Лабораторная работа № 1.3 «Объектно-ориентированный лексический анализатор»

Скоробогатов С.Ю., Коновалов А.В.

22 марта 2016

1 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.

2 Задание

В лабораторной работе предлагается реализовать на объектно-ориентированном языке (Java, C++, Python и т.д.) 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. Например,

```
IDENT (1, 2)-(1, 4): str
ASSIGN (1, 8)-(1, 9):
STRING (1, 11)-(1, 16): qwerty
```

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).

Идентификаторы и числовые литералы не могут содержать внутри себя пробельных символов, если в задании явно не указано иного. Комментарии, строковые и символьные литералы могут содержать внутри себя пробельные символы.

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.

Для лексических доменов должны вычисляться их атрибуты (если указанный лексический домен есть в варианте задания):

- для целых чисел атрибут должен быть целым числом наибольшей разрядности (например, в Java — `long`).
- для вещественных чисел атрибут должен быть вещественным числом (например, `double` в Java или C++),
- для идентификаторов — номер в таблице идентификаторов (см. слайды лекции),
- для строковых констант — значение, изображаемое самой строковой константой (т.е. без окружающих кавычек и с интерпретацией escape-последовательностей, если они есть в варианте задания),
- для комментариев токен не порождается, вместо этого координаты комментария помещается в список комментариев (см. слайды лекции).

3 Индивидуальный вариант

Комментарии: начинаются с «`//`» и продолжаются до окончания строки текста. Идентификаторы: любой текст, не содержащий «`/`» и ограниченный символами «`/`». Ключевые слова: «`/while/`», «`/do/`», «`/end/`».

4 Отчёт по лабораторной работе № 1.3

Отчёт выполняется в разметке Markdown по следующему шаблону:

````
% Лабораторная работа № 1.3 «Объектно-ориентированный
  лексический анализатор»
% 21 марта 2023 г.
% Вася Пупкин, ИУ9-63Б
# Цель работы
‹переписываете цель работы из задания›
# Индивидуальный вариант
‹переписываете индивидуальный вариант›
# Реализация
```java
…
```
# Тестирование
Входные данные
```
…
```
Вывод на `stdout`
```
…
```
# Вывод
‹пишете, чему научились›
````

В отчёте приведён лишь необходимый минимум.

Шаблон отчёта

Ваш отчёт будет конвертирован в PDF при помощи pandoc следующей командой:

```
pandoc \
  --pdf-engine=xelatex \
  -V 'mainfont:Liberation Serif' \
  -V 'monofont:Liberation Mono' \
  "$SOURCE" -o "$PDF"
```

Язык реализации: Markdown

Код решения

Из файла

Отправить