

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

ОТЧЕТ
по лабораторной работе №1
по дисциплине «АиСД»
ТЕМА: РЕКУРСИЯ

Студент гр. 9384

Звега А.Р.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2020

Цель работы.

При освоении этого раздела студент должен познакомиться с основными понятиями и приемами рекурсивного программирования, получить навыки программирования рекурсивных процедур и функций на языке программирования C++.

Задание.

Построить синтаксический анализатор для понятия простое выражение.

простое_выражение::=простой_идентификатор |
(простое_выражение знак_операции простое_выражение)
простой_идентификатор::= буква
знак_операции::= - | + | *

Выполнение работы.

В ходе выполнения работы было реализованы 4 функции: 3 рекурсивные функции **simple()**, **abc()**, **oper()** и функция по обработке ошибок **err()**. Функции вызывают друг друга пока не закончится выражение или не встретится исключение. Функция **abc()** вызывается когда следующий символ должен быть буквой(**a+b**) или открывающей скобкой(**a+(b+c)**), а функция **oper()** когда следующий символ операция(**a+b**) или закрывающая скобка(**((a+b))**), или если выражение заканчивается буквой(**a+b_**). Функция **simple()** вызывается в начале обработки и в конце обработки, или после закрывающей скобки(**((a+b)+c)**).

Входные данные	Выходные данные
((a+(c+d))+(a+n))	((a+(c+d))+(a+n)) Все верно.

a	a Все верно.
((d+c)+c+d)	((d+c)+c+ Лишний символ. ^ Не правильное выражение.

Разработанный программный код см. в приложении А.

Выводы.

Получен практический навык работы с рекурсией и обработкой ошибок.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: code.cpp

```
#include <iostream>
#include <fstream>
#include <ctype.h>

using namespace std;

bool simple(ifstream &file, char s, int chek, bool start);
bool abc(ifstream &file, char s, int chek);
bool oper(ifstream &file, char s, int chek, bool op);
bool err(int i);

int main()
{
    ifstream file("test.txt");

    if (!file)
        cout << "Файл не открыт." << endl;

    if (simple(file, '(', 0, 1))
        cout << "\nВсе верно" << endl;
    else
        cout << "\nНе правильное выражение" << endl;

    return 0;
}

bool err(int i)
{
    switch (i)
    {
        case 0:
            cout << " Лишний символ";
            break;
        case 1:
            cout << " Неправильно поставленны скобки";
            break;
        case 2:
            cout << " После оператора нет буквы";
            break;
        case 3:
            cout << " Нет операторов";
            break;
        case 4:
            cout << "Пустая строка";
            break;

        default:
            break;
    }
    return 0;
}
```

```

bool simple(ifstream &file, char prev, int chek, bool start)
{
    char c;
    if (file >> c)
    {
        cout << c;
        if ((c == '('))
        {
            return abc(file, c, ++chek);
        }
        else if (((c == '+') || (c == '-') || (c == '*')))
            return err(0);
        else if (prev == ')')
            return simple(file, c, --chek, 0);
        else if (isalpha(c) && file >> c)
            return err(0);
        else
            return 1;
    }
    else if (file.eof())
        if (start)
            return err(4);
        else if (chek == 0)
            return 1;
        else
            return err(1);
    return err(0);
}

bool abc(ifstream &file, char prev, int chek)
{
    char c;
    if (file >> c)
    {
        cout << c;
        if (isalpha(c))
            return oper(file, prev, chek, 1);
        else if (c == '(')
            return abc(file, c, ++chek);
    }
    else if (file.eof())
        return err(2);
    return err(0);
}

bool oper(ifstream &file, char prev, int chek, bool op)
{
    char c;
    if (file >> c)
    {
        cout << c;
        if (((c == '+') || (c == '-') || (c == '*')) && (prev != '+')
&& (prev != '-') && (prev != '*'))
            return abc(file, c, chek);
        else if (c == ')')
            return simple(file, c, --chek, 0);
        else

```

```
        return err(0);
    }
    else if (file.eof())
        if (!op)
            return err(3);
        else if (chek == 0)
            return 1;
        else
            return err(1);
    return err(0);
}
```