

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

Студентка гр. 7383

Ханова Ю.А.

Преподаватель

Размочаева Н.В

Санкт-Петербург

2018

Содержание

Цель работы	3
Реализация задачи	3
Тестирование	4
Выводы	4
Приложение А. Код программы	5
Приложение Б. Тестовые случаи	7

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 список параметров, где скобки определяются как:

Список параметров: = параметр | параметр, список параметров

параметр: = имя = цифра цифра | имя = (список параметров)

имя: = буква буква буква

Реализация задачи

В данной лабораторной работе были реализованы следующие функции для работы со списком параметров:

`Parameter_list(ifstream &infile, bool q);` - осуществляется начало работы с каждым параметром или списком параметров, рекурсивная функция, вход в рекурсию осуществляется при верном значении возвращенном функцией `Name`.

`Parameter(ifstream &infile, char s);` - проверка на параметр или список параметров, вызывается в функции `Name` и происходит анализ строки на соответствие определению параметра (2-х значное число или список параметров в скобках).

`Name(ifstream &infile, char s);` - проверка строки на соответствие формату имени (3 буквы идущие подряд и знак « = ») а затем вызывается функция `Parameter`.

`Output(ifstream &infile);` - проверка входного файла на пустоту и вывод ответа, в зависимости от возвращаемого значения функции `Parameter_list`.

`SkipSpace(ifstream &infile, char s);` - учитывает то, что некоторые пробелы не влияют на корректность строки.

`Error(short k);` - выводит текст ошибок.

В функции `main()` было создано меню с возможным выбором ввода данных (1 – из файла, 2 – с клавиатуры, 3 – выход из программы).

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 данной лабораторной работы были освоены основные принципы работы с рекурсивными функциями на языке программирования C++. Также был создан набор функций для выполнения поставленной задачи.

ПРИЛОЖЕНИЕ А. Код программы

```
#include <iostream>
#include <fstream>
#include <iomanip>
#include <cctype>
#include <string>
#include <vector>
#include <cstdio>
using namespace std;
#define N 500

bool Parameter_list(ifstream &infile, bool q = false);
bool Parameter(ifstream &infile, char s);
bool Name(ifstream &infile, char s);
void Error(short k);
void Output(ifstream &infile);
void SkipSpace(ifstream &infile);

int main()
{
    char str[N];
    bool ex = true;
    FILE* f;
    while(ex) {
        cout << "Анализатор понятия список_параметров:" << endl;
        cout << "1 - Ввод из файла" << endl;
        cout << "2 - Ввод с клавиатуры" << endl;
        cout << "3 - Выход из программы" << endl;
        cout << "Введите номер из меню: ";
        int ch;
        cin >> ch;
        switch(ch) {
            case 1: {
                ifstream infile("test.txt");
                Output(infile);
                break;
            }
            case 2: {
                f=fopen("test1.txt", "w");
                cin>>str;
                fputs(str,f);
                fclose(f);
                ifstream infile("test1.txt");
                Output(infile);
                remove("test1.txt");
                break;
            }
            case 3: {
                ex = false;
                break;
            }
        }
    }
}
```

```

    }
    default: {
        ex = false;
        cout << "Ошибка ввода!" << endl;
        break;
    }
}
}
return 0;
}

void Output(ifstream &infile) {
    if(!infile) {
        cout << "Входной файл не открыт!" << endl;
    }
    bool b = Parameter_list(infile);
    cout << endl;
    if(b) cout << "Это список параметров!" << endl;
    else cout << "Это НЕ список параметров!" << endl;
}

void SkipSpace(ifstream &infile) {
    string str;
    int i = 0;
    getline(infile, str);
    for(i = 0; i < str.length(); i++) {
        if(str[i] == ' '){
            str.erase(i, 1);
        }
    }
    for(int i = 0; i < str.length(); i++){
        infile.putback(str[i]);
    }
}

bool Parameter(ifstream &infile, char s) {
    bool isname = false;
    if(isdigit(s) && infile >> s) {
        cout << s;
        if(isdigit(s)) {
            return true;
        }
        else {
            Error(6);
            return false;
        }
    }
    else if (s == '(' && infile >> s) {
        cout << s;
        isname = Name(infile, s);
        if(isname) {
            if(infile >> s) {
                if(s == ')') {
                    cout << s;
                }
            }
        }
    }
}

```

```

        return true;
    }
    else if(s==',') {
        cout << s;
        infile >> s;
        cout << s;
        isname=Name(infile, s);
        if(isname) return true;
    }
}
}
else {
    Error(5);
    return false;
}
}
else {
    Error(5);
    return false;
}
}

```

```

bool Name(ifstream &infile, char s) {
    bool isprm=false;
    for (int i =0; i < 3; i++)
    {
        if(isalpha(s) && infile >> s) {
            cout << s;
        }
        else {
            Error(3);
            return false;
        }
    }
    if(s=='=' && infile >> s) {
        cout << s;
        isprm=Parameter(infile, s);
        if(isprm) {
            return true;
        }
    }
    else {
        Error(4);
        return false;
    }
}

bool Parameter_list(ifstream &infile, bool q) {
    char s;
    bool list = false;
    if(infile >> s)
    {
        cout << s;
        if (isalpha(s)) {
            list = Name(infile, s);
        }
    }
}

```

```

        if(infile >>s)
            cout << s;
        Parameter_list(infile, true);
    }
    else if (s == ',') Parameter_list(infile, true);
    else Error(2);
    infile >> s;
}
else if (q == false) Error(0);
return list;
}
void Error(short k) {
    cout << endl << "err#" << k;
    switch (k) {
        case 0:
            cout << "!-Пустая входная строка" << endl;
            break;
        case 1:
            cout << "!-Лишние символы во входной строке" << endl;
            break;
        case 2:
            cout << "!-Недопустимый начальный символ" << endl;
            break;
        case 3:
            cout << "!-Не имя" << endl;
            break;
        case 4:
            cout << "!-Нет знака =" << endl;
            break;
        case 5:
            cout << "!-Нет закрывающей скобки" << endl;
            break;
        case 6:
            cout << "!-Не цифра" << endl;
            break;
        default:
            cout << "!-... ";
            break;
    };
}
}

```


ПРИЛОЖЕНИЕ Б. Тестовые случаи

Таблица 1 - Результаты тестов.

input	output	True/false
aaa=89	Это список параметров!	True
aaa=908	!-Не цифра Это НЕ список параметров!	True
dfg=67,agh=90	Это список параметров!	True
adf=67,sjd=(dgh=90,dfg=(jsv=90))	Это список параметров!	True
adf=67,sjd=(dgh=90,dfg=(jsv=90)),dgh=(jdf=00,dgh=90)	Это список параметров!	True