерії**

**Лабораторна робота № 1
ПРОЄКТУВАННЯ ТА СПЕЦИФІКАЦІЯ
ПРОСТОЇ ІМПЕРАТИВНОЇ МОВИ ПРОГРАМУВАННЯ
ЗАГАЛЬНОГО ПРИЗНАЧЕННЯ
(з дисципліни «Побудова компіляторів»)**

Звіт студента I курсу, групи ІП-51мн
спеціальності F2 Інженерія програмного забезпечення

Панченко С. В.

(Прізвище, ім'я, по батькові)

(Підпис)

Перевірив: доцент, к.т.н. Стативка Ю.І.

(Посада, науковий ступінь, прізвище та ініціали)

(Підпис)

Зміст

1 Виконання.....	3
1.1 Визначення граматички.....	3
1.2 Приклад мови.....	4
1.3 Діаграми.....	4

1 ВИКОНАННЯ

1.1 Визначення граматики

Визначимо граматику мови lab у форматі ANTLR4.

```
grammar lab;

WS: [ \t\r\n]+ -> skip;

TYPE: TYPE_INTEGER | TYPE_REAL | TYPE_STRING | TYPE_BOOL;

TYPE_INTEGER: 'int';
TYPE_REAL: 'real';
TYPE_STRING: 'string';
TYPE_BOOL: 'bool';
KEYWORD_FN: 'fn';
KEYWORD_LET: 'let';
KEYWORD_IF: 'if';
KEYWORD_WHILE: 'while';
LPAREN: '(';
RPAREN: ')';

statement
: statementVariableDeclaration
| statementIf
| statementWhile
| expressionFunctionCall
;

functionCallArgument: LITERAL | ID | expressionFunctionCall;
expressionFunctionCall: LPAREN ID functionCallArgument* RPAREN;
statementVariableDeclaration: LPAREN KEYWORD_LET ID
functionCallArgument RPAREN;
statementsList: LPAREN statement* RPAREN;
statementIf: LPAREN KEYWORD_IF expressionFunctionCall statementsList
statementsList RPAREN;
statementWhile: LPAREN KEYWORD_WHILE expressionFunctionCall
statementsList RPAREN;

funcParam: LPAREN ID TYPE RPAREN;
funcParamList: funcParam+;
funcDef: LPAREN KEYWORD_FN ID LPAREN funcParamList? RPAREN
statementsList RPAREN;

program: funcDef+;
```

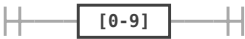
```
LITERAL: LITERAL_BOOL | LITERAL_REAL | LITERAL_INTEGER |  
LITERAL_STRING;  
LITERAL_BOOL: 'true' | 'false';  
LITERAL_REAL: DIGIT+ '.' DIGIT* | '.' DIGIT+;  
LITERAL_INTEGER: DIGIT+;  
LITERAL_STRING: ''' ( ~["\\r\\n"] | '\\ ' . )* ''';  
  
ID : [a-zA-Z_+\\-*/^<=>]+;  
fragment DIGIT : [0-9];
```

1.2 Приклад мови

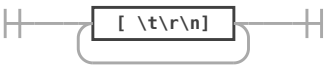
```
(fn otherfunc () (  
  (display (** 2 6))  
)  
  
(fn main ((a int) (b real) (c string)) (  
  (let d (concat c "Hello KPI"))  
  (if (<= 55 104) (  
    (display (concat d "I AM TRUE"))  
  ) (  
    (display (concat d "I AM FALSE"))  
  )  
  (let i 0)  
  (while (le i 5) (  
    (display (* 2 (- a (/ i b))))  
    (set i (+ i 1))  
  )  
  )  
)
```

1.3 Діаграми

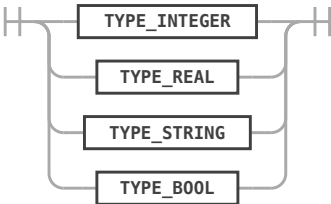
DIGIT



WS



TYPE



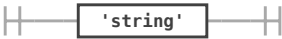
TYPE_INTEGER



TYPE_REAL



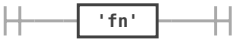
TYPE_STRING



TYPE_BOOL



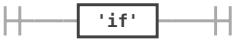
KEYWORD_FN



KEYWORD_LET



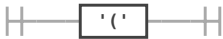
KEYWORD_IF



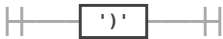
KEYWORD_WHILE



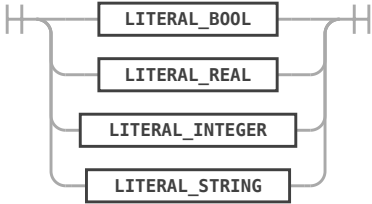
LPAREN



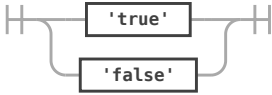
RPAREN



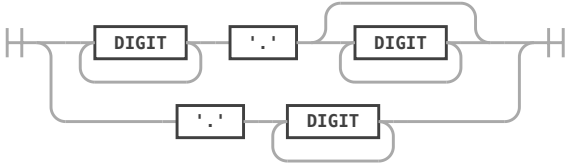
LITERAL



LITERAL_BOOL



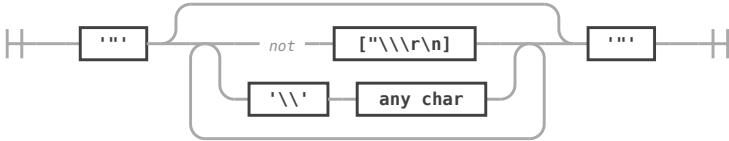
LITERAL_REAL



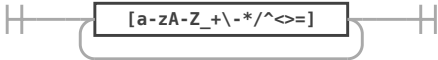
LITERAL_INTEGER



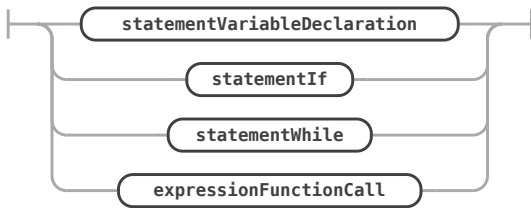
LITERAL_STRING



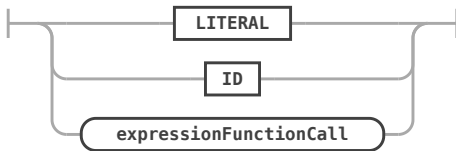
ID



statement



functionCallArgument



expressionFunctionCall



statementVariableDeclaration



statementsList



statementIf



statementWhile



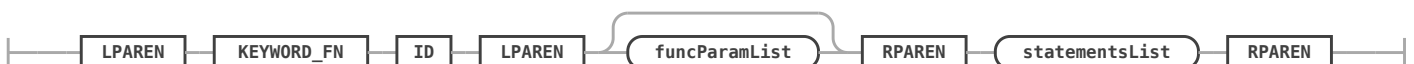
funcParam



funcParamList



funcDef



program

