Національний Технічний Університет України

«Київський Політехнічний Інститут ім. Ігоря Сікорського»

Кафедра автоматизації проектування енергетичних процесів та систем ТЕФ

Звіт до лабораторної роботи №1

з дисципліни

«Лінгвістичне забезпечення САПР»

Виконав: Круглий Д. В.

студент ТЕФ

групи ТР-61

Перевірив: Стативка Ю. І.

Київ – 2018 р.

***Завдання:***

Мова програмування повинна містити в собі оператор присвоювання, оператори введення та виведення. Оператори циклу та умовного переходу: for (<ідентифікатор> = <вираз> ; <відношення>; <ідентифікатор> = <вираз>) if <відношення> { < список операторів >}. Алгоритм лексичного аналізу: розбір до роздільника.

**Граматика мови:**

1. ***<program>::=*** program <назва> <сп. оголошень> { <сп.операторів> }

2. ***<назва>::=*** <ідентифікатор>

3. ***<ідентифікатор>::=*** <буква> | <ідентифікатор> <буква> | <ідентифікатор> <цифра>

4. ***<буква>::=*** a | b | c | … | x | y | z | A | B | C | … | X | Y | Z

5. ***<цифра>::=*** 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9

6. ***<сп.оголошень>::=*** <оголошення> ¶ | <сп. оголошень> < оголошення > ¶

7. ***<оголошення >::=*** <тип> <сп.ідентифікаторів>

8. ***<тип>::=*** integer | float | double

9. ***<сп.ідентифікаторів>::=*** <ідентифікатор>|<список ідентифікаторів>, <ідентифікатор>

10. ***<сп.операторів>::=*** <оператор>¶ | <сп.операторів> <оператор> ¶

11. <***оператор>::=*** <присвоювання> | <ввід> | <вивід> | <цикл> | <умова>

12. ***<*** ***присвоювання >::=*** <ідентифікатор> = <вираз>

13. ***<ввід>::=*** read (<сп.ідентифікаторів>)

14. ***<вивід>::=*** write(<сп.ідентифікаторів>)

15. ***<цикл>::=*** for (<ідентифікатор> = <вираз> ; <відношення>; <ідентифікатор> = <вираз>)

16. ***<умова>::=*** if <відношення> { < список операторів >}

17. ***<лог.вираз>::=*** <лог.терм> | <лог.вираз> or <лог.терм>

18. ***<лог.терм>::=*** <лог.множник> | <лог.терм> and <лог.множник>

19. ***<лог.множник>::=*** <відношення> | [<лог.вираз>] | not <лог.множник>

20. ***<відношення>::=*** <вираз> <знак выразу> <вираз>

21. ***<знак виразу>::*** = > | < | >= | <= | == | !=

22. ***<вираз>::=*** <терм> | <вираз> + <терм>| < вираз > - <терм> | - <терм>

23. ***<терм>::=*** <множник> | <терм> \* < множник > | <терм> / < множник>

24. ***<множник>::=*** <перш. вираз> | < множник > \* <перш.вираз>

25. ***<перш. вираз>::=*** (<вираз>) | <ідентифікатор> | <константа>

26. ***<константа>::=*** <цифра> | <константа> <цифра>

**Таблиця лексем:**

|  |  |
| --- | --- |
| ***Лексема*** | ***Код*** |
| Program | 1 |
| if | 2 |
| do | 3 |
| step | 4 |
| int | 5 |
| float | 6 |
| const | 7 |
| read | 8 |
| write | 9 |
| for | 10 |
| to | 11 |
| } | 12 |
| { | 13 |
| / | 14 |
| ; | 15 |
| : | 16 |
| , | 17 |
| = | 18 |
| + | 19 |
| - | 20 |
| \* | 21 |
| ( | 22 |
| ) | 23 |
| < | 24 |
| > | 25 |
| == | 26 |
| != | 27 |
| Idn | 28 |
| Con | 29 |

**Вхідна програма:**

***program Test;***

***const a = 5, b = 10;***

***int c;***

***if(a < b) {***

***c = a;***

***}***

***int i, sum = 0;***

***for(i = 0; i < 10; i = 15){***

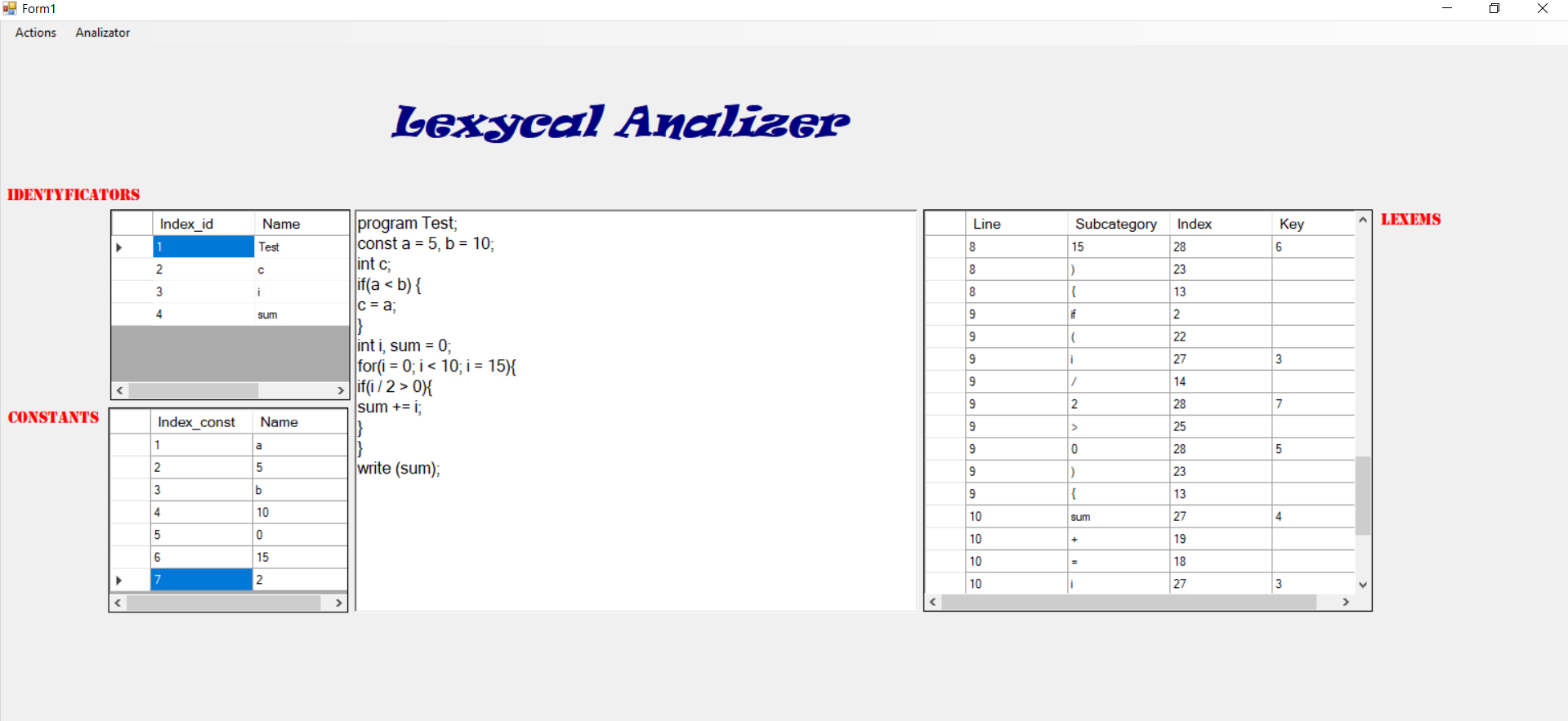
***if(i / 2 > 0){***

***sum += i;***

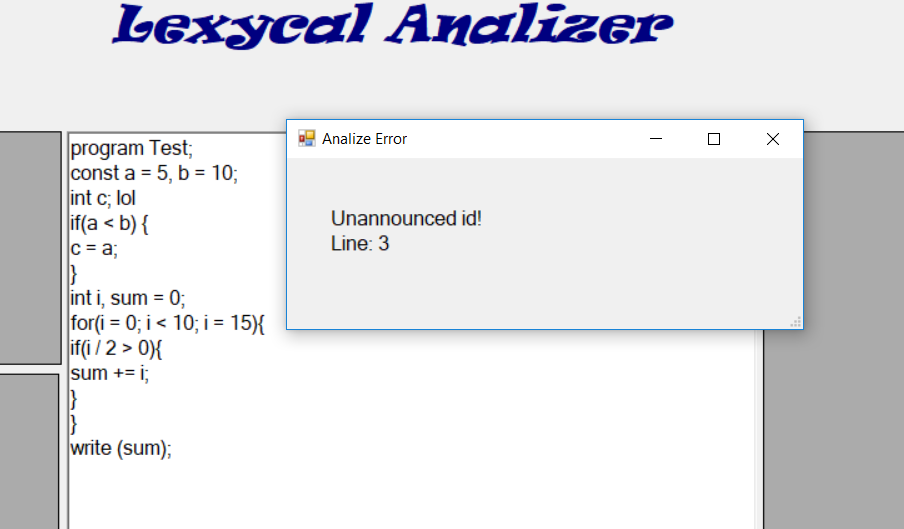
***}***

***}***

***write (sum);***

**Приклад виконання програми:**

**При виникненні помилки, лексичний аналізатор видає повідомлення про помилку:**



***Висновок:*** В результаті виконання завдання, було побудовано програму, яка виконує лексичний аналіз заданої мови. В даному випадку, було створене текстове вікно, а також три таблиці: констант, ідентифікаторів та лексем.