

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

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

Студент гр. 9303

Скворчевский Б.С.

Преподаватель

Филатов А.Ю.

Санкт-Петербург

2020

Цель работы.

Ознакомиться с основными понятиями и приемами рекурсивного программирования. Научиться решать задачи при помощи рекурсивных функций.

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.

21. Построить синтаксический анализатор для понятия *скобки*.

скобки::=*квадратные* | *круглые*

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

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

Выполнение работы.

Программа принимает строку и записывает её в переменную *input*. Далее она проверяет первый символ строки: если он является квадратной либо круглой скобкой или буквой *A* либо *B*, то вызывается функция *Brackets()*, которая

проверяют верность введённого выражения и результат которой записывается в переменную `ans`. Иначе программа выводит текст «Введена неверная строка».

Функция `bool Brackets(string &input, int index)` сначала проверяет несколько условий для обработки ошибок. Далее для каждого из возможных символов, а именно [или], или (, или), или A, или B, выполняются определённые проверки и программа снова запускает данную функцию до тех пор, пока не пройдётся по всем функциям. В случае, если хотя бы одно условие не выполняется, то функция возвращает `false`.

После выполнения функции `Brackets()` производится проверка переменной `ans`, в результате чего выводится результат выполнения программы в виде одной из строк: «Круглые скобки» либо «Квадратные скобки», либо «Не скобки».

Выводы.

Были изучены основные понятия и приёмы рекурсивного программирования. Была реализована программа, представляющая синтаксический анализатор скобок, и проверяющая введённую строку на соответствие разрешённым шаблонам.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: main.cpp

```
#include <iostream>
#include <string>

using namespace std;

bool Brackets(string &input, int index) {
    index += 1;
    char c = input[index]; // for debug
    if (input[0] == 'B' && input.length() == 1) {
        return true;
    }

    if (input[0] == 'A' && input.length() == 1) {
        return true;
    }

    if((input[0] == '[' && input[input.length() - 1] != ']') ||
(input[0] == '(' && input[input.length() - 1] != ')'))
        return false;

    if(input[index] == '['){
        bool ans = Brackets(input, index);
        if(ans)
            return true;
        else
            return false;
    }
}
```

```

    if(input[index] == '('){
        bool ans = Brackets(input, index);
        if(ans)
            return true;
        else
            return false;
    }

    if(input[index] == ']){
        if((index + 1) == input.length() && (input[index - 1]
== ')' || input[index - 1] == 'A')){
            return true;
        } else if(input[index - 1] != ')' && input[index - 1]
!= 'A')
            return false;
        else {
            bool ans = Brackets(input, index);
            if(ans)
                return true;
            else
                return false;
        }
    }

    if(input[index] == '){
        if((index + 1) == input.length() && (input[index - 1]
== ']' || input[index - 1] == 'B')){
            return true;
        } else if(input[index - 1] != ']' && input[index - 1]
!= 'B')
            return false;
        else {
            bool ans = Brackets(input, index);
            if(ans)
                return true;

```

```

        else
            return false;
    }
}

if(input[index] == 'A') {
    if(input[index - 1] == ']' ||
        input[index - 1] == '(' || input[index - 1] == 'B')
    {

        bool ans = Brackets(input, index);
        if(ans)
            return true;
        else
            return false;
    }
    else{
        return false;
    }
}

if(input[index] == 'B') {
    if(input[index - 1] == '[' ||
        input[index - 1] == ')' || input[index - 1] == 'A')
    {

        bool ans = Brackets(input, index);
        if(ans)
            return true;
        else
            return false;
    }
    else {
        return false;
    }
}
}

```

```

int main() {
    string input;
    cin >> input;
    string A = "()[]";
    string B = "[]()";
    bool ans = false;

    if(input[0] == '[' || input[0] == 'B' ||
        input[0] == '(' || input[0] == 'A'){
        ans = Brackets(input, -1);

    } else {
        cout << "Введена неверная строка" << endl;
        return 0;
    }

    if((input[0] == '(' || input[0] == 'A') && ans)
        cout << "Круглые скобки" << endl;
    else if((input[0] == '[' || input[0] == 'B') && ans)
        cout << "Квадратные скобки" << endl;

    if(!ans)
        cout << "Не скобки" << endl;

    return 0;
}

```

ПРИЛОЖЕНИЕ Б

ТЕСТИРОВАНИЕ

Таблица 1 - Примеры тестовых случаев

№	Входные данные	Выходные данные	Комментарий
1	[BA]	Квадратные скобки	Программа работает корректно
2	((AB)[BA])	Круглые скобки	Программа работает корректно
3	((AB)[BA]	Не скобки	Программа работает корректно
4	((AB)[B])	Не скобки	Программа работает корректно
5	[AB]	Не скобки	Программа работает корректно
6	((AB)[[BA]A])	Круглые скобки	Программа работает корректно
7	((((AB)[BA])B)[[BA]A])	Круглые скобки	Программа работает корректно

8	[[[BA](AB)]((AB)[BA])]	Квадратные скобки	Программа работает корректно
9	[[[BA](AB)]((AB)[B((AB)[BA])])]	Квадратные скобки	Программа работает корректно
10	A[BA]	Не скобки	Программа работает корректно