**Техническое задание**

**Автор:** Шляжко Кирилл 23531/3

**Интерпретатор SMALL BASIC**

**Задача:** реализовать интерпретатор диалекта SMALL BASIC

**Минимальные требования:**

­ целочисленные переменные;

­ инфиксные выражения c “+”, “-­“, “\*”, “/”, “%”, скобками;

­ операторы сравнения “<”, “=”, “>”, “>=”, “<=”, “<>” ;

­ Операторы INPUT, PRINT, GOTO, GOSUB/RETURN, IF

Интерпретатор SMALL BASIC должен уметь анализировать, обрабатывать и выполнять исходный код программы или запроса.

**Алгоритм работы интерпретатора:**

1) прочитать вводный файл;

2) проанализировать файл на синтаксические ошибки;

3) выполнить соответствующие действия, описанные в программе;

4) если не достигнуто условие завершения программы, перейти к пункту 3, если достигнуто условия, то вывести в выходной файл результат работы программы;

Для выполнения пункта 3 необходимо будет создать синтаксический анализатор выражений, который будет разбивать исходное выражение на составные части, компоненты (неделимые элементы выражения – лексемы).

Функция, разбивающая выражение на составные части, должна решать несколько задач:

1) игнорировать пробелы и символы табуляции;

2) извлекать каждую лексему из текста;

3) определять тип лексемы.

Анализатор будет использовать пять типов лексем: разделитель, переменная, число, команда, кавычка.

* Вид командной строки: D:\>prog.exe inputFile.txt outFile.txt

* Если пользователь просто запустил “prog.exe”, программа будет печатать:   
  Use format: <executable file>.exe <input file>.txt <output file>.txt
* Пример входного файла:

INPUT a

|  |
| --- |
|  |
|  | INPUT b |
|  | c = a + b |
|  |  |
|  | IF a > b THEN |
|  | PRINT "a > b" |
|  | ENDFI |
|  |  |
|  |
|  |  |
|  | IF a < b THEN |
|  | PRINT "a < b" |
|  | ENDFI |
|  |  |
|  | IF a = b THEN |
|  | PRINT "a = b" |
|  | ENDFI |
|  |  |
|  | IF a <> b THEN |
|  | PRINT "a <> b" |
|  | ENDFI |
|  | END |

* Пример выходного файла:

a > b

a <> b

* Самые основные функции реализации:

struct resultByAnalyz analyzing(char \*program) // Анализ синтаксиса

void start(char \*program, struct infoLabels \*infoLabels, struct infoVariables \*infoVars, char \*fileResult) // Начало разбора данных на лексемы

void getToken(struct lexem \*token) // Получение следующей лексемы

int main(int argc, char \*argv[]) // Запуск интерпретатора

А также такие функции вычисления алгебраических операций, функции использования Basic-команд, и прочие функции для поддержания правильной работы интерпретатора.