



МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение  
высшего образования

«МИРЭА – Российский технологический университет»

РТУ МИРЭА

---

Институт Информационных Технологий

Кафедра Вычислительной техники

---

**ОТЧЕТ О ВЫПОЛНЕНИИ ПРАКТИЧЕСКОЙ  
РАБОТЫ №5**

«Простой анализатор JSON-файла с использованием ПО  
Flex»

**по дисциплине**

«Теория формальных языков»

Выполнил студент группы ИКБО-04-22

*Основин А.И.*

Принял старший преподаватель

*Боронников А.С.*

Практическая работа  
выполнена

«\_\_»\_\_\_\_\_2023 г.

«Зачтено»

«\_\_»\_\_\_\_\_2023 г.

Москва 2023

# СОДЕРЖАНИЕ

1	ПОСТАНОВКА ЗАДАЧИ .....	3
1.1	Условия задачи .....	3
1.2	Метод получения данных.....	3
2	РЕАЛИЗАЦИЯ ПРОГРАММЫ .....	4
3	РЕЗУЛЬТАТЫ ТЕСТИРОВАНИЯ .....	5
4	ВЫВОД.....	6

# 1 ПОСТАНОВКА ЗАДАЧИ

## 1.1 Условия задачи

С помощью ПО FLEX реализовать простой анализатор JSON-файла. Обработка несуществующей лексемы: вывести лексему без соотношения к определенному классу токенов.

Лексемы:

- Символы
  - BEGIN\_OBJECT ( { );
  - END\_OBJECT ( } );
  - BEGIN\_ARRAY ( [ );
  - END\_ARRAY ( ] );
  - COMMA ( , );
  - COLON ( : );
- Литералы
  - LITERAL (true, false, null);
- Строки
  - STRING ( “string” );
- Числа
  - NUMBER ( 1, -1, +1, 1e1000 ).

## 1.2 Метод получения данных

Ввод в программу осуществляется из файла, название которого передано в качестве аргумента при вызове из командной строки.

## 2 РЕАЛИЗАЦИЯ ПРОГРАММЫ

В Листинге 1 представлена реализация программы-анализатора JSON-файла.

*Листинг 1 – Код программы на языке программирования Flex*

```
%option noyywrap yylineno
%{
    #include <stdio.h>
    #include <string.h>
    int ch;
}%
literal (true|false|null)
string ((\"[^\"]*"|'[^']*'))
digit [0-9]
number ([-+]?({digit}*\.({digit}+|({digit}+\.({digit}+)([eE]
+)?({digit}+)?[fLFL]?))
ws [ \t\n]+
%%
\{                                {printf("(BEGIN_OBJECT, %s)\n", yytext); ch +=
yylen; }
\}                                {printf("(END_OBJECT, %s)\n", yytext); ch += yylen; }
\[                                {printf("(BEGIN_ARRAY, %s)\n", yytext); ch += yylen; }
\]                                {printf("(END_ARRAY, %s)\n", yytext); ch += yylen; }
,                                {printf("(COMMA, %s)\n", yytext); ch += yylen; }
:                                {printf("(COLON, %s)\n", yytext); ch += yylen; }

{literal}                        {printf("(LITERAL: %s)\n", yytext); ch += yylen; }
{string}                        {printf("(STRING, %s)\n", yytext); ch += yylen; }
{number}                        {printf("(NUMBER, %s)\n", yytext); ch += yylen; }
{ws}                            {ch += yylen; }
[^a-zA-Z0-9 \. \" ' \+ \- \n \t] {printf("(UNKNOWN CHARACTER, %s)\n", yytext); ch
+= yylen; }
%%
int main(int argc, char **argv) {
    if(argc < 2) {
        printf("\nNot enough arguments. Please specify filename.\n");
        return -1;
    }

    if((yyin = fopen(argv[1], "r")) == NULL) {
        printf("\nCannot open file %s.\n", argv[1]);
        return -1;
    }

    ch = 1;
    yylineno = 1;
    printf("Tokenlist:\n");
    yylex();
    fclose(yyin);
    return 0;
}
```

### 3 РЕЗУЛЬТАТЫ ТЕСТИРОВАНИЯ

В Таблице 1 представлены результаты тестирования программы-анализатора JSON-файла.

Таблица 1 – Тестирование программы-анализатора JSON-файла

Ввод	Вывод программы	Ожидаемый вывод
{ "name": "Ivan", "age": 37, "children": [ "Masha", "Igor" ], "married": true }	Tokenlist: (BEGIN_OBJECT, '{') (STRING, 'name') (COLON, ':') (STRING, 'Ivan') (COMMA, ',') (STRING, 'age') (COLON, ':') (NUMBER, 37) (COMMA, ',') (STRING, 'children') (COLON, ':') (BEGIN_ARRAY, '[') (STRING, 'Masha') (COMMA, ',') (STRING, 'Igor') (END_ARRAY, ']') (COMMA, ',') (STRING, 'married') (COLON, ':') (LITERAL, 'true') (END_OBJECT, '}')	Tokenlist: (BEGIN_OBJECT, '{') (STRING, 'name') (COLON, ':') (STRING, 'Ivan') (COMMA, ',') (STRING, 'age') (COLON, ':') (NUMBER, 37) (COMMA, ',') (STRING, 'children') (COLON, ':') (BEGIN_ARRAY, '[') (STRING, 'Masha') (COMMA, ',') (STRING, 'Igor') (END_ARRAY, ']') (COMMA, ',') (STRING, 'married') (COLON, ':') (LITERAL, 'true') (END_OBJECT, '}')

## 4 ВЫВОД

В ходе выполнения данной практической работы был изучен программный инструмент Flex, который позволяет определить лексический анализатор с помощью регулярных выражений для описания шаблонов токенов; была разработана программа-анализатор JSON-файла, разбивающая JSON-файл на лексемы. Программа успешно прошла тестирование, следовательно, реализация корректна.