

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: Рекурсия.

Студент гр. 7383

МЕДВЕДЕВ И. С.

Преподаватель

РАЗМОЧАЕВА Н. В.

Санкт-Петербург

2018

Содержание

1. Цель работы	3
2. Реализация задачи	4
3. Тестирование	6
4. Вывод.....	7
5. Приложения	8
5.1. Приложение А: Код программы	8
5.2. Приложение Б: Тестовые случаи	12

1. Цель работы.

Познакомиться с основными понятиями и приемами рекурсивного программирования, получить навыки программирования рекурсивных процедур и функций на языке программирования C++.

Формулировка задачи: Требуется написать синтаксический анализатор понятия скобки, где скобки определяются как:

Скобки: $= A \mid (B \text{ скобки скобки })$.

2. Реализация задачи.

В данной работе было написано несколько функций для реализации задачи. Перечень функций:

`void Error (short k)` — функция, которая сообщает о той или иной ошибке. В функцию передается код ошибки.

`bool bracket (ifstream &infile, char ch)` — функция для проверки скобок. В данную функцию передается объект класса `ifstream` для работы с файлом, в котором введены скобки и переменную типа `char` для посимвольного считывания файла. Данная функция использует переменную *forCheck* типа *bool* для проверки правильности введенных скобок. В начале программа сравнивает *char ch* с символом 'А'. Встретив символ 'А', функция возвращает значение `true`. Иначе функция сравнивает символ со '('. Если же данный символ оказался открывающей скобкой, то функция считывает следующий символ с файла и сравнивает его с 'В', затем считывает следующий символ и запускает саму же себя. Не встретив '(' или 'В', функция вызывает функцию *Error (short k)*. Так же функция проверяет поданную строку на наличие закрывающих скобочек.

`int main ()` — головная функция. В данной функции используются переменные *exit* и *check* типа *bool* для выхода из цикла *while* и для проверки правильности скобок соответственно. Значение *exit* изначально `true`. Так же используется переменная *char ch* и массив `arr [100]` типа *char*. Переменная используется для посимвольного считывания с файла, а массив для считывания скобок из входного потока. Переменная *int forSwitch* используется для оператора ветвления *switch*. Указатель на файл *FILE* fp* используется для создания файла (если это требуется) и записи в него скобок, вводимых пользователем. В начале функция заходит в цикл *while* и считывает значение переменной *forSwitch*, затем, в зависимости от значения данной переменной, запускает тот или иной алгоритм.

Если значение равно '1', то программа считывает первый символ из файла test.txt и если этот символ открывающая скобка или 'А', то запускается функция *bool bracket (ifstream &infile, char ch)*. Если файл пуст или считанный символ не является открывающей скобкой или 'А', то функция сообщает об ошибке. После завершения работы функции *bool bracket*, в зависимости от вернувшегося значения, функция либо выводит строку "ЭТО СКОБКИ", либо нет.

Если значение равно '2', то функция создает файл test1.txt (если его нет) и записывает в него значение, вводимое пользователем, а затем запускается алгоритм описанный выше при *forSwitch = 1*.

Если значение равно '0', то функция выходит из цикла *while* и заканчивает свою работу.

В других случаях функция выводит строку "НЕВЕРНЫЙ ВВОД".

Код описанных выше функций представлен в Приложении А.

Рассмотрим конкретный пример работы программы. Пусть на вход подается строка: (B(BAA)A). На рис. 1 можно увидеть, когда запускается программа (глубина рекурсии увеличивается), а когда возвращается значение true (глубина рекурсии уменьшается). Синей стрелочкой показан запуск программы, красной – рекурсивный запуск, зеленой – завершение программы со значением true.

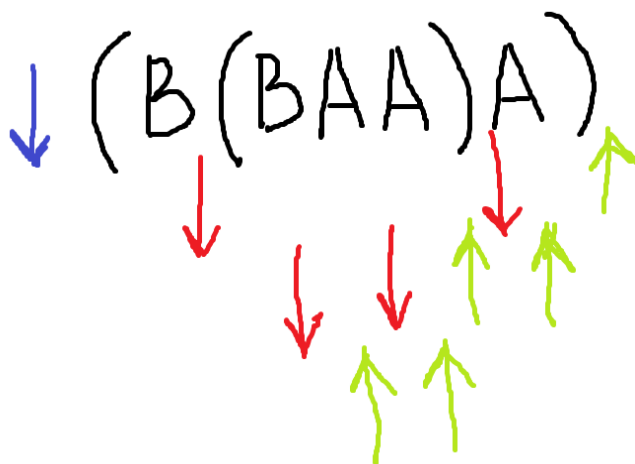


Рисунок 1 — пример работы программы.

3. Тестирование.

Программа собрана в операционной системе Ubuntu 17.04 с использованием компилятора g++. В других ОС и компиляторах тестирование не проводилось. Результаты тестирования показали, что поставленная цель выполнена. Результаты тестирования представлены в Приложении Б.

4. Вывод.

В ходе выполнения лабораторной работы были изучены основные понятия и приемы рекурсивного программирования, получены навыки программирования рекурсивных процедур и функций на языке программирования C++. Также был написан синтаксический анализатор скобок.

5. Приложения.

5.1. Приложение А: Код программы.

```
#include <iostream>
#include <fstream>
#include <cstdio>
#include <cstring>
using namespace std ;
bool bracket (ifstream &infile, char ch);
void Error (short k);

int main ( ){
    cout<<"ВАС ПРИВЕТСТВУЕТ АНАЛИЗАТОР СКОБОК! ДЛЯ ТОГО ЧТОБЫ
СЧИТАТЬ СКОБКИ С ФАЙЛА test.txt НАЖМИТЕ 1, ДЛЯ ВВОДА С КЛАВИАТУРЫ И
ЗАПИСИ В ФАЙЛ test1.txt НАЖМИТЕ 2, ДЛЯ ВЫХОДА НАЖМИТЕ 0"<<endl;
    bool exit = true, check;
    char ch;
    char arr[100];
    int forSwitch;
    FILE* fp;
    while (exit){
        cin>>forSwitch;
        switch (forSwitch){
            check = false;
            case 1:{
                ifstream infile ("test.txt");
                if (infile >> ch){
                    cout << ch;
                    if ((ch == 'A') || (ch == '('))
                        check = bracket (infile, ch);

                    else{
                        Error(0);
                        break;
                    }
                }
            }
            else{
                Error(6);
                break;
            }
        }
        cout << endl;
        if (check)
            cout<<"ЭТО СКОБКИ"<<endl;
    }
}
```



```

        break;
    }
    case 2:{
        fp = fopen("test1.txt", "w");
        if (!fp)
            return 0;
        cin>>arr;
        fputs(arr,fp);
        fclose(fp);
        ifstream infile ("test1.txt");
        if (infile >> ch){
            cout << ch;
            if ((ch == 'A') || (ch == '('))
                check = bracket (infile, ch);
            else{
                Error(0);
                break;
            }
        }
        else{
            Error(6);
            break;
        }
        cout << endl;
        if (check)
            cout<<"ЭТО СКОБКИ"<<endl;
        break;
    }
    case 0:{
        exit = false;
        break;
    }
    default: {
        cout<<"НЕВЕРНЫЙ ВВОД"<<endl;
        break;
    }
}

}
return 0;
}

bool bracket (ifstream &infile, char ch){
    static string tab;
    bool forCheck;
    if (ch == 'A')

```

```

        return true;
    else if ( ch == '(' ){
        if (infile >> ch){
            cout << tab << ch << endl;
            if (ch == 'B'){
                tab.push_back('\t');
                if (infile >> ch){
                    cout << tab << ch << endl;
                    forCheck = bracket (infile,ch);
                }
            }
            else{
                Error(3);
                return false;
            }
            if (forCheck){
                if (infile >> ch){
                    cout << tab << ch << endl;
                    forCheck = bracket
(infile,ch);

                }
            }
            else{
                return false; //nepravilnie skobki
            }
            tab.pop_back();
            if (forCheck) {
                if (infile >> ch){
                    cout << tab << ch << endl;
                    if (ch != ')')
                        Error(5);
                    return (ch == ')');
                }
            }
            else{
                Error(5);
                return false;
            }
        }
        else{
            Error (4);
            return false; //net zakrivayuschey
skobki
        }
    }
}

```

```

        else{
            Error(2);
            return false;
        }
    }
    else{
        Error(1);
        return false;
    }
}
else{
    Error(0);
    return false; //не а и не skobka
}
}

```

```

void Error (short k){
cout << endl << "err#" << k << endl;
    switch (k) {
        case 0: cout << "! - НЕВЕРНЫЙ СИМВОЛ" << endl; break;

        case 1: cout << "! - ОТКРЫТАЯ СКОБКА" << endl; break;

        case 2: cout << "! - ОТСУТСТВУЕТ СИМВОЛ В" << endl; break;

        case 3: cout << "! - НЕТ СКОБОК ПОСЛЕ В" << endl; break;

        case 4: cout << "! - НЕВЕРНЫЕ СКОБКИ" << endl; break;

        case 5: cout << "! - НЕТ ЗАКРЫВАЮЩЕЙ СКОБКИ" << endl; break;

        case 6: cout << "! - ПУСТОЙ ФАЙЛ" << endl; break;

        default : cout << "! - ..."; break;

    };
}

```

5.2. Приложение Б: Тестовые случаи.

Таблица 1 — Результаты тестов.

Input	Output	True/False
(B(B(BAA)A)A)	ЭТО СКОБКИ	True
(BAA	Err#5 ! - НЕТ ЗАКРЫВАЮЩЕЙ СКОБКИ	True
(Q)	Err#2 ! - ОТСУТСТВУЕТ СИМВОЛ B	True
A	ЭТО СКОБКИ	True
(AA)	Err#2 ! - ОТСУТСТВУЕТ СИМВОЛ B	True
Пустой файл	Err#6 ! – ПУСТОЙ ФАЙЛ	True