

Санкт-Петербургский политехнический университет Петра Великого  
Кафедра компьютерных систем и программных технологий

**Отчёт по лабораторной работе**

**Дисциплина:** Транслирующие системы

**Тема:** Программа для распознавания химических формул на языке *lex*

Выполнил студент гр. 43501/3  
Преподаватель

Мальцев М.С.  
Цыган В.Н.

Санкт-Петербург  
22 марта 2019 г.

## 1 Задание

Составить LEX-программу для следующего перевода. Пусть для записи химических формул используются следующие восемь элементов

**H, C, N, O, SI, S, CL, SN.**

Элементы в формулах разделяются запятыми. Элементы могут появляться в любом порядке и в любых сочетаниях. Для указания количества атомов в формуле используется цифр, записанная вслед за обозначением химического элемента. Формулы не обязательно представляют реально существующие соединения. Несколько примеров записи формул:

**H2, O**

**O, H7**

**SN, S, O4**

Таким образом имеется девять входных символов:

**C H I L N O S,  $\mu$**

где  $\mu$  - обозначение цифры.

Необходимо осуществлять распознавание допустимых (в рамках описанных выше правил) химических формул и вычислять молекулярный вес вещества, описанной формулой.

Входные предложения, которые не являются допустимыми должны печататься с соответствующим диагностическими сообщениями.

Предусмотреть не менее двух разных диагностических сообщений.

## 2 Ход работы

### 2.1 Применяемый шаблон

Для определения указанных химических элементов, был разработан шаблон:

$H[2 - 9]?(", "|\backslash n")$

Вместо  $H$  подставляются остальные химические элементы.

По условию, за буквенным именованием элемента может идти его номер. Номер - это цифра, но так как в данном случае у нас не может идти ни цифра 1(т.к. она в таком случае она не несет никакой информации), ни цифра 0(т.к. это означает отсутствие элемента, что в рамках логики приложения бессмысленно), то диапазон выбран от 2 до 9.

Заканчиваться конструкция может двумя способами: либо запятой и пробелом, что предполагает следующий элемент в формуле, либо символом окончания строки, что говорит об окончании формулы.

### 2.2 Разработанные функции

Для более удобной обработки текста была разработана функция:

*void handler(int, intmassOfOne)*

Её задача определение молекулярной массы элемента и формирование значения молекулярной массы формулы. Также в задачи решаемые функцией входит возвращения последнего элемента в поток обработки, в случае, если этот элемент - это символ перевода строки. Это сделано для того, чтобы обработчик конца формулы смог зафиксировать её и вывести результат вычисления молекулярной массы либо диагностическое сообщение.

Первый параметр *lenOfElement* обозначает длину элемента в символах. Например, для элемента *H* он будет равен 1, а для *CL* - 2.

Второй параметр *massOfOne* обозначает молекулярную массу элемента. Для элемента *H* он будет равен 1, а для *CL* - 35.

## 2.3 Вывод результата и диагностическое сообщение

Возможно два типа сообщений: сообщение о том, что формула обработалась корректно, содержащее значение молекулярной массы для формулы, либо сообщение о некорректности введенной формулы. В первом случае, вывод выглядит следующим образом:

*H | 1 mass*

Во втором, помимо сообщения об ошибке, справа от некорректного символа выводится восклицательный знак и выводится краткое сообщение, почему она произошла. Программа может сообщить о трёх ошибках:

*H5!H | Error - correct syntax, but incorrect lexis !!!*

*!!o!! !CL2 | Error - incorrect syntax !!!*

*O0! | Error - incorrect number !!!*

## 2.4 Текст программы

```
1  %{
2  void handler(int lenOfElement, int massOfOne);
3  %}
4
5      int mass = 0;
6      int err = 0;
7
8  %%
9  H[2-9]?(", "|"\n") handler(1, 1);
10 C[2-9]?(", "|"\n") handler(1, 12);
11 N[2-9]?(", "|"\n") handler(1, 14);
12 O[2-9]?(", "|"\n") handler(1, 16);
13 SI[2-9]?(", "|"\n") handler(2, 28);
14 S[2-9]?(", "|"\n") handler(1, 32);
15 CL[2-9]?(", "|"\n") handler(2, 35);
16 SN[2-9]?(", "|"\n") handler(2, 118);
17
18 \n {
19     if (err == 0) {
20         printf(" | %d mass \n", mass);
```

```

21     }
22     else if (err == 3) {
23         printf(" | Error - correct syntax, but incorrect lexis !!! \n");
24     }
25     else if (err == 2) {
26         printf(" | Error - incorrect number !!! \n");
27     }
28     else if (err == 1) {
29         printf(" | Error - incorrect syntax !!! \n");
30     }
31     err = 0;
32     mass=0;
33 }
34
35
36 (H|C|N|O|SI|S|CL|SN)[2-9]? {
37     ECHO;
38     printf("!");
39     err = 3;
40 }
41
42 (H|C|N|O|SI|S|CL|SN)[0-1]? {
43     ECHO;
44     printf("!");
45     err = 2;
46 }
47
48 . {
49     ECHO;
50     printf("!");
51     err = 1;
52 }
53
54 %%
55
56 void handler(int lenOfElement, int massOfOne) {
57     if (yytext[lenOfElement] > 49 && yytext[lenOfElement] < 58) {
58         mass += massOfOne * (yytext[lenOfElement] - 48 );
59     } else {
60         mass += massOfOne;
61     }
62     if (yytext[yyleng-1] == 10) {
63         for (int i=0; i<yyleng-1; i++) {
64             printf("%c",yytext[i]);
65         }
66         yless(yyleng-1);
67     } else {
68         ECHO;
69     }
70 }
71
72 #ifndef yywrap
73 int yywrap() { return 1; }
74 #endif
75
76 main () { while (yylex()); }

```

## 2.5 Тестирование программы

Тестовые входные данные были сформированы таким образом, чтобы максимально полно продемонстрировать функциональность программы.

Входные данные:

```
1 H2, O
2 O, H7
3 SN, S, O4
4 H, H5, H9, HH
5 H5H
6 H
7 H7
8 SN2
9 lol CL2
10 gfhdfgjd dfsd
11 H9, O1
12 O0
13 H9, O2, SN3, CL, SI4
14 C4
15 S7, C4, N3
16 SN, CL, SI
```

Результат работы программы:

```
1 H2, O | 18 mass
2 O, H7 | 23 mass
3 SN, S, O4 | 214 mass
4 H, H5, H9, H!H | Error – correct syntax, but incorrect lexis !!!
5 H5!H | Error – correct syntax, but incorrect lexis !!!
6 H | 1 mass
7 H7 | 7 mass
8 SN2 | 236 mass
9 l!o!l! !CL2 | Error – incorrect syntax !!!
10 g!f!h!d!f!g!j!d! !d!f!s!d! | Error – incorrect syntax !!!
11 H9, O1! | Error – incorrect number !!!
12 O0! | Error – incorrect number !!!
13 H9, O2, SN3, CL, SI4 | 542 mass
14 C4 | 48 mass
15 S7, C4, N3 | 314 mass
16 SN, CL, SI | 181 mass
```

Полученные результаты соответствуют требованиям, указанным в задании. В связи с этим, можно считать, что программа успешно прошла тест.

## 3 Вывод

В ходе работы была написана программа, которая проверяет корректность введенных химических формул, а в случае корректности вычисляет молекулярную массу формулы. Программа успешно справилась с тестовой задачей.

В результате работы закрепились навыки работы с программой для генерации лексических анализаторов *Lex*. Получен опыт в разработке и отладке приложений основывающихся на этой программе.