

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

ОТЧЕТ
по лабораторной работе №1
по дисциплине «Алгоритмы и структуры данных»
Тема: Рекурсия

Студент гр. 8381

Муковский Д.В.

Преподаватель

Жангиров Т.Р.

Санкт-Петербург

2019

Цель работы

Познакомиться с основными понятиями рекурсивного программирования, научиться реализовывать рекурсивные процедуры на языке программирования C++.

Задание

Построить синтаксический анализатор для понятия *скобки*.

$скобки := A \mid скобка\ скобки$

$скобка := (B\ скобки)$

Основные теоретические положения

Любая функция в своем теле может вызывать сама себя. Рекурсия — это такой способ определения функции, при котором результат возврата из функции для данного значения аргумента определяется на основе результата возврата из той же функции для предыдущего (меньшего или большего) значения аргумента.

Если функция вызывает сам себя, то такой вызов называется рекурсивным вызовом функции. При каждом рекурсивном вызове запоминаются предыдущие значения внутренних локальных переменных и полученных параметров функции. Чтобы следующий шаг рекурсивного вывода отличался от предыдущего, значение как-минимум одного из параметров функции должно быть изменено. Остановка процесса рекурсивных вызовов функции происходит, когда изменяемый параметр достиг некоторого конечного значения, например, обработан последний элемент в массиве.

Рекурсивное обращение к функции может быть осуществлено, если алгоритм определен рекурсивно.

Чтобы циклический процесс преобразовать в рекурсивный, нужно уметь определить (выделить) три важных момента:

- условие прекращения последовательности рекурсивных вызовов функции

- формулу следующего элемента или итератора, который используется в рекурсивном процессе
 - список параметров, которые передаются в рекурсивную функцию.
- Один из параметров обязательно есть итератор (счетчик) изменяющий свое значение.

Выполнение работы

Написание работы производилось на базе операционной системы Windows 10 в среде разработки Visual Studio. Сборка и отладка, а также тестирование производилось там же.

1. Программа считывает строку одним из 3 способов(по желанию пользователя): с помощью аргументов `argv` и `argc`, из файла или из консоли.

2. После чего строка передается в функцию `brackets`(см. Приложение А), принимающую 3 аргумента: саму строку, счетчик вхождений и индекс текущего элемента. Она рекурсивно проверяет правильность заданной строки, а именно ее соответствие понятию *скобки*.

3. В случае нахождения в строке символа «(», программа вызывает функцию *bracket*, которая рекурсивно проверяет соответствие части строки понятию *скобка*.

4. В случае корректности введенных данных функция *bracket* возвращает *true*, иначе *false*.

5. В зависимости от вернувшихся значений программа выводит результат в консоль, а также в файл *output.txt*.

Пример работы

Пример строки	Результат
(B (BA) A) A	<зашел в brackets> <зашел в bracket> <зашел в brackets> <зашел в bracket> <зашел в brackets> <зашел в brackets> <зашел в brackets>

	(B(BA)A)A – это определено скобки Количество заходов в рекурсию: 7
(B(B(BA)(BA)A)A)A	<зашел в brackets> <зашел в bracket> <зашел в brackets> <зашел в bracket> <зашел в brackets> <зашел в bracket> <зашел в brackets> <зашел в brackets> <зашел в bracket> <зашел в brackets> <зашел в brackets> <зашел в brackets> <зашел в brackets> (B(B(BA)(BA)A)A)A – это определено скобки Количество заходов в рекурсию: 13

Оценка эффективности алгоритма

Сложность по времени не превышает линейную зависимость, то есть оценивается как $O(n)$.

Выводы

В ходе выполнения лабораторной работы был изучен такой вид программ, как синтаксические анализаторы. А также была реализована программа, которая анализирует строку рекурсивным методом.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

```
#include <iostream>
#include <cstring>
#include <fstream>
#include <windows.h>
#include "conio.h"
#include <fstream>

#define MAX_SIZE 100

using namespace std;

bool bracket(char* arr,int*i,int*numb);

bool brackets(char* arr, int* i, int* numb) {
    cout << "<зашел в brackets>\n";
    (*numb)++;
    if (arr[(*i)] == 'A') return true;
    if (arr[(*i)] == '('){
        (*i)++;
        if (bracket(arr,i,numb) == true)
            if (brackets(arr, i,numb) == true)
                return true;
    }
    return false;
}

bool bracket(char* arr, int* i, int*numb){
    cout << "<зашел в bracket>\n";
    (*numb)++;
    if (arr[(*i)] == 'B') {
        (*i)++;
        if (brackets(arr, i,numb) == true) {
            (*i)++;
            if (arr[(*i)] == ')') {
                (*i)++;
                return true;
            }
        }
    }
    return false;
}

char menu() {
    cout << "Введите '1' в случае считывания из файла\nВведите '2' в случае считывания из консоли\n";
    char key = _getch();
    return key;
}

int check(char* arr) {
    if (!(arr[0] == 'A' || arr[0] == '(')) {
        cout << "Неверные данные\n";
        delete[] arr;
    }
}
```

```

        return 1;
    }
    return 0;
}

int reading_data(char** arr, char key) {
    int ch;
    if (!(key == '1' || key == '2')) {
        cout << key << "Ошибка введенного ключа\n";
        return 0;
    }
    if (key == '1') {
        ifstream fs("input.txt", ios::in | ios::binary);
        if (!fs) return 0;
        //Считывание из файла
        fs.istream::getline(*arr, MAX_SIZE - 1, '\n');
        cout << *arr << endl;
        if (ch = check(*arr)) return 0;
        fs.close();
    }
    if (key == '2') {
        cout << "Введите скобки\n";
        fgets(*arr, MAX_SIZE, stdin); //Считывание из
        //консоли
        (*arr)[strlen(*arr) - 1] = '\0';
        if (ch = check(*arr)) return 0;
    }
    return 1;
}

void entry_results_in_file(char* ptr0_arr, int numb, int flag) {
    std::ofstream out;
    out.open("output.txt");
    if (out.is_open()) {
        if (flag) {
            out << ptr0_arr << " - это определено скобки \n";
            out << "Количество заходов в рекурсию: " << numb;
        }
        //Вывод информации в файл
        if (!flag) {
            out << "Это не скобки\nЧасть прошедшая проверку:" <<
            ptr0_arr << "\n";
            out << "Количество заходов в рекурсию: " << numb;
        }
    }
    return;
}

int main(int argc, char* argv[]) {
    setlocale(LC_ALL, "Russian");
    char* arr;
    char* ptr0_arr;
    arr = new char[MAX_SIZE];
    if (argc > 1) {
        strcpy_s(arr, MAX_SIZE, argv[1]);
        cout << arr;
    }
    else {
        char key = menu();
    }
}

```

```

        int ret = reading_data(&arr, key);
        if (!ret) return 0;
    }
    ptr0_arr = arr;
    int flag = 0;
    int i = 0, numb = 0;
    bool answer = brackets(arr, &i,&numb);
    if (answer && arr[i + 1] != '\0') {
        arr[i+1] = '\0';
        cout << "Это не скобки\nЧасть прошедшая проверку:" << ptr0_arr
<< "\n";
        cout << "Количество заходов в рекурсию: " << numb;
        entry_results_in_file(ptr0_arr,numb,flag);
        delete[] ptr0_arr;
        return 0;
    }
    if (answer) {
        flag = 1;
        cout << ptr0_arr << " - это определенно скобки \n";
        cout << "Количество заходов в рекурсию: " <<numb;
        entry_results_in_file(ptr0_arr, numb,flag);
    }
    else {
        arr[i+1] = '\0';
        cout << "Это не скобки\nЧасть прошедшая проверку:" << ptr0_arr
<< "\n";
        cout << "Количество заходов в рекурсию: " <<numb;
        entry_results_in_file(ptr0_arr, numb,flag);
    }
    delete[] ptr0_arr;
    return 0;
}

```