Министерство науки и высшего образования Российской Федерации
Калужский филиал
федерального государственного бюджетного
образовательного учреждения высшего образования
«Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)»
(КФ МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ *ИУК «Информатика и управление»*

КАФЕДРА *ИУК5 «Системы обработки информации и управления»*

ЛАБОРАТОРНАЯ РАБОТА №2

«Разработка синтаксического анализатора»

ДИСЦИПЛИНА: «Конструирование компиляторов»

Выполнил: студент гр. ИУК5-11М _____(_Стародуб Я.Н._)
(Подпись) (Ф.И.О.)

Проверил: _____(_Потапов А.Е._)
(Подпись) (Ф.И.О.)

Дата сдачи (защиты):

Результаты сдачи (защиты):

- Балльная оценка:
- Оценка:

Калуга, 2024

Выполнение лабораторной работы

Цели:

- Получить практические навыки синтеза синтаксических анализаторов.
- Разработать программу синтаксического анализа входного файла лексем с проверкой его на соответствие БНФ.

Порядок выполнения работы:

- Ознакомиться разделами «LL(1)-грамматики», «Построение синтаксического графа», «Построение программы грамматического разбора для заданного синтаксиса».
- По варианту задания построить синтаксический граф для реализуемого языка.
- Составить контрольные примеры на реализуемом языке. Хотя бы один пример должен проверять поведение вашей программы при наличии синтаксических ошибок в контрольном примере.
- Запрограммировать и отладить программу, производящую синтаксический анализ реализуемого языка.
- Выполнить тестирование на контрольных примерах. При этом пример пропускается через программу лексического анализа, а файл с лексемами является входным для программы синтаксического анализа. При необходимости доработать модуль сканирования. Лабораторная работа считается выполненной, если программа выдает правильные и понятные сообщения о синтаксических ошибках с указанием строк, где эта ошибка имеет место.
- Оформить отчет.

Ход работы

1. Построение синтаксического графа:

Синтаксический граф стоится на основе БНФ.

Вариант 3

```
<Программа> ::= <Объявление переменных> <Описание вычислений>
<Описание вычислений> ::= [<Список присваиваний>]
<Объявление переменных> ::= Var <Список переменных>;
<Список переменных> ::= <Идент>|<Идент>, <Список переменных>
<Список присваиваний>::= <Присваивание> |
                         <Присваивание> <Список присваиваний>
<Присваивание> ::= <Идент> = <Выражение>;
<Выражение> ::= <Ун.оп.> <Подвыражение>|<Подвыражение>
<Подвыражение> :: = ( <Выражение> ) | <Операнд>|
                     <Подвыражение> <Бин.оп.> <Подвыражение>
<Ун.оп.> ::="-"
<Бин.оп.> ::="-" | "+" | "*" | "/"
<Операнд> ::= <Идент><Константа>
<Идент> ::= <Буква><Идент>|<Буква>
<Константа> ::= <Цифра><Константа>|<Цифра>
```

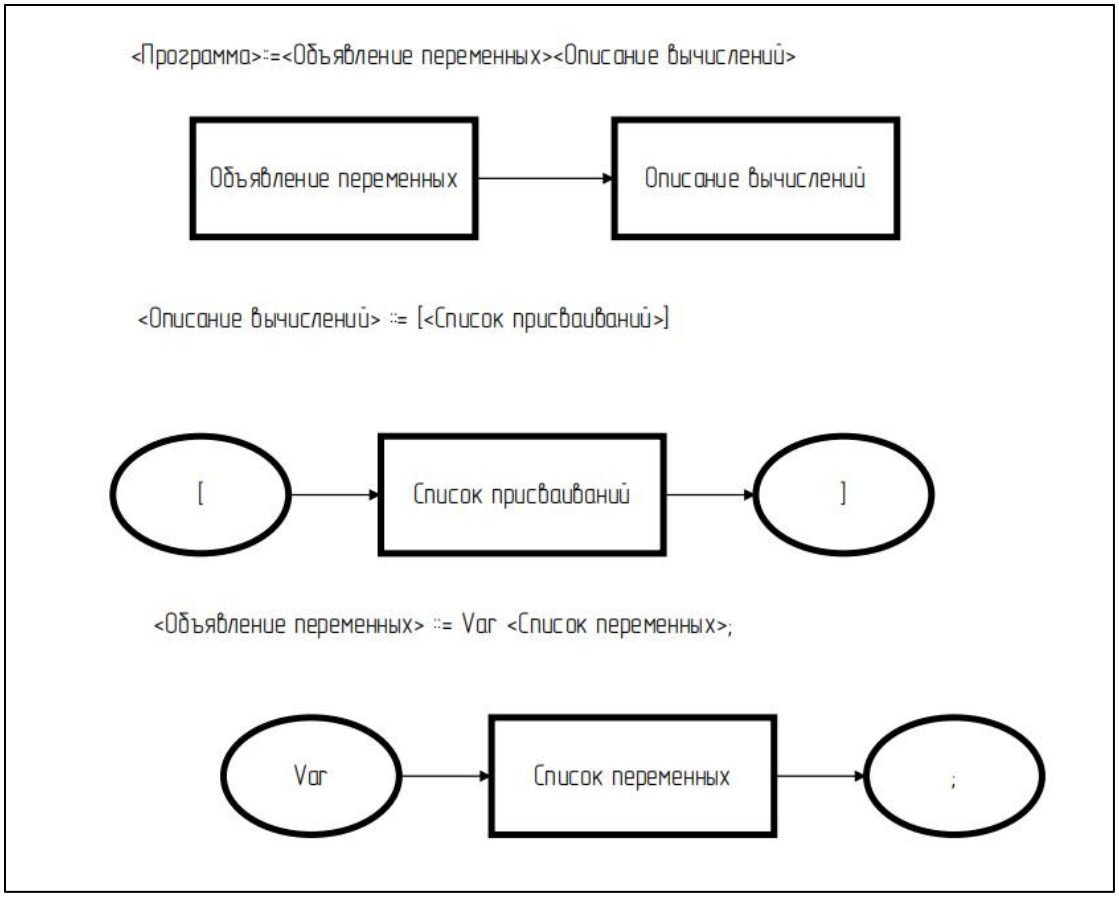

Рис 1. Синтаксический граф 1

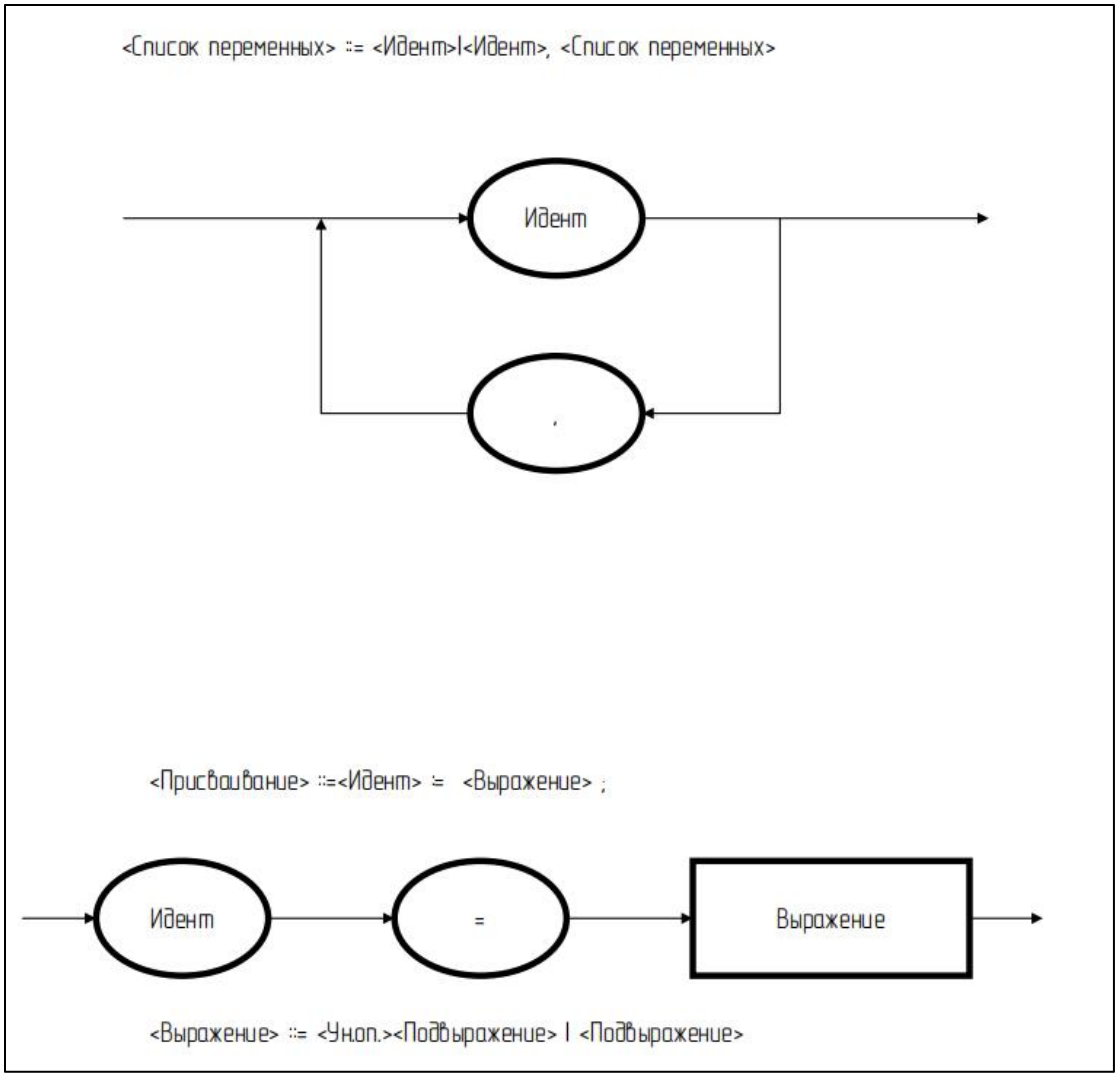

Рисунок 2. Синтаксический граф 2

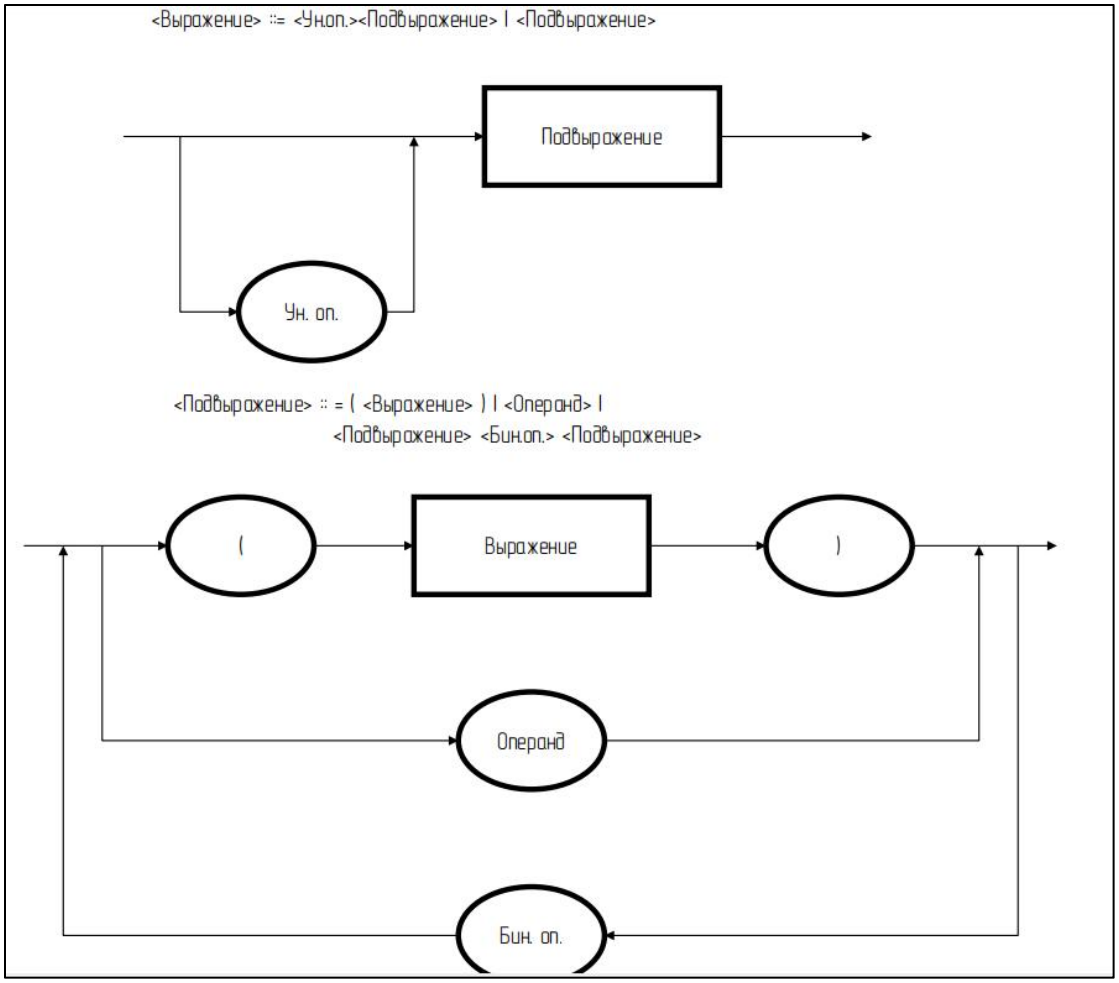

Рисунок 3. Синтаксический граф 3

2. Контрольные примеры:

- Пример 1: Программа с правильным синтаксисом.

```
Var x, y;
[
    x = 10;
    y = x + 5;
]
```

- Здесь синтаксический анализатор должен успешно обработать объявление переменных, операторы присваивания и выражения.

- Пример 2: Программа с ошибкой синтаксиса.

```
Var x, y;
[   x = 10
```

y = x +;]

- В этом примере ожидаются ошибки: первая ошибка — отсутствие точки с запятой после оператора присваивания, вторая ошибка — неправильное выражение с оператором ‘+’ в конце.

3. Программирование синтаксического анализатора:

- В коде использована рекурсивная грамматика для разбора программы.

- ‘program()’: основная функция, которая начинает синтаксический анализ. Она проверяет наличие ключевого слова ‘Var’ или открывающей квадратной скобки ‘[‘ в начале программы и проверяет, что программа заканчивается закрывающей квадратной скобкой ‘]’.

- ‘statement()’: проверка отдельных операторов, таких как объявления переменных и присваивания.

- ‘variableDeclaration()’: разбор объявления переменных, начиная с ключевого слова ‘Var’.

- ‘assignment()’: разбор оператора присваивания.

- ‘expression()’, ‘term()’, ‘factor()’: функции для обработки арифметических выражений, термов и факторов (например, чисел, идентификаторов, скобок).

- Каждая функция выбрасывает исключение `SyntaxException` в случае синтаксической ошибки с указанием строки и типа ошибки.

4. Обработка ошибок:

Ошибки синтаксиса включают:

- Ожидание точки с запятой после операторов.

- Отсутствие закрывающей скобки.

- Нарушения правил синтаксиса, например, два оператора подряд без выражения между ними или использование неинициализированных переменных.

- В случае обнаружения синтаксической ошибки, программа сообщает о ней с указанием строки, где она была обнаружена.

5. Тестирование:

- Программа проходит тестирование с различными примерами:

- Пример с правильным синтаксисом должен быть успешно проанализирован без ошибок.

- Пример с синтаксической ошибкой должен привести к выбрасыванию исключения с описанием ошибки.

6. Пример обработки ошибки:

- Пример с ошибкой:

```
Var x, y;
x = 10
y = x +;
```

- В этом примере:

- Пропущена точка с запятой после оператора присваивания ‘x = 10’, что приводит к выбрасыванию исключения ‘SyntaxException’ с сообщением "Ожидалась точка с запятой, найдено: 10 на строке X".

- Во второй строке есть ошибка с выражением ‘x +;’, где отсутствует второй операнд для оператора ‘+’, что также вызывает исключение с сообщением о синтаксической ошибке.

7. Использование лексического анализатора:

- Для синтаксического анализа лексемы проходят через программу лексического анализа, которая генерирует токены. Эти токены передаются в синтаксический анализатор, который проверяет соответствие синтаксической грамматике языка.

- В случае ошибок в лексемах, лексический анализатор выдает исключения, и программа синтаксического анализа должна корректно обработать эти ошибки.

В результате выполнения задания синтаксический анализатор должен правильно распознавать структуру программы и выдавать четкие и понятные сообщения об ошибках с указанием строк.

Пример работы программы

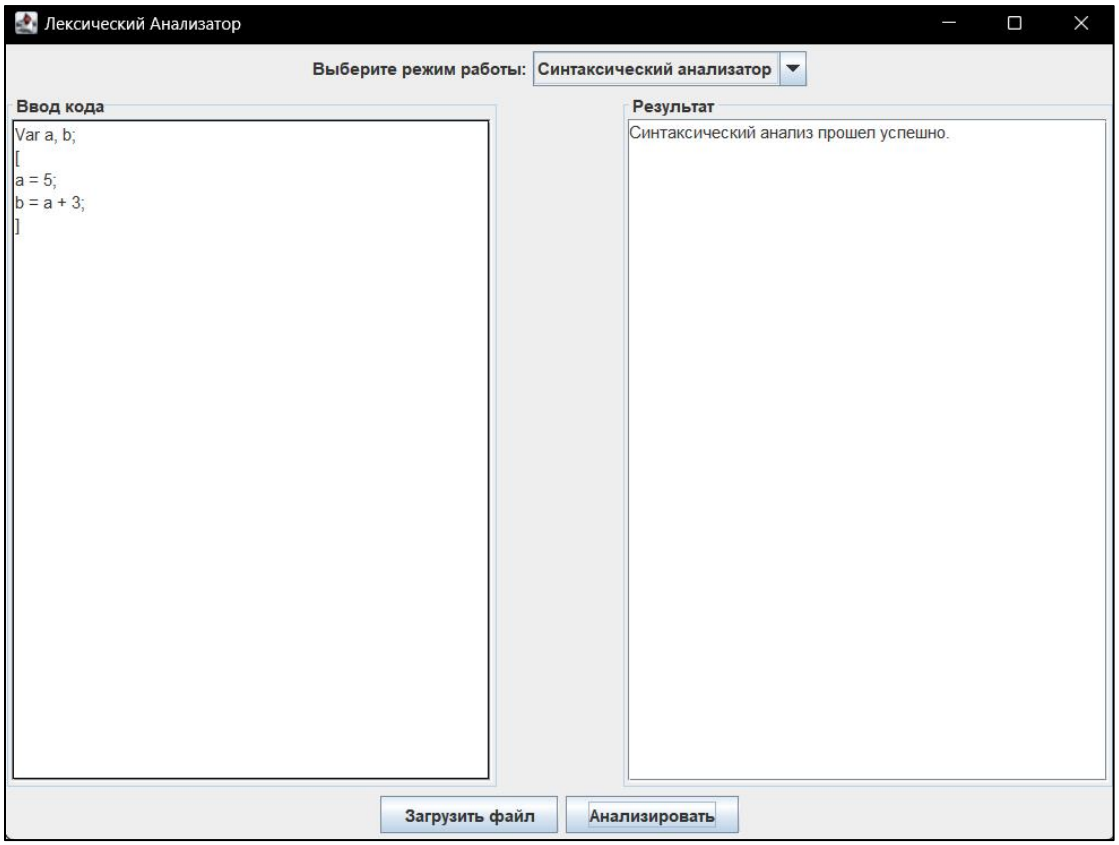

Рисунок 4. Работа программы с корректными данными

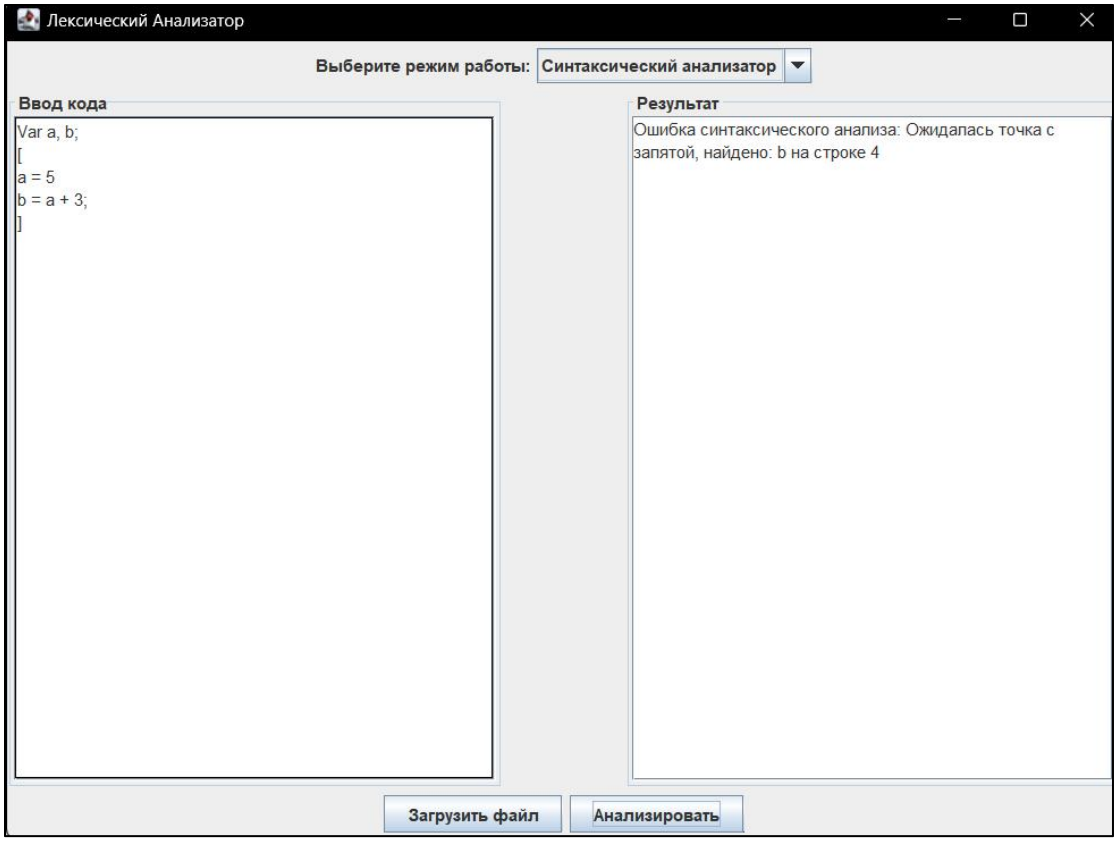

Рисунок 5. Работа программы с некорректными данными

6. Отладка и исправление ошибок:

- После реализации модуля, проведено тестирование на различных примерах, включая примеры с ошибками, чтобы убедиться в корректной обработке всех типов лексем и правильной генерации ошибок.

Вывод

В ходе выполнения лабораторной работы были получены практические навыки синтеза синтаксического анализатора, а также разработана программа синтаксического анализатора входного файла лексем с проверкой его на соответсвие БНФ.