***Лабораторная работа 3***

***Проведение лексического анализа восходящим методом***

Задание. С использованием лексического анализатора lex разработать и реализовать программу восходящего лексического анализа.

*Документация по Lex*: http://www.cs.man.ac.uk/~pjj/cs211/flexdoc.html

Описание индивидуальных вариантов приведено в задании.

1. Разработать регулярную грамматику лексем входного языка в соответствии с заданием.
2. Для проверки рекомендуется построить ДКА, распознающий грамматику.
3. Изучить средства для лексического анализа в среде lex.
4. Разработать алгоритм синтаксического анализа для индивидуального варианта задания методом восходящего анализа.

Предусмотреть тесты для проверки распознавания всех лексем.

**Варианты заданий к лаб. работе №3**

Все варианты заданий предусматривают возможность ***наличия комментариев неограниченной длины*** во входном языке. Форма организации комментариев соответствует языку Паскаль.

Любые лексемы, не предусмотренные вариантом задания, встречающиеся в исходном тексте, должны трактоваться как ошибочные.

1. Входной язык содержит арифметические выражения, разделенные символом **;**(точка с запятой). Арифметические выражения состоят из идентификаторов, десятичных чисел с плавающей точкой, знака присваивания (**:=**), знаков операций **+**, **-**, **\***, **/** и круглых скобок.
2. Входной язык содержит логические выражения, разделенные символом **;**(точка с запятой). Логические выражения состоят из идентификаторов, констант **true** («истина») и **false** («ложь»), знака присваивания (**:=**), знаков операций **or**, **xor**, **and**, **not** и круглых скобок.
3. Входной язык содержит операторы условия типа **if** … **then** … **else** и **if** … **then**, разделенные символом **;**(точка с запятой). Операторы условия содержат идентификаторы, знаки сравнения **<**, **>**, **=**, десятичные числа с плавающей точкой, знак присваивания (**:=**).
4. Входной язык содержит операторы цикла типа **for** **(**…**;** …**;** …**)** **do**, разделенные символом **;**(точка с запятой). Операторы цикла содержат идентификаторы, знаки сравнения **<**, **>**, **=**, десятичные числа с плавающей точкой, знак присваивания (**:=**).
5. Входной язык содержит операторы выбора типа **case** … **of** … **end**, разделенные символом **;**(точка с запятой). Операторы выбора содержат идентификаторы, знак двоеточия (**:**), знаки операций **+**, **-**, десятичные числа с плавающей точкой, знак присваивания (**:=**).
6. Входной язык содержит операторы цикла типа **while** … **do** …, разделенные символом **;**(точка с запятой). Операторы цикла содержат идентификаторы, знаки сравнения **<**, **>**, **=**, знаки операций **+**, **-**, десятичные числа с плавающей точкой, знак присваивания (**:=**).
7. Входной язык содержит арифметические выражения, разделенные символом **;**(точка с запятой). Арифметические выражения состоят из идентификаторов, римских чисел, знака присваивания (**:=**), знаков операций **+**, **-**, **\***, **/** и круглых скобок.
8. Входной язык содержит логические выражения, разделенные символом **;**(точка с запятой). Логические выражения состоят из идентификаторов, констант **0** и **1**, знака присваивания (**:=**), знаков операций **or**, **xor**, **and**, **not** и круглых скобок.
9. Входной язык содержит операторы условия типа **if** … **then** … **else** и **if** … **then**, разделенные символом **;**(точка с запятой). Операторы условия содержат идентификаторы, знаки сравнения **<**, **>**, **=**, римские числа, знак присваивания (**:=**).
10. Входной язык содержит операторы цикла типа **for** **(**…**;** …**;** …**)** **do**, разделенные символом **;**(точка с запятой). Операторы цикла содержат идентификаторы, знаки сравнения **<**, **>**, **=**, римские числа, знак присваивания (**:=**).
11. Входной язык содержит операторы выбора типа **case** … **of** … **end**, разделенные символом **;**(точка с запятой). Операторы выбора содержат идентификаторы, знак двоеточия (**:**), знаки операций **+**, **-**, римские числа, знак присваивания (**:=**).
12. Входной язык содержит операторы цикла типа **while** … **do** …, разделенные символом **;**(точка с запятой). Операторы цикла содержат идентификаторы, знаки сравнения **<**, **>**, **=**, знаки операций **+**, **-**, римские числа, знак присваивания (**:=**).
13. Входной язык содержит арифметические выражения, разделенные символом **;**(точка с запятой). Арифметические выражения состоят из идентификаторов, шестнадцатеричных чисел, знака присваивания (**:=**), знаков операций **+**, **-**, **\***, **/** и круглых скобок.
14. Входной язык содержит логические выражения, разделенные символом **;**(точка с запятой). Логические выражения состоят из идентификаторов, шестнадцатеричных чисел, знака присваивания (**:=**), знаков операций **or**, **xor**, **and**, **not** и круглых скобок.
15. Входной язык содержит операторы условия типа **if** … **then** … **else** и **if** … **then**, разделенные символом **;**(точка с запятой). Операторы условия содержат идентификаторы, знаки сравнения **<**, **>**, **=**, шестнадцатеричные числа, знак присваивания (**:=**).
16. Входной язык содержит операторы цикла типа **for** **(**…**;** …**;** …**)** **do**, разделенные символом **;**(точка с запятой). Операторы цикла содержат идентификаторы, знаки сравнения **<**, **>**, **=**, шестнадцатеричные числа, знак присваивания (**:=**).
17. Входной язык содержит операторы выбора типа **case** … **of** … **end**, разделенные символом **;**(точка с запятой). Операторы выбора содержат идентификаторы, знак двоеточия (**:**), знаки операций **+**, **-**, шестнадцатеричные числа, знак присваивания (**:=**).
18. Входной язык содержит операторы цикла типа **while** … **do** …, разделенные символом **;**(точка с запятой). Операторы цикла содержат идентификаторы, знаки сравнения **<**, **>**, **=**, знаки операций **+**, **-**, шестнадцатеричные числа, знак присваивания (**:=**).
19. Входной язык содержит арифметические выражения, разделенные 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, разделенные символом **;**(точка с запятой). Логические выражения состоят из идентификаторов, символьных констант ‘T’ («истина») и ‘F’ («ложь), знака присваивания (**:=**), знаков операций **or**, **xor**, **and**, **not** и круглых скобок.
21. Входной язык содержит операторы условия типа **if** … **then** … **else** и **if** … **then**, разделенные символом **;**(точка с запятой). Операторы условия содержат идентификаторы, знаки сравнения **<**, **>**, **=**, строковые константы (последовательность символов в двойных кавычках), знак присваивания (**:=**).
22. Входной язык содержит операторы цикла типа **for** **(**…**;** …**;** …**)** **do**, разделенные символом **;**(точка с запятой). Операторы цикла содержат идентификаторы, знаки сравнения **<**, **>**, **=**, строковые константы (последовательность символов в двойных кавычках), знак присваивания (**:=**).
23. Входной язык содержит операторы выбора типа **case** … **of** … **end**, разделенные символом **;**(точка с запятой). Операторы выбора содержат идентификаторы, знак двоеточия (**:**), знаки операций **+**, **-**, символьные константы (один символ в одинарных кавычках), знак присваивания (**:=**).
24. Входной язык содержит операторы цикла типа **while** … **do** …, разделенные символом **;**(точка с запятой). Операторы цикла содержат идентификаторы, знаки сравнения **<**, **>**, **=**, знаки операций **+**, **-**, символьные константы (один символ в одинарных кавычках), знак присваивания (**:=**).

|  |  |
| --- | --- |
| 25 | Входной язык содержит операторы цикла типа **for** **(**…**;** …**;** …**)** **do**, разделенные символом **;**(точка с запятой). Операторы цикла содержат идентификаторы, знаки сравнения **<**, **>**, **=**, строковые константы (последовательность символов в двойных кавычках), знак присваивания (**:=**). |
| 26 | Входной язык содержит операторы выбора типа **case** … **of** … **end**, разделенные символом **;**(точка с запятой). Операторы выбора содержат идентификаторы, знак двоеточия (**:**), знаки операций **+**, –, символьные константы (один символ в одинарных кавычках), знак присваивания (**:=**). |
| 27 | Входной язык содержит операторы цикла типа **while** … **do** …, разделенные символом **;**(точка с запятой). Операторы цикла содержат идентификаторы, знаки сравнения **<**, **>**, **=**, знаки операций **+**, –, символьные константы (один символ в одинарных кавычках), знак присваивания (**:=**). |
| 28 | Входной язык содержит операторы цикла типа **repeat** … **until** …, разделенные символом **;**(точка с запятой). Операторы цикла содержат идентификаторы, знаки сравнения **<**, **>**, **=**, знаки операций **+**, –, символьные константы (один символ в одинарных кавычках), знак присваивания (**:=**). |
| 29 | Входной язык содержит арифметические выражения, разделенные символом **;**(точка с запятой). Арифметические выражения состоят из идентификаторов, десятичных чисел с фиксированной точкой, знака присваивания (**:=**), знаков операций **+**, –, **\***, **/** и круглых скобок. |
| 30 | Входной язык содержит операторы условия типа **if** … **then** … **else** и **if** … **then**, разделенные символом **;**(точка с запятой). Операторы условия содержат идентификаторы, знаки сравнения **<**, **>**, **=**, десятичные числа с фиксированной точкой, знак присваивания (**:=**). |

*Примечания:*

1. Римскими числами считать последовательности больших латинских букв **X**, **V** и **I**. Если в задании требуется использование римских чисел, то обычные (арабские) числа должны восприниматься как недопустимые лексемы.
2. Шестнадцатеричными числами считать последовательность цифр и символов ‘a’, ‘b’, ‘c’,’d’, ’e’ и ‘f’, начинающуюся с цифры (например: 56, 45ac9, 0abc4). Допускается использовать специальные символы для выделения шестнадцатеричных чисел (например: $0abc4 или 0x0abc4).

В отчёте по работе должно быть представлено:

* Краткое изложение цели работы.
* Индивидуальный вариант задания
* Грамматика языка для индивидуального варианта в форме БНФ
* (в случае создания) Построенный ДКА, распознающий грамматику, и только её
* Набор тестов для проверки распознавания всех лексем
* Описание используемого программного обеспечения (либо библиотеки), а также описание всех входных данных, необходимых для построения сканера в соответствии с заданием
* Пример анализируемого входного текста и результат работы лексического анализатора.
* Разработанный код анализатора
* Выводы по проделанной работе.