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. 7383 _____ Прокопенко Н.

Преподаватель _____ Размочаева Н. В.

Санкт-Петербург
2018

Содержание

Цель работы

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

Формулировка задачи: Требуется написать синтаксический анализатор понятия скобки, где скобки определяются как:

скобки: = A | A (ряд_скобок)
ряд_скобок: = скобки | скобки ; ряд_скобок.

Реализация задачи

В данной работе было написано несколько функций для реализации задачи. Перечень функций:

bool bracket_exp(string str) - функция для проверки скобок. В данную функцию передается строка для поиска скобок, которая проверяется посимвольно. Если символ не является нарушает определения скобки, то продолжается рекурсивный поиск до тех пор, пока символ не нарушит определение скобок или не кончится вводимая строка. В данной программе учитывается возможное наличие пробелов между символами при условии, что ранее они не были встречены. Функция возвращает значение true или false.

string delet(string str) – функция удаления пробелов. Данной функции передается строка, которая проверяется на пробелы. При нахождении данный пробел удаляется. После чего измененная строка возвращается.

int main ( ) - головная функция. В данной функции выводится меню программы, после чего считывается выбранный вариант и запускается функция switch( ), которая находясь в теле цикла while( ), будет выполняться до тех пор, пока пользователь не захочет выйти из программы.

При введении пользователем «1», программа запрашивает входную строку, вводимую с клавиатуры. Затем вызывается рекурсивная функция,

описанная выше. После чего выводится результат проверки строки и пользователю предлагается выбрать другой пункт меню.

При введении пользователем «2», программа предлагает ввести имя входного файла, и если файл с таким файлом существует, то запускается функция проверки выражения. Перед выводом результата проверки выводится на экран изначальная строка, содержащаяся в файле. Если же такого файла не существует, то выводится соответствующая надпись и предлагается повторить выбор.

При введении пользователем «0», происходит выход из программы.

При введении пользователем других данных предлагается выбрать один из пунктов меню, описанных выше. Код программы представлен в Приложении А.

Тестирование

Программа собрана в операционной системе Ubuntu 17.04 с использованием компилятора g++. В других ОС и компиляторах тестирование не проводилось. Результаты тестирования показали, что поставленная цель выполнена. Результаты тестирования представлены в Приложении Б.

Выводы

В ходе выполнения лабораторной работы были изучены основные понятия и приемы рекурсивного программирования, получены навыки программирования рекурсивных процедур и функций на языке программирования C++. Была изучена функция считывания и игнорирования символа cin.ignore( ). Также был написан синтаксический анализатор для понятия скобок.

Приложение А: Код программы

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

using namespace std;

bool bracket_exp(string str, unsigned int cur_pos=0, bool
flag=false, int opened_br=0)
{
    if(cur_pos < str.length())
    {
        if(str[cur_pos]=='A' && flag==false)
        {
            flag = true;
            flag = bracket_exp(str, cur_pos+1, flag, opened_br);
        }
        else if(str[cur_pos] == '(' && flag==true && str[cur_pos-1]
!= ')')
        {
            flag = false;
            opened_br++;
            flag = bracket_exp(str, cur_pos+1, flag, opened_br);
        }
        else if(str[cur_pos] == ';' && flag == true && opened_br >
0)
        {
            flag = false;
            flag = bracket_exp(str, cur_pos+1, flag, opened_br);
        }
        else if(str[cur_pos] == ')' && flag == true && opened_br >
0)
        {
            opened_br--;
            flag = bracket_exp(str, cur_pos+1, flag, opened_br);
        }
        else
        {
            return flag = false;
        }
    }
    else if(opened_br != 0)
    {
        return flag = false;
    }
    return flag;
}

string delet(string str){
  int i=0;
  while(str.length()>i){
    if(str[i] == ' ')
      str.erase(i,1);
```

```
    else  i++;
  }
  return str;
}

int main()
{
    int sw_var, i=0;
    string str;
    bool result = false;
    string file_name;
    fstream file;
    cout << "Menu"<< '\n';
    cout << "0-exit from the program"<< '\n';
    cout << "1-input a string from the keyboard"<< '\n';
    cout << "2-input a line from a file" << '\n';
    cin >> sw_var;
    cin.ignore();
    if(sw_var == 3)
        sw_var = 4;

    while(sw_var)
    {
        switch(sw_var)
        {
        case 1:
            cout << "Enter the string:"<< '\n';
            getline(cin, str);
            if(str != ""){
                result = bracket_exp(delet(str));
              }
            sw_var = 3;
            break;
        case 2:
            cout << "Enter the name of the file:"<< '\n';
            cin >> file_name;
            cin.ignore();
            file.open(file_name, fstream::in);
            if(!file.is_open()){
                cout << "Error opening file.\n";
                sw_var = 4;
              }
            else{
                getline(file, str);
                cout << delet(str) << endl;
                if(str != "")
                    result = bracket_exp(delet(str));
                sw_var = 3;
                file.close();
              }
            break;
        case 3:
            if(result)
                cout << "True. This is the concept of brackets.\n";
```

```
            else
                cout << "False. This is not the concept of
brackets.\n";
            sw_var = 4;
            break;
        case 0:
            break;
        default:
            cout << "Enter again.\n";
            cin >> sw_var;
            cin.ignore();
            if(sw_var == 3)
                sw_var = 4;
            break;
        }
    }
    return 0;
}
```

Приложение Б: Тестовые случаи

Таблица 1 — Результаты тестов.

Input	Output	True/False
A(A)	True. This is the concept of brackets.	True
(BAA	False. This is not the concept of brackets.	False
(Q)	False. This is not the concept of brackets.	False
A(A; A; A(A))	True. This is the concept of brackets.	True
A (A(A:A); A(A(A)))	True. This is the concept of brackets.	True
test.txt	A (A(A:A); A(A(A))) True. This is the concept of brackets.	True