

Лабораторная работа №2 Синтаксический анализ

Доработать программу лексического анализатора из лабораторной работы № 1 так, чтобы генерируемый ею поток токенов поступал на вход синтаксического анализатора. Выполнить программную реализацию синтаксического анализатора. Результаты работы программы представить в виде дерева разбора.

Варианты заданий

1. Входной язык содержит арифметические выражения, разделённые символом ; (точка с запятой). Арифметические выражения состоят из идентификаторов, десятичных чисел с плавающей точкой (в обычной и экспоненциальной форме), знака присваивания (:=), знаков операций +, -, *, / и круглых скобок.
2. Входной язык содержит логические выражения, разделённые символом ; (точка с запятой). Логические выражения состоят из идентификаторов, констант **0** и **1**, знака присваивания (:=), операций **or**, **xor**, **and**, **not** и круглых скобок.
3. Входной язык содержит операторы условия **if ... then ... else** и **if ... then**, разделённые символом ; (точка с запятой). Операторы условия содержат идентификаторы, знаки сравнения <, >, =, шестнадцатеричные числа, знак присваивания (:=). Шестнадцатеричными числами считать последовательность цифр и символов **a**, **b**, **c**, **d**, **e**, **f**, начинающуюся с цифры (например, 89, 45ac, 0abc).
4. Входной язык содержит операторы цикла **for (...; ...; ...) do ...**, разделённые символом ; (точка с запятой). Операторы цикла содержат идентификаторы, знаки сравнения <, >, =, строковые константы (последовательность символов в двойных кавычках), знак присваивания (:=).
5. Входной язык содержит операторы цикла **while (...) ... done**, разделённые символом ; (точка с запятой). Операторы цикла содержат идентификаторы, знаки сравнения <, >, =, десятичные числа с плавающей точкой (в обычной и экспоненциальной форме), знак присваивания (:=).
6. Входной язык содержит операторы цикла **do ... while (...)**, разделённые символом ; (точка с запятой). Операторы цикла содержат идентификаторы, знаки сравнения <, >, =, римские числа, знак присваивания (:=). Римскими считать числа, записанные большими буквами **X**, **V** и **I**.
7. Входной язык содержит арифметические выражения, разделённые символом ; (точка с запятой). Арифметические выражения состоят из идентификаторов, римских чисел, знака присваивания (:=), знаков операций +, -, *, / и круглых скобок. Римскими считать числа, записанные большими буквами **X**, **V** и **I**.
8. Входной язык содержит логические выражения, разделённые символом ; (точка с запятой). Логические выражения состоят из идентификаторов, констант **true** и **false**, знака присваивания (:=), операций **or**, **xor**, **and**, **not** и круглых скобок.

скобок.

9. Входной язык содержит операторы условия **if ... then ... else** и **if ... then**, разделённые символом ; (точка с запятой). Операторы условия содержат идентификаторы, знаки сравнения <, >, =, десятичные числа с плавающей точкой (в обычной и экспоненциальной форме), знак присваивания (:=).
10. Входной язык содержит операторы цикла **for (...; ...; ...)** **do ...**, разделённые символом ; (точка с запятой). Операторы цикла содержат идентификаторы, знаки сравнения <, >, =, шестнадцатеричные числа, знак присваивания (:=). Шестнадцатеричными числами считать последовательность цифр и символов **a**, **b**, **c**, **d**, **e**, **f**, начинающуюся с цифры (например, 89, 45ac, 0abc).
11. Входной язык содержит операторы цикла **while (...) ... done**, разделённые символом ; (точка с запятой). Операторы условия содержат идентификаторы, знаки сравнения <, >, =, строковые константы (последовательность символов в двойных кавычках), знак присваивания (:=).
12. Входной язык содержит операторы цикла **do ... while (...)**, разделённые символом ; (точка с запятой). Операторы цикла содержат идентификаторы, знаки сравнения <=, =>, =, десятичные числа с плавающей точкой (в обычной и экспоненциальной форме), знак присваивания (:=).
13. Входной язык содержит арифметические выражения, разделённые символом ; (точка с запятой). Арифметические выражения состоят из идентификаторов, шестнадцатеричных чисел, знака присваивания (:=), знаков операций +, -, *, / и круглых скобок. Шестнадцатеричными числами считать последовательность цифр и символов **a**, **b**, **c**, **d**, **e**, **f**, начинающуюся с цифры (например, 89, 45ac, 0abc).
14. Входной язык содержит логические выражения, разделённые символом ; (точка с запятой). Логические выражения состоят из идентификаторов, символьных констант 'T' и 'F', знака присваивания (:=), операций **or**, **xor**, **and**, **not** и круглых скобок.
15. Входной язык содержит операторы условия **if ... then ... else** и **if ... then**, разделённые символом ; (точка с запятой). Операторы условия содержат идентификаторы, знаки сравнения <, >, =, римские числа, знак присваивания (:=). Римскими считать числа, записанные большими буквами X, V и I.
16. Входной язык содержит операторы цикла **for (...; ...; ...)** **do ...**, разделённые символом ; (точка с запятой). Операторы цикла содержат идентификаторы, знаки сравнения <, >, =, десятичные числа с плавающей точкой (в обычной и экспоненциальной форме), знак присваивания (:=).
17. Входной язык содержит операторы цикла **do ... while (...)**, разделённые символом ; (точка с запятой). Операторы условия содержат идентификаторы, знаки сравнения <=, =>, =, строковые константы (последовательность символов в двойных кавычках), знак присваивания (:=).

18. Входной язык содержит операторы цикла **while** (...) ... **done**, разделённые символом ; (точка с запятой). Операторы цикла содержат идентификаторы, знаки сравнения <, >, =, римские числа, знак присваивания (:=). Римскими считать числа, записанные большими буквами **X**, **V** и **I**.

19. Входной язык содержит арифметические выражения, разделённые символом ; (точка с запятой). Арифметические выражения состоят из идентификаторов, символьных констант (один символ в одинарных кавычках), знака присваивания (:=), знаков операций +, -, *, / и круглых скобок.

20. Входной язык содержит логические выражения, разделённые символом ; (точка с запятой). Логические выражения состоят из идентификаторов, шестнадцатеричных чисел, знака присваивания (:=), операций **or**, **xor**, **and**, **not** и круглых скобок. Шестнадцатеричными числами считать последовательность цифр и символов **a**, **b**, **c**, **d**, **e**, **f**, начинающуюся с цифры (например, 89, 45ac, 0abc).

21. Входной язык содержит операторы цикла **for** (...; ...; ...) **do** ..., разделённые символом ; (точка с запятой). Операторы цикла содержат идентификаторы, знаки сравнения <, >, =, римские числа, знак присваивания (:=). Римскими считать числа, записанные большими буквами **X**, **V** и **I**.

22. Входной язык содержит операторы цикла **do** ... **while** (...), разделённые символом ; (точка с запятой). Операторы цикла содержат идентификаторы, знаки сравнения <=, =>, =, шестнадцатеричные числа, знак присваивания (:=). Шестнадцатеричными числами считать последовательность цифр и символов **a**, **b**, **c**, **d**, **e**, **f**, начинающуюся с цифры (например, 89, 45ac, 0abc).

23. Входной язык содержит операторы условия **if** ... **then** ... **else** и **if** ... **then**, разделённые символом ; (точка с запятой). Операторы условия содержат идентификаторы, знаки сравнения <, >, =, строковые константы (последовательность символов в двойных кавычках), знак присваивания (:=).

24. Входной язык содержит операторы цикла **while** (...) ... **done**, разделённые символом ; (точка с запятой). Операторы цикла содержат идентификаторы, знаки сравнения <, >, =, шестнадцатеричные числа, знак присваивания (:=). Шестнадцатеричными числами считать последовательность цифр и символов **a**, **b**, **c**, **d**, **e**, **f**, начинающуюся с цифры (например, 89, 45ac, 0abc).