



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 №1

«Разработка лексического 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:

Выполнение лабораторной работы

Цели:

- Получить практические навыки синтеза лексического анализаторов.
- Разработать программу лексического анализа входного файла программы с разбиением его на лексемы и записи их в выходной файл

Порядок выполнения работы:

- Ознакомиться с разделом «Построение лексического анализатора» .
- По варианту задания определить, какие классы лексем будут в языке.
- Составить контрольные примеры на реализуемом языке. Хотя бы один пример должен проверять поведение вашей программы при наличии недопустимых символов в транслируемом файле.
- Запрограммировать и отладить модуль сканирования. Выполнить тестирование на контрольных примерах. Результатом работы должна быть таблица, содержащая лексемы и признаки их классов. Необходимо включить в результирующий файл информацию о номерах строк исходного текста транслируемой программы.
- Оформить отчет.

Ход работы

1. Определение классов лексем:

- В языке программирования, для выполнения лексического анализа, необходимо выделить основные классы лексем, такие как:

- Идентификаторы: переменные, функции и другие имена, состоящие из букв и цифр, начинающиеся с буквы.

- Константы: числовые значения.

- Операторы: символы арифметических операций (например, '+', '-', '*', '/') и другие операторы (например, '=').

- Ключевые слова: такие как 'Var'.

- Разделители: запятая (','), точка с запятой (';'), скобки ('()', '[]').

- Прочие символы: например, пробелы, которые игнорируются, или символы, не относящиеся к допустимым лексемам, для которых будет выброшена ошибка.

2. Составление контрольных примеров:

- Пример 1: Проверка корректного кода, состоящего только из допустимых лексем.

```
Var x = 10;
```

```
x = x + 5;
```

- Здесь проверяется правильность идентификации переменных, оператора присваивания, числа и оператора.

- Пример 2: Пример с недопустимым символом.

```
Var x = 10;
```

```
x = @ + 5;
```

- В этом примере символ '@' является недопустимым, и программа должна выбросить ошибку.

3. Разработка модуля сканирования:

- Модуль сканирования был реализован с использованием регулярных выражений для каждого типа лексем:

- Регулярные выражения позволяют выделить идентификаторы, константы, операторы, ключевые слова и разделители.

- В ходе обработки исходного кода, каждый символ анализируется, и на основе его значения добавляется соответствующая лексема в список токенов.

4. Обработка ошибок:

- В случае нахождения недопустимого символа, программа выбрасывает исключение с описанием ошибки, включая позицию и строку, на которой был найден некорректный символ.

5. Тестирование и вывод лексем:

- После выполнения лексического анализа, программа генерирует список токенов с их типами и значениями. Каждый токен также содержит информацию о номере строки, на которой он был найден.

Пример работы программы

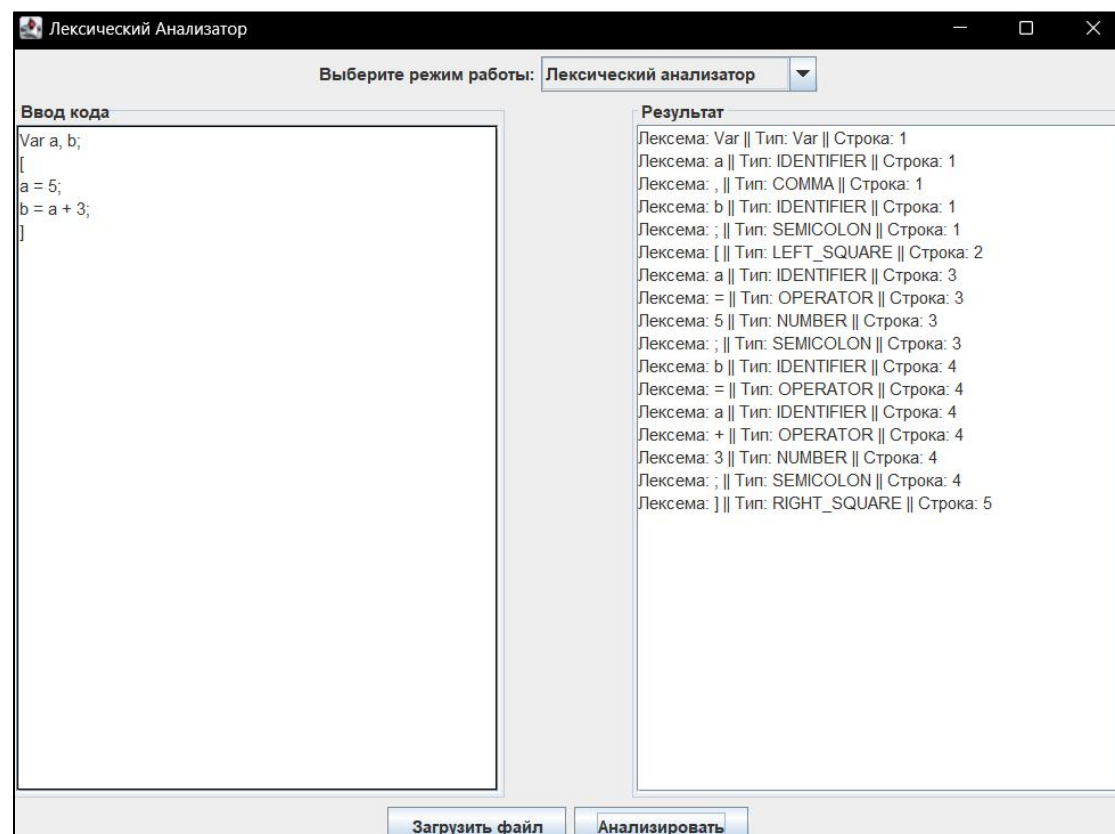


Рисунок 1. Работа программы с корректными данными

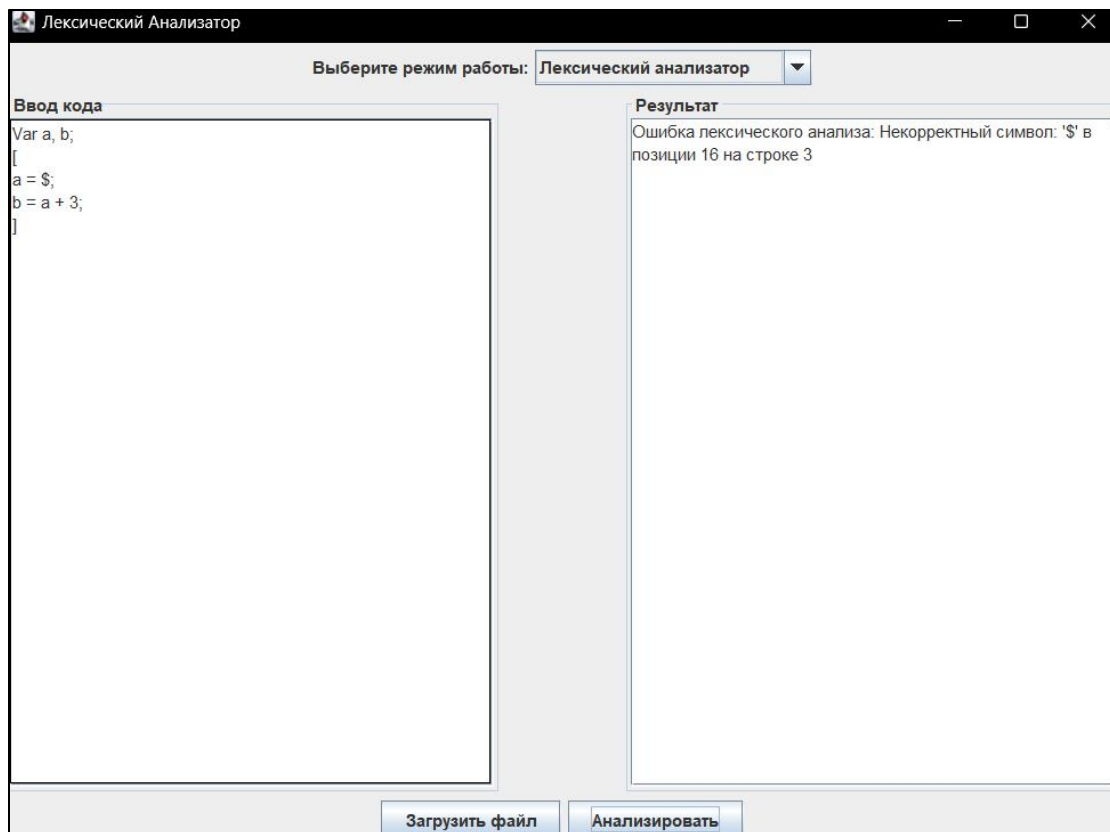


Рисунок 2. Работа программы с некорректными данными

6. Отладка и исправление ошибок:

- После реализации модуля, проведено тестирование на различных примерах, включая такие 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 лексического анализатора, а также разработана программа лексического анализатора входного файла с разбиением его на лексемы и запись результата в выходной файл