

**МИНОБРНАУКИ РОССИИ**  
**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**  
**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**  
**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**  
**Кафедра МО ЭВМ**

**ОТЧЕТ**  
**по лабораторной работе №1**  
**по дисциплине «Алгоритмы и структуры данных»**  
**Тема: Рекурсивные функции**

Студентка гр. 9382

\_\_\_\_\_

Пя С.

Преподаватель

\_\_\_\_\_

Фирсов М.А.

Санкт-Петербург

2020

### **Цель работы.**

Реализовать синтаксический анализатор для понятия скобки, используя Си++. Изучить методы реализации с помощью рекурсии.

### **Задание.**

15. Построить синтаксический анализатор для понятия *скобки*.

*скобки* ::= A | A ( *ряд\_скобок* )

*ряд\_скобок* ::= *скобки* | *скобки* ; *ряд\_скобок*

### **Выполнение работы.**

В этой работе была написана программа, анализирующая, является ли входящее выражение скобками. После каждого считывания символа выражение проверяется на соответствие с понятием скобок. В данной программе ключевыми символами являются «()A;» - наличие или их отсутствие определяет правильность выполнения определения выражением. Для большинства случаев были обработаны и написаны ошибки, понятные для пользователя. Ввод значений происходит из файла, который открывается в главной функции main().

В программу входит функция void error(int k);, которая отвечает за вывод сообщения об определенной ошибке. Также реализована функция, являющаяся рекурсивной:

Сигнатура: bool group\_of\_brackets(std::fstream& fin, int& count); //fin – предоставляет доступ к файлу, count – считает количество одиночных скобок.

bool g; //хранит значение, отвечающее за правильность соблюдения необходимых условий.

char c; //принимает значение очередного символа.

Возвращаемое значение: bool g – возвращаемое значение.

В ней проверяются условия, определяющие скобки. В функции main() происходит посимвольное считывание выражения, которое сопровождается посимвольным выводом. Если обрабатывается недопустимое значение, то вызывается Error(int k). flag соответственно, являясь bool-типом, выполняла схожую с g функцию.

### Тестирование.

Номер теста	Выражение	Вывод	Комментарий
1	A (A;A)	A (A;A) This expression is brackets!	Показано правильное выполнение программы
2	A (A (A) ;A)	A (A (A) ;A) This expression is brackets!	Показано правильное выполнение программы
3	A (A;A (A) )	A (A;A (A) ) This expression is brackets!	Показано правильное выполнение программы
4	A	A This expression is brackets!	Показано правильное выполнение программы
5	a (A;A)	a Invalid first character! This expression is not brackets!	Показана нулевая ошибка
6	A (A ( ) ;A)	A (A ( ) Empty brackets! This expression is not brackets!	Показана четвертая ошибка
7	A (A;A (A)	A (A;A (A) Not enough bracket! This expression is not brackets!	Показана вторая ошибка
8	A (A;Aa ( ) )	A (A;Aa	Показана третья

		Invalid character in recursion! This expression is not brackets!	ошибка в случае с недопустимым символом
9	A (A;A) )	A (A;A) ) Needless bracket! This expression is not brackets!	Показана пятая ошибка в случае со скобкой

### Пример работы программы.

Входные данные	Выходные данные
A(A;A(A;A))	В main: A( В group_of_brackets(1): A(A; В group_of_brackets(2): A(A;A( Из group_of_brackets(2): A(A;A( В group_of_brackets(1): A(A;A(A; В group_of_brackets(2): A(A;A(A;A) Из group_of_brackets(2): A(A;A(A;A) В group_of_brackets(1): A(A;A(A;A))

### Выводы.

В ходе проделанной работы была изучена работа с рекурсивными функциями в Си++, выполнена реализация синтаксического анализатора для понятия скобки.

## ПРИЛОЖЕНИЕ А

### ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: main.cpp

```
#include <iostream>
#include <fstream>
// #include <locale>

/* brackets = скобки, group_of_brackets(gob) = ряд_скобок
   скобки::=A|A(ряд_скобок)
   ряд_скобок::=скобки|скобки;ряд_скобок */

void error (int k) { // функция вывода ошибок
    switch (k) {
        case 0:
            std::cout << "Invalid first character!" << "\n";
            break;
        case 1:
            std::cout << "Invalid second character!" << "\n";
            break;
        case 2:
            std::cout << "Not enough bracket!" << "\n";
            break;
        case 3:
            std::cout << "Invalid character in recursion!" << "\n";
            break;
        case 4:
            std::cout << "Empty brackets!" << "\n";
            break;
        case 5:
            std::cout << "Needless bracket!" << "\n";
    }
}

bool group_of_brackets(std::fstream& fin, int& count) { // рекурсивная функция
    определения скобок
    bool g = true; // переменная, определяющая, является ли выражение скобками
    char c = fin.get();
    std::cout << c;
    if (c != 'A') {
        std::cout << "\n";
        if (c == ')')
            error(4);
        else
            error(3);
        fin.seekg(0, std::ios::end);
        count = -2;
        return false;
    }
    c = fin.get();
    std::cout << c;
    if (c == '(') { // использование рекурсии
        count++;
        g = g && group_of_brackets(fin, count);
        c = fin.get();
        if (c != EOF)
            std::cout << c;
    }
    if (c == ';') { // использование рекурсии
        g = g && group_of_brackets(fin, count);
        c = fin.get();
    }
}
```

```

        if (c != EOF)
            std::cout << c;
    }
    if (c == '(') {//проверка правильности работы алгоритма
        count--;
        g = true;
    } else if (c == EOF && count == 0)//проверка наличия пар для каждой скобки
        g = true;
    else if (c == EOF && count > 0) {
        std::cout << "\n";
        error(2);
        g = false;
        count = -2;
    } else if (c != EOF) {
        std::cout << "\n";
        error(3);
        count = -2;
        g = false;
        fin.seekg(0, std::ios::end);
    }
    if (count == -1) {
        g = false;
        std::cout << "\n";
        error(5);
        count--;
    }
    return g;
}

int main() {
    // setlocale(LC_CTYPE, ""); //корректное использование кириллицы
    std::fstream fin("9.txt"); //получение входных данных
    bool flag = false;//переменная, определяющая, является ли выражение скобками
    int count = 1;//счетчик скобок
    char c;
    if (!fin.is_open())
        std::cout << "File is not opened!\n";
    else {
        c = fin.get();
        if (c != EOF)
            std::cout << c;
        if (c != 'A') {
            std::cout << "\n";
            error(0);
            flag = false;
        } else if ((c = fin.get()) == EOF)
            flag = true;
        else if (c != '(') {
            std::cout << "\n";
            error(1);
        } else //начало использования рекурсивной функции
            std::cout << c;
            flag = group_of_brackets(fin, count);
        }
    }
    if (flag)//ВЫВОД ОТВЕТА
        std::cout << "\n" << "This expression is brackets!" << "\n";
    else
        std::cout << "This expression is not brackets!" << "\n";
    fin.close(); //закрытие файла
    return 0;
}

```