

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

**ОТЧЕТ**  
**по практической работе №1**  
**по дисциплине «Алгоритмы и структуры данных»**  
**Тема: Программирование рекурсивных алгоритмов**

Студент(ка) гр. 7383

\_\_\_\_\_

Чемова К.А.

Преподаватель

\_\_\_\_\_

Размочаева Н.В.

Санкт-Петербург

2018

## **Содержание**

Цель работы.....	3
Основные теоретические положения.....	3
Постановка задачи.....	3
Реализация.....	3
Тестирование.....	4
Выводы.....	4
ПРИЛОЖЕНИЕ А КОД ПРОГРАММЫ.....	5
ПРИЛОЖЕНИЕ Б ТЕСТИРОВАНИЕ ПРОГРАММЫ.....	12

## Цель работы

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

## Основные теоретические положения

Рекурсивным называется объект, содержащий сам себя или определенный с помощью самого себя. Мощность рекурсии связана с тем, что она позволяет определить бесконечное множество объектов с помощью конечного высказывания. Точно так же бесконечные вычисления можно описать с помощью конечной рекурсивной программы. Рекурсивные алгоритмы лучше всего использовать, когда решаемая задача, вычисляемая функция или обрабатываемая структура данных определены с помощью рекурсии. Если процедура (функция)  $P$  содержит явное обращение к самой себе, она называется прямо рекурсивной. Если  $P$  содержит обращение к процедуре (функции)  $Q$ , которая содержит (прямо или косвенно) обращение к  $P$ , то  $P$  называется косвенно рекурсивной.

## Постановка задачи

Реализовать основанную на рекурсивном алгоритме программу – синтаксический анализатор понятия скобки:

*скобки::=квадратные | круглые*

*квадратные::= [ [ квадратные ] ( круглые ) ] | B*

*круглые::=( ( круглые ) [ квадратные ] ) | A*

## Реализация

Были реализованы булевы функции `Bracket()`, `Round()`, `Square()` для проверки выражения на соответствие понятию *скобки*. Функция `main()` выводит меню для выбора способа ввода данных: 1 – для ввода с клавиатуры и 2 - для использования данных из файла (при невозможности открыть файл выводится

сообщение об ошибке). Далее вызывается Bracket(), в которой происходит вызов Square() и Round(). Последние две функции определены через самих себя и через друг друга, что делает их рекурсивными. Square() проверяет выражение на соответствие выражения понятию квадратные скобки, Round(), в свою очередь, на – круглые скобки. После проверки, выводится сообщение о результате работы программы. Если выражение оказалось скобками, то выводится само выражение и фраза «Это скобки». В противном случае, выводится сообщение об ошибке и выражение до того места, в котором было найдено не соответствие.

### **Тестирование**

Программа собрана в операционной системе Ubuntu 16.04 LTS с использованием компилятора g++. В других ОС и компиляторах тестирование не проводилось.

### **Выводы**

В ходе лабораторной работы было изучено понятие рекурсии и освоены основные принципы рекурсивного программирования, написана программа, позволившая овладеть навыками программирования рекурсивных процедур и функций на языке программирования C++.

## ПРИЛОЖЕНИЕ А

### КОД ПРОГРАММЫ

```
/* Вариант 21
   Построить синтаксический анализатор для понятия скобки.
   скобки::=квадратные | круглые
   квадратные::=[[кваратные](круглые)] | В
   круглые::=((круглые)[квадратные]) | А
*/

#include <iostream>
#include <cstdio>
#include <fstream>
#include <iomanip>
using namespace std;

void File(ifstream &infile, char ch);
bool Bracket(ifstream &infile, char ch);
bool Round(ifstream &infile, char ch, int &err);
bool Square(ifstream &infile, char ch, int &err);
void Error(short k);

int main() {

    int n;
    char ch;
    char data[1000];
    FILE *f;
    cout<<"\nВас приветствует анализатор скобок!\n\nВыберите способ ввода данных:\n1 - Для ввода данных с клавиатуры\n2 -
    Для использования данных из файла\n0 - Для выхода из программы"<<endl;
    cin>>n;

    switch(n) {

    case 0:

        return 0;

    case 1: {

        f = fopen("test.txt", "w");
        cin>>data;
        fputs(data,f);
        fclose(f);

        ifstream infile("test.txt");
        File(infile, ch);
        infile.close();
        remove("test.txt");
        return 0;
    }

    case 2: {

        ifstream infile("test1.txt");
        File(infile, ch);
        return 0;
    }
    }

    void File(ifstream &infile, char ch){

        if(!infile) {
            cout<< "Входной файл не открыт!"<<endl;
        }

        if (Bracket(infile, ch) && !(infile>>ch)) cout<<"\nЭто скобки!"<<endl;
        else cout<<"\nЭто НЕ скобки!"<<endl;
    }

    bool Bracket(ifstream &infile, char ch) {

        // скобки::= квадратные | круглые
```

```

int err = 0, err1;

infile>>ch;

if (!infile>>ch) {
    Error(err);
    return false;
}

if (Square(infile, ch, err)) return true;
else { // это не квадратные скобки
    err1 = err; // в err1 записана ошибка из Square

    if (Round(infile, ch, err)) return true;
    else {
        Error(err); // ошибка в круглых скобках
        return false;
    }
    Error(err1); // ошибка в квадратных скобках
    return false;
}
}

bool Square(istream &infile, char ch, int &err) {

// квадратные::=[[кваратные](круглые)] | В
// ch - текущий символ входной строки

    if (ch == 'B') {
        cout<<ch;
        return true;
    }

    if (ch != '[') {
        err = 6;
        return false;
    }

    cout<<ch;
    infile>>ch;

    if (ch != '[') {
        err = 1;
        return false;
    }

    cout<<ch;
    infile>>ch;

    if (!Square(infile, ch, err)) return false;

    infile>>ch;

    if (ch != ']') {
        err = 2;
        return false;
    }

    cout<<ch;
    infile>>ch;

    if (ch != '(') {
        err = 3;
        return false;
    }

    cout<<ch;
    infile>>ch;

    if (!Round(infile, ch, err)) return false;

    infile>>ch;

    if (ch != ')') {
        err = 4;
        return false;
    }
}

```

```

    cout<<ch;
    infile>>ch;

    if (ch != ']') {
        err = 2;
        return false;
    }

    cout<<ch;
    return true;
}

bool Round(ifstream &infile, char ch, int &err) {
    // круглые:=((круглые)[квадратные]) | A
    // ch - текущий символ входной строки

    if (ch == 'A') {
        cout<<ch;
        return true;
    }

    if (ch != '(') {
        err = 7;
        return false;
    }

    cout<<ch;
    infile>>ch;

    if (ch != '(') {
        err = 3;
        return false;
    }

    cout<<ch;
    infile>>ch;

    if (!Round(infile, ch, err)) return false;

    infile>>ch;

    if (ch != ')') {
        err = 4;
        return false;
    }

    cout<<ch;
    infile>>ch;

    if (ch != '[') {
        err = 1;
        return false;
    }

    cout<<ch;
    infile>>ch;

    if (!Square(infile, ch, err)) return false;

    infile>>ch;

    if (ch != ']') {
        err = 2;
        return false;
    }

    cout<<ch;
    infile>>ch;

    if (ch != ')') {
        err = 4;
        return false;
    }

    cout<<ch;
    return true;
}

```

```

}

void Error(short k) {

    cout<< endl <<"\nОшибка №"<< k << ": ";

    switch(k) {
    case 0:
        cout<<"Пустая входная строка.";
        break;
    case 1:
        cout<<"Не хватает '['.";
        break;
    case 2:
        cout<<"Не хватает ']'. ";
        break;
    case 3:
        cout<<"Не хватает '('.";
        break;
    case 4:
        cout<<"Не хватает ')'. ";
        break;
    case 5:
        cout<<"Неправильная последовательность!";
        break;
    case 6:
        cout<<"Не хватает '[' или 'B'.";
        break;
    case 7:
        cout<<"Не хватает '(' или 'A'.";
        break;
    }
    return;
}

```



## ПРИЛОЖЕНИЕ Б

### ТЕСТИРОВАНИЕ ПРОГРАММЫ

Входные данные	Выходные данные
1 А	А Это скобки!
1 В	В Это скобки!
1 ((A)[B])	((A)[B]) Это скобки!
[[B](A)]	[[B](A)] Это скобки!
((A)[[[B](A)]])	((A)[[[B](A)]]) Это скобки!
2 В файле: (((((((((((((((((0))))))))))))))	(((((((((((((((((( Ошибка №7: Не хватает '(' или 'A'. Это НЕ скобки!
1 ((A))))	((A) Ошибка №1: Не хватает '['. Это НЕ скобки!
1 [[B](A_]	[[B](A Ошибка №7: Не хватает '(' или 'A'. Это НЕ скобки!
2 В файле: [[[[B]](((A)[B]))]](A)]	[[[[B]] Ошибка №7: Не хватает '(' или 'A'. Это НЕ скобки!
1 (((A))[B]))[B])	(((A) Ошибка №1: Не хватает '['. Это НЕ скобки!
2 В файле (((A)[[[B](A)]]))[[[B](A)]]	(((A)[[[B](A)]]))[[[B](A)]] Это скобки!
2 В файле (((A)[[[B](A)]]))[[[B]chgjb	(((A)[[[B](A)]]))[[[B](A)]] Это НЕ скобки!