

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. 9384

Соседков К.С.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2020

Цель работы.

Разработать синтаксический анализатор используя рекурсивную функцию.

Задание. (Вариант № 16)

Построить синтаксический анализатор для понятия скобки.

скобки::=A | B | (скобки скобки)

Выполнение работы.

При выполнении работы была написана рекурсивная функция *isBracket*, принимающая на вход строку. Функция возвращает true если входная строка является скобкой, иначе возвращает false.

Алгоритм работы функции *isBracket()*:

- считывается следующий символ из строки
- выполняется проверка является ли данный символ скобкой A или B
- если символ не является скобкой A или B, выполняется проверка на наличие конструкции (скобки скобки).
- в остальных случаях функция возвращает false

Так же были написаны дополнительные функции *readNextChar* и *input*.
char readNextChar(string s) — считывает следующий символ из строки.
int input() - считывает число от 1 до 3 с терминала, и возвращает его.

Тестирование.

Результаты тестирования представлены в табл. 1.

Таблица 1 – Результаты тестирования

№ п/п	Входные данные	Выходные данные	Комментарии
1.		Is not a bracket	Пустая строка
2.	((B A) B)	((B A) B) is a bracket	
3.	((B A) (A B))	((B A) (A B)) is a bracket	

Выводы.

При выполнении работы было изучено понятие рекурсия, как в теории так и на практике, а так же был написан синтаксический анализатор.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: lab1.cpp

```
#include <iostream>
```

```
#include <fstream>
```

```
#include <limits>
```

```
#include <string>
```

```
using namespace std;
```

```
bool isBrackets(string s);
```

```
char readNextChar(string s);
```

```
int input();
```

```
// скобки ::= A | B | ( скобки скобки )
```

```
int main() {
```

```
    int type = input();
```

```
    std::string line;
```

```
    if(type == 1) {
```

```
        cin.ignore();
```

```
        ifstream infile ("file.txt");
```

```
        std::getline(infile, line);
```

```
    }
```

```
    else if(type == 2) {
```

```
        cout << "Input string: \n";
```

```
        cin.ignore();
```

```

        getline(std::cin, line);
    }

    cout << "Current string: " << line << std::endl;
    bool result = isBrackets(line);

    if(result && ((int)readNextChar(line)==0)) {
        cout << line << " is a bracket\n";
    }

    else {
        cout << line << " is not a bracket\n";
    }

    return 0;
}

int input() {
    cout << "Lab 1(16) -> скобки::= A | B | ( скобки скобки )\n";
    cout << "1) Input from file ('file.txt' by default)\n";
    cout << "2) Terminal input\n";
    cout << "Input type (1, 2): " << endl;

    int type = 0;
    do {
        cin >> type;
        if (cin.fail()) {
            cout << "Please enter an integer (1, 2)\n";
            cin.clear();
            cin.ignore(numeric_limits<streamsize>::max(), '\n');

```

```

    }
    } while(type != 1 && type != 2);

    return type;
}

char readNextChar(string s) {
    static int str_size = s.size();
    static int index = 0;
    static string last_string = s;
    if(last_string != s) {
        str_size = s.size();
        index = 0;
        last_string = s;
    }

    while(s[index] == ' ' && index < str_size) {
        index++;
    }
    index++;
    return s[index-1];
}

bool isBrackets(string s) {
    char symbol;
    symbol = readNextChar(s);
    cout << "Read char: " << symbol << std::endl;
    if(symbol == 'A' || symbol == 'B') {
        cout << "Current char is " << symbol << ". Return true\n";
        return true;
    }
}

```

```

}
else if(symbol == '(') {
    cout << "Current char == '('\n";
    cout << "Read first bracket\n";
    bool firstBracket = isBrackets(s);
    cout << "Read second bracket\n";
    bool secondBracket = isBrackets(s);
    symbol = readNextChar(s);
    if(!firstBracket) {
        cout << "First bracket read error. Return false\n";
        return false;
    }
    if(!secondBracket) {
        cout << "Second bracket read error. Return false\n";
        return false;
    }
    if(symbol != ')') {
        cout << symbol << " Last char is not ')'. Return false\n";
        return false;
    }
    cout << "OK\n";
    return true;
}
cout << "Current char != 'A', 'B', '('. Return false\n";
return false;
}

```