

**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**

Студентка гр. 7383

\_\_\_\_\_

Маркова А. В.

Преподаватель

\_\_\_\_\_

Размочаева Н.В.

Санкт-Петербург

2018

## **СОДЕРЖАНИЕ:**

1.ЦЕЛЬ РАБОТЫ .....	3
2.РЕАЛИЗАЦИЯ ЗАДАЧИ .....	4
3.ТЕСТИРОВАНИЕ .....	6
3.1.ПРОЦЕСС ТЕСТИРОВАНИЯ .....	6
3.2.РЕЗУЛЬТАТЫ ТЕСТИРОВАНИЯ .....	6
4.ВЫВОД.....	7
5.ПРИЛОЖЕНИЕ А: ТЕСТОВЫЕ СЛУЧАИ.....	8
6.ПРИЛОЖЕНИЕ Б: КОД ПРОГРАММЫ .....	9

## 1.ЦЕЛЬ РАБОТЫ

Цель работы: ознакомиться с основными понятиями и приемами рекурсивного программирования, получить навыки программирования рекурсивных процедур и функций на языке программирования C++.

Формулировка задачи: построить синтаксический анализатор для понятия скобки.

скобки  $:: = A \mid \text{скобка скобки}$

скобка  $:: = (B \text{ скобки})$

## 2.РЕАЛИЗАЦИЯ ЗАДАЧИ

В данной работе используется главная функция `main()` и дополнительные функции `bool Brackets(ifstream &infile)`, `void Error (short s)`.

После запуска программы функция `main()` выводит меню на консоль, где можно выбрать пункт, соответствующий нужной операции. Считывается целое число и при помощи оператора `switch()`, выполняется необходимое действие. При нажатие «1», программа проверяет набор скобок, считанный с файла, при нажатие «2» пользователь сам вводит нужную последовательность для проверки. При выборе «3» функция завершает свою работу. Если было введено другое значение, отличное от стандартных, то программа выведет сообщение об ошибке: «Некорректный выбор!» и завершит работу. Программа завершается при выборе «3» или введение неизвестной команды, в противном случае – ожидает дальнейших указаний. Если была выбрана опция проверки скобок в файле, то главная функция проверяет существование этого файла и возможность его прочесть, а затем вызывается функция `bool Brackets(ifstream &infile)`. Если пользователь ввел набор символов сам, то `main()` сначала создаст текстовый файл, куда будет записана считанная с консоли строка, а затем функция обратится к `bool Brackets(ifstream &infile)`. По завершении проверки созданный документ будет удален.

Рекурсивная функция `bool Brackets(ifstream &infile)` получает на вход поток символов, считанных с файла и проверяет является ли первый знак «А» или «(», если да, то в первом случае функция возвращает значение `true`, а во втором продолжает проверку строки. Следующий символ последовательности – «В», если так, то продолжается проверка, в противном случае появляется сообщение ошибки: «Не хватает В!». После

того как функция успешно проверила последовательность символов, она снова вызывает себя же для проверки на скобки. Если вернулось значение `true`, то программа проверяет следующий символ, чтобы он был «)», а после этого опять обращается к себе при положительном результате.

Функция `void Error (short s)` благодаря оператору `switch()` определяет полученную ошибку и выводит сообщение на экран, чтобы пользователь мог понять, где нарушилась правильная последовательность символов.

Разберем для примера работы программы строку `(BA)(BA)A`:



Рисунок 1 – пример работы программы

Цвет стрелочки соответствует одинаковой глубине вложенности рекурсии. Каждый раз вызывая функцию проверки на скобки мы опускаемся ниже и получая значение `true` возвращаемся обратно.

## **3.ТЕСТИРОВАНИЕ**

### **3.1.ПРОЦЕСС ТЕСТИРОВАНИЯ**

Программа собрана в операционной системе Ubuntu 17.04, с использованием компилятора gcc версии 5.4.0 20160609. В других ОС и компиляторах тестирование не проводилось.

Программа может быть скомпилирована с помощью команды:

```
g++ - Wall <имя файла>.cpp
```

### **3.2.РЕЗУЛЬТАТЫ ТЕСТИРОВАНИЯ**

Тестовые случаи представлены в Приложении А.

Исходя из тестовых случаев можно заметить, что во втором тесте программа ведет себя неверно, поэтому была исправлена программа: добавлено в `default` изменение значения флага на `false`, чтобы функция не закичивалась.

Так же была выявлена ошибка в пятом тесте, в связи с этим в программу добавилась проверка на конец строки.

После, тестовые случаи не выявили неправильного поведения программы, что говорит о том, что по результатам тестирования было показано, поставленная задача была выполнена.

## **4.ВЫВОД**

В ходе лабораторной работы были изучены основные свойства и способы использования рекурсии. Была написана программа реализующая рекурсивную функцию **Brackets**, а так же создан синтаксический анализатор скобок на языке программирования C++.

## 5.ПРИЛОЖЕНИЕ А: ТЕСТОВЫЕ СЛУЧАИ

Ввод	Вывод	Верно?
1 В файле: «(BA)A»	(BA)A ЭТО СКОБКИ:)	Да
4	Некорректный выбор! Некорректный выбор! ..... Некорректный выбор!	Нет
8	Некорректный выбор! До свидания!	Да
1 (файл не был создан)	Входной файл не может быть открыт!	Да
2 A()()B 3	A ЭТО СКОБКИ:) До свидания!	Нет
2 A((( 3	A Не хватает ')' или присутствуют лишние символы после скобок. Нет, это не скобки:( До свидания!	Да
2 (B(B(BA)A)A)A	(B(B(BA)A)A)A ЭТО СКОБКИ:)	Да
1 В файле: «(B)()» 3	(B Не хватает 'A' или '(' До свидания!	Да



## 6.ПРИЛОЖЕНИЕ Б: КОД ПРОГРАММЫ

```
#include <iostream>
#include <fstream>
#include <iomanip>

#define N 500

using namespace std;

bool Brackets(ifstream &infile);
void Error(short s); //символ короткой длины, короткий "int"

int main(){
    bool a, run=true;
    char arr[N];
    FILE* f;
    int pick;
    cout<<"\033[34m\tЗдравствуй! Я анализатор скобок, выбери что ты
    хочешь:\033[0m\n 1) Нажми 1, если хочешь проверить скобки в файле.\n
    2) Нажми 2, если хочешь ввести скобки в консоль сам.\n 3) Нажми 3,
    если хочешь выйти из программы.\n";

    while(run){
        cin>>pick; //выбор

        switch(pick){

        case 1:{
            ifstream infile("test.txt");
            if(!infile)cout<<"Входной файл не может быть открыт!"<<endl;
            else{
                a=Brackets(infile);
            }
        }
    }
}
```

```

cout«endl;
if(a) cout«"\033[32mЭТО СКОБКИ:)\033[0m"«endl;
else cout«"\033[31mНет, это не скобки:(\033[0m"«endl;
}
break;
}

case 2:{
f=fopen("test1.txt", "w"); //создание файла
cin»arr;
fputs(arr,f);
fclose(f);
ifstream infile("test1.txt");
if(!infile)cout«"Входной файл не может быть открыт!"«endl;
else{
a=Brackets(infile);
cout«endl;
if(a) cout«"\033[32mЭТО СКОБКИ:)\033[0m"«endl;
else cout«"\033[31mНет, это не скобки:(\033[0m"«endl;
}
remove("test1.txt"); // удаление файла
break;
}

case 3:{
run=false;
break;
}

default:{
run=false;
cout«"\033[31mНекорректный выбор!\033[0m\n"«endl;
break;
}

```

```

}
}
}

cout<<"\033[34mДо свидания!\033[0m\n"«endl;
return 0;
}

bool Brackets(ifstream &infile){
char sim;
infile»sim;
if(sim=='A'){
cout«sim;
if(infile»sim ){
if(sim==')'){
cout«sim;
if(Brackets(infile)) return true;
else {Error(5);return false;} //нет скобок
return true;
}
else {Error(4);return false;} //нет ')'
}
else return true;
}

if(sim=='('){
cout«sim;
infile»sim;//считывание следующего символа
if(sim=='B'){
cout«sim;
if(Brackets(infile));
else {Error(3);return false;} //нет скобок
}

```

```

else {Error(2);return false;} //нет "В"
}
else{Error(1);return false;}//нет 'А' или '('
}

void Error(short s) {
cout<<endl<<"Ошибка №"<<s<<endl;

switch(s){

case 1:
cout<<"Не хватает 'А' или '('"<<endl;
break;
case 2:
cout<<"Не хватает 'В'"<<endl;
break;
case 3:
cout<<"Нет скобок"<<endl;
break;
case 4:
cout<<"Не хватает ')' или присутствуют лишние символы после
скобок"<<endl;
break;
case 5:
cout<<"Нет скобок"<<endl;
break;
}
}
}

```