

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

Студент гр. 8304

Птухов Д.А.

Преподаватель

Фирсов М.А.

Санкт-Петербург

2019

Цель работы.

Изучить основы рекурсии и составления эффективных алгоритмов.

Основные теоретические положения.

Построение синтаксического анализатора для понятия *скобки*.

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

Ход работы:

Для решения данной задачи была реализована рекурсивная функция `check`, которая заменяет каждую подстроку “(B A A)” полученной строки на подстроку “A” и рекурсивно вызывает саму себя от измененной строки. Выход из рекурсии осуществляется посредством двух проверок. Первая – проверка на то, что полученная строка равна строке “A” (так как дальнейшая замена бесполезна) и возвращает значение `True`, вторая – проверка на подстроки (B A A) в полученной строке, если на одном из шагов таковой не было найдено, то функция возвращает значение `False`. Диалог с пользователем осуществляется при помощи функции `Dialog`, которая обрабатывает полученные данные и передает входную строку в функцию `check`. Также было реализовано 3 дополнительных функции, позволяющих избежать дублирования кода.

Выводы.

Был получен опыт работы с рекурсией и с синтаксическим анализатором.

ПРИЛОЖЕНИЕ

1) ТЕСТИРОВАНИЕ:

Программой было успешно пройдено большое кол-во тестов. Примеры тестирования:

1. Входные данные – (B A (B (B (B A (B A A)) (B A A)) A)). Способ ввода – консольный. Способ вывода – консольный.

```
Choose input format:
1)Enter string

2) Read from file (Default file is located along the path:D:/LAB1_SOURCE/InputSource.txt)
If you want to change file location, you have to enter path as second argument
Don't forget to change all '\' to '/'

1
Choose output format:
1)Console

2)File (Default file is located along the path: D:/LAB1_SOURCE/OutputSource.txt)
If you want to change file location, you have to enter path as second argument
Remember that debugging output will be saved with programm result

1
Enter string:
(B A (B (B (B A (B A A)) (B A A)) A))

Value of the check-string after the next function call : (B A (B (B (B A (B A A)) (B A A)) A))

Recursion depth: 0
-----
Value of the check-string after the next function call : (B A (B (B (B A A) (B A A)) A))

Recursion depth: 1
-----
Value of the check-string after the next function call : (B A (B (B A (B A A)) A))

Recursion depth: 2
-----
Value of the check-string after the next function call : (B A (B (B A A) A))

Recursion depth: 3
-----
Value of the check-string after the next function call : (B A (B A A))

Recursion depth: 4
-----
Value of the check-string after the next function call : (B A A)

Recursion depth: 5
-----
Entered string: (B A (B (B (B A (B A A)) (B A A)) A))
Result: True
-----
```

2. Входные данные – (B A (B (B (B A (B A A)) (B A B)) A)). Способ ввода – консольный. Способ вывода – файловый (использование пути по умолчанию).

```

OutputSource.txt — Блокнот
Файл  Правка  Формат  Вид  Справка
Value of the check-string after the next function call : (B A (B (B (B A (B A A)) (B A B)) A))

Recursion depth: 0
-----
Value of the check-string after the next function call : (B A (B (B (B A A) (B A B)) A))

Recursion depth: 1
-----
Entered string: (B A (B (B (B A (B A A)) (B A B)) A))
Result: False
-----

```

3. Входные данные – (B A (B (B (B (B A A) A) A) B)). Способ ввода – файловый (использование пути по умолчанию). Способ вывода – консольный.

```

InputSource.txt — Блокнот
Файл  Правка  Формат  Вид  Справка
|(B A (B (B (B (B A A) A) A) B))

```

```

Choose input format:
  1)Enter string

  2) Read from file (Default file is located along the path:D:/LAB1_SOURCE/InputSource.txt)
    If you want to change file location, you have to enter path as second argument
    Don't forget to change all '\' to '/'
2
Choose output format:
  1)Console

  2)File (Default file is located along the path: D:/LAB1_SOURCE/OutputSource.txt)
    If you want to change file location, you have to enter path as second argument
    Remember that debugging output will be saved with programm result
1
Value of the check-string after the next function call : (B A (B (B (B (B A A) A) A) B))

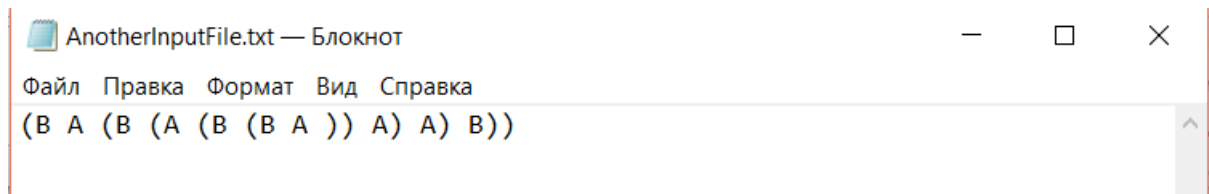
Recursion depth: 0
-----
Value of the check-string after the next function call : (B A (B (B (B A A) A) B))

Recursion depth: 1
-----
Value of the check-string after the next function call : (B A (B (B A A) B))

Recursion depth: 2
-----
Entered string: (B A (B (B (B (B A A) A) A) B))
Result: False
-----

```

4. Входные данные – (B A (B (A (B (B A)) A) A) B)). Способ ввода – файловый. Способ вывода – консольный.

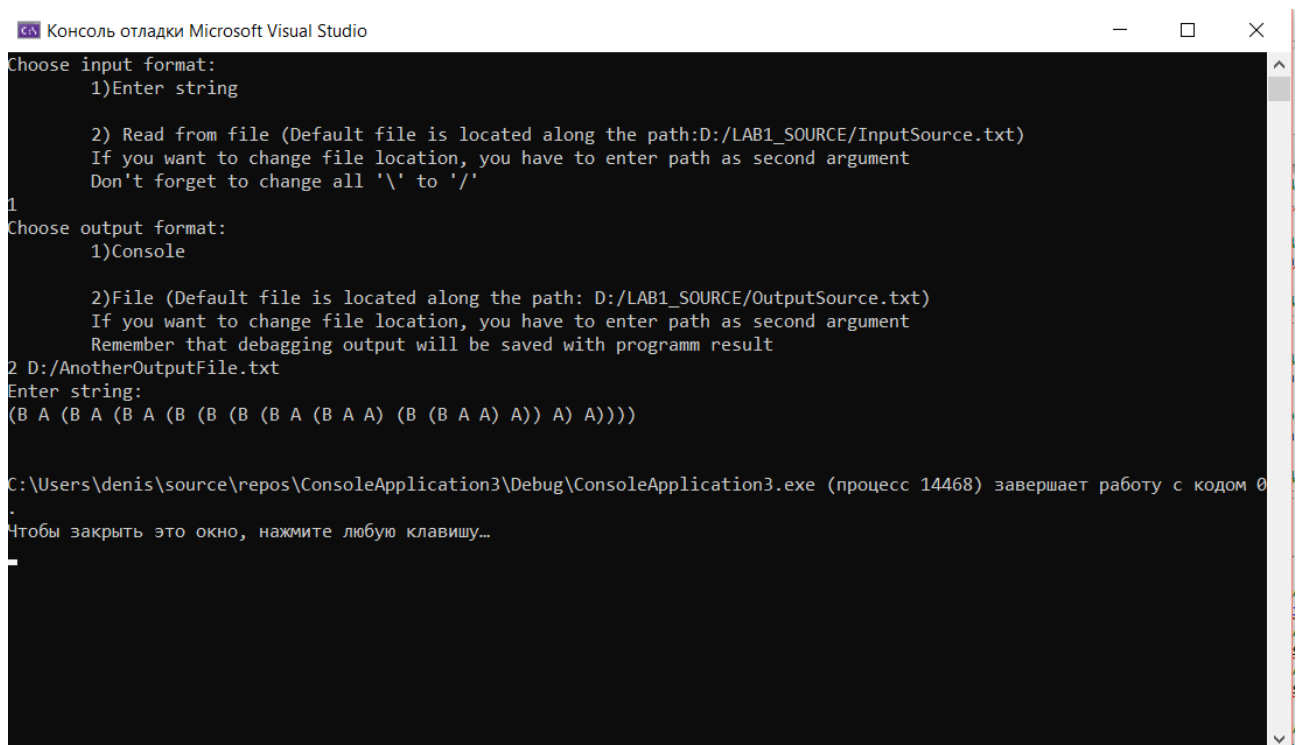


```
Choose input format:
  1)Enter string

  2) Read from file (Default file is located along the path:D:/LAB1_SOURCE/InputSource.txt)
    If you want to change file location, you have to enter path as second argument
    Don't forget to change all '\' to '/'
2 D:/AnotherInputFile.txt
Choose output format:
  1)Console

  2)File (Default file is located along the path: D:/LAB1_SOURCE/OutputSource.txt)
    If you want to change file location, you have to enter path as second argument
    Remember that debugging output will be saved with programm result
1
Entered string: (B A (B (A (B (B A )) A) A) B))
Result: False
-----
```

5. Входные данные – (B A (B A (B A (B (B (B (B A (B A A) (B (B A A) A)) A) A)))). Способ ввода – консольный. Способ вывода – файловый.




```

//создание стандартного потока вывода
if (ovar[0] == '1')
    of = 0;
//создание файлового потока вывода с сопутствующей проверкой
else if (ovar[0] == '2')
{
    out.open(make_path(ovar, "D:/LAB1_SOURCE/OutputSource.txt"));
    if (!(out.is_open()))
    {
        std::cout << "Uncorrect path!!!";
        return;
    }
    of = 1;
}
//иные случаи
else
{
    std::cout << "Goodbye" << std::endl;
    return;
}

//считывание входных данных через консоль
if (ivar[0] == '1')
{
    std::string s;

    std::cout << "Enter string:\n";
    std::getline(std::cin, s);
    std::cout << std::endl;

    //проверка на отсутствие "лишних" символов
    if (!check_strElements(s))
    {
        std::cout << "Entered string: " + s + "\nResult: False - Uncorrect
symbols";
        return;
    }

    //выбор потока вывода
    if (!of)
        call(s, std::cout);
    else
    {
        call(s, out);
        out.close();
    }
}
//считывание входных данных через файл
else if (ivar[0] == '2')
{
    std::string data_;

    //открытие потока
    std::ifstream f(make_path(ivar, "D:/LAB1_SOURCE/InputSource.txt"));

    //проверка на то, что поток действительно был открыт
    if (!(f.is_open()))
    {
        std::cout << "Uncorrect path!!!";
        return;
    }

    //считывание данных из файла
    std::getline(f, data_);
}

```

```

        //проверка на отсутствие "лишних" символов
        if (!check_strElements(data_))
        {
            std::cout << "Entered string: " + data_ + "\nResult: False - Incorrect
symbols";
            return;
        }

        //выбор потока вывода
        if (!of)
            call(data_, std::cout);
        else
        {
            call(data_, out);
            out.close();
        }
    }
    //иные случаи
    else
    {
        std::cout << "Goodbye!";
        return;
    }
}

bool check_strElements(std::string s)
{
    for (char i : s)
        if (i != 'A' && i != 'B' && i != '(' && i != ')' && i != ' ')
            return 0;
    return 1;
}

std::string make_path(std::string var, std::string default_path)
{
    std::string path;
    std::string data_;

    //запись полученного пути в переменную path
    for (auto i = ++var.begin(); i != var.end(); i++)
        if (*i != ' ')
            path += *i;

    //проверка на то, что путь был введен
    if (path.empty())
        path = default_path;
    return path;
}

void call(std::string s, std::ostream& const out)
{
    std::string message = "Entered string: " + s;
    out << message << std::endl << "Result: " + bool_to_string(check(s, out)) <<
std::endl;
    out << "-----" << std::endl;
}

bool check(std::string& terminate, std::ostream& const out, size_t n)
{
    //Выход из рекурсии, если дальнейшая замена подстроки (B A A) на подстроку A
    бесполезно

```



```

    if (terminate == "A")
        return 1;

    //Нахождение очередной подстроки (B A A)
    size_t it = terminate.find("(B A A)");

    //Проверка на наличие вышеуказанной подстроки
    if (it == std::string::npos)
        return 0;

    //Отладочные выводы
    out << "Value of the check-string after the next function call : " << terminate <<
std::endl << std::endl;
    out << "Recursion depth: " << n << std::endl << "-----" << std::endl;

    //Удаление подстроки (B A A) из введенной пользователем строки
    terminate.erase(it + terminate.begin(), it + terminate.begin() + 7);

    //Вставка на место удаленной подстроки символа A
    terminate.insert(it + terminate.begin(), 'A');

    //Рекурсивный вызов функции от измененной строки
    return check(terminate, out, ++n);
}

std::string bool_to_string(bool a)
{
    return a ? "True" : "False";
}

int main()
{
    std::cout << "Choose input format:\n\t1)Enter string\n\n\t2) Read from file (Default
file is located along the path:"
        "D:/LAB1_SOURCE/InputSource.txt)\n\tIf you want to change file location, you
have to enter path as second argument\n\t"
        "Don't forget to change all '\\\\' to '/'\n";
    //Чтение входной строки (использование getline обусловлено возможностью присутствия
пробелов во входной строке)
    Dialog();

    return 0;
}

```