



Министерство науки и высшего образования Российской Федерации

**Федеральное государственное бюджетное образовательное учреждение высшего образования
«МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ имени Н.Э.БАУМАНА
(национальный исследовательский университет)»**

Факультет: Информатика и системы управления

Кафедра: Теоретическая информатика и компьютерные технологии

Лабораторная работа № 4

Объектно-ориентированный лексический анализатор

Вариант 7

Работу выполнил
студент группы ИУ9-62
Чугунов Д. С.

Москва, 2019

Цель работы

Целью данной работы является приобретение навыка реализации лексического анализатора на объектно-ориентированном языке без применения каких-либо средств автоматизации решения задачи лексического анализа.

Задание

В лабораторной работе предлагается реализовать две первые фазы стадии анализа: чтение входного потока и лексический анализ. При этом следует придерживаться схемы реализации объектно-ориентированного лексического анализатора, рассмотренной на лекции.

Входной поток должен загружаться из файла (в UTF-8). В результате работы программы в стандартный поток вывода должны выдаваться описания распознанных лексем в формате

Тег (координаты фрагмента): атрибут лексемы

При этом для лексем, не имеющих атрибутов, нужно выводить только тег и координаты.

Лексемы во входном файле могут разделяться пробельными символами (пробел, горизонтальная табуляция, маркеры окончания строки), а могут быть записаны слитно (если это не приводит к противоречиям).

Лексический анализатор должен иметь программный интерфейс для взаимодействия с парсером. Рекомендуется реализовывать его как метод *nextToken()* для императивных языков или функцию, возвращающую список лексем, для функциональных языков.

Входной файл может содержать ошибки, при обнаружении которых лексический анализатор должен выдавать сообщение с указанием координаты, восстанавливаться и продолжать работу.

Индивидуальный вариант

Идентификаторы: последовательности латинских букв и десятичных цифр, оканчивающиеся на цифру. Числовые литералы: непустые последовательности десятичных цифр, ограниченные знаками «<» и «>». Операции: «<=», «==», «>».

Реализация

На листинге 1 представлен объектно-ориентированный лексический анализатор, реализованный на языке Python 3.

```
from copy import copy
from abc import ABC, abstractmethod
from functools import total_ordering

...<пропущено>...

if __name__ == '__main__':
    main()
```

Листинг 1 – объектно-ориентированный лексический анализатор

Тестирование

Для входной последовательности, представленной на листинге 2, лексический анализатор выдал последовательность лексем, представленных на листинге 3.

```
abc45
abc8abc2
abc2abc2
<12>
<12> <==== <456> abdf5 sd5
```

Листинг 2 – входные данные

```
ID (1, 1)-(1, 6): abc45
ID (2, 1)-(2, 9): abc8abc2
ID (3, 1)-(3, 9): abc2abc2
NUM (4, 1)-(4, 5): 12
NUM (5, 1)-(5, 5): 12
OP (5, 6)-(5, 8): <=
OP (5, 8)-(5, 10): ==
OP (5, 10)-(5, 11): =
NUM (5, 12)-(5, 17): 456
ID (5, 18)-(5, 23): abdf5
ID (5, 24)-(5, 27): sd5
END (5, 27)-(5, 27):
```

Листинг 3 – выходные данные

Вывод

В ходе лабораторной работы были приобретены навыки написания объектно-ориентированных лексических анализаторов, а также был разработан лексический анализатор, способный разобрать 3 типа лексем и восстанавливаться в случае лексической ошибки.