

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

ОТЧЕТ
по лабораторной работе №1
по дисциплине «Алгоритмы и структуры данных»
Тема: Рекурсия

Студент гр. 7383

Корякин М.П.

Преподаватель

Размочева Н.В.

Санкт-Петербург

2018

ОГЛАВЛЕНИЕ

Цель работы.....	3
Реализация задачи.....	4
Тестирование.....	6
Вывод.....	7
Приложение А. Тестовые случаи.....	8
Приложение Б. Исходный код программы	9

1. ЦЕЛЬ РАБОТЫ

Цель работы: познакомиться с основными понятиями и приемами рекурсивного программирования, получить навыки программирования рекурсивных процедур и функций на языке программирования C++.

Формулировка задачи: Формулировка задачи: Разработать программу, которая по заданному простому_логическому выражению: TRUE | FALSE | простой_идентификатор | NOT простое_логическое | (простое_логическое знак_операции простое_логическое), не содержащему вхождений простых идентификаторов, вычисляет значение этого выражения. Знак-операции::= AND|OR.

2. РЕАЛИЗАЦИЯ ЗАДАЧИ

В функции `main` выводится меню, которое спрашивает откуда необходимо считать логическое выражение. В зависимости от введенного числа оператор `switch` начнет выполнять серию команд. После проведения этих операций. Функция `main` получит ответ на логическое выражение и выведет его в консоль, либо выведет строку «Вы вышли из программы». На этом программа закончится

Переменные, используемые в функции `main`:

- `bool res` — используется для записи результата логического выражения
- `bool cur` — промежуточный ответ логического выражения, передается в рекурсивную функцию.
- `int is_okey` — Переменная для определения правильности выполнения программы для вывода нужного результата.
- `int flag` — флаг для рекурсивной функции, который показывает какой раз производится вхождение в функцию.
- `int choise` — для определения способа считывания логического выражения (или выхода из программы).
- `stringbuf exp` — буфер под строку, введенную из терминала.
- `filebuf file` — буфер под строку, взятую из файла.
- `unsigned h` – объявленная переменная для запуска функции `check`.
- `bool m` – объявленная переменная для запуска функции `check`.
- `string s` - объявленная переменная для запуска функции `check`.

Функция `oper` определяет знак операции между двумя логическими выражениями, и возвращает в зависимости от этого 1, если знак операции `or`, и 2, если знак операции `and`

Переменные, используемые в функции `oper`:

- `char str` — хранит считанную строку(знак операции).

Функция `write` определяет считанное логическое выражение и возвращает его.

Переменные, используемые в функции `write`:

- `char str` — хранит считанную строку(логическое выражение).

Функция `res` определяет результат полного логического выражения, по ходу вызывая такие функции, как `write` и `oper`. Эта функция работает по принципу нисходящей рекурсии.

Переменные, используемые в функции `write`:

- `bool cur` — промежуточный ответ логического выражения,
- `int flag` — флаг для рекурсивной функции, который показывает какой раз производится вхождение в функцию.
- `int op` — переменная применяет значение знака операции, если `op=1`, то значит это операция `or`. Соответствующий комментарий оставлен в коде программы.

Функция `check` используется в качестве синтаксического анализатора, и возвращает на выходе либо `“true”`, либо `“false”`, в зависимости от правильности ввода логического выражения.

Переменные, используемые в функции `check`:

- `string str` – переменная содержит в себе строку с самим логическим выражением.
- `bool flag` – используется для определения простого логического выражения, а также для возвращения в функцию `main`, если введенное логическое выражение верно.
- `unsigned int k` – используется для ведения учета прочтенных символов строки `str`.

3. ТЕСТИРОВАНИЕ

Программа была собрана в компиляторе G++ в OS Linux Ubuntu 18.02.

В ходе тестирования была найдена ошибка, из-за которой ответ на задачу был некорректен. Изначально было принято решения сравнивать строки посимвольно, через обычное сравнение. Так как программа выполнялась неправильно, было рассмотрено решение этой проблемы, использованием функции `strcmp`, для которой была подключена библиотека `<cstring>`

Некорректный случай представлен в табл. 1.

Таблица 1 — Некорректный случай с неправильным ответом.

Входные данные	Результат
1 true and false	false

Корректные тестовые случаи представлены в приложении А.

4. ВЫВОД

В ходе работы были получены навыки рекурсивного программирования на языке C++. В данной задаче использовался принцип нисходящей рекурсии, т.е. которая имела граничные условия. Для решения данной задачи нисходящая рекурсия является одним из самых эффективных инструментов, потому как заранее мы не знаем степень вложенности простого логического выражения.

ПРИЛОЖЕНИЕ А.

ТЕСТОВЫЕ СЛУЧАИ

Таблица 2 — Корректные тестовые случаи

Входные данные	Результат
1 true and true	true
1 not true and true	false
1 not true or not true	false
2 (In file:)true or false	true
2 (In file:)false or false	false

ПРИЛОЖЕНИЕ Б.

ИСХОДНЫЙ КОД ПРОГРАММЫ

Файл main.cpp:

```
#include "func.hpp"

using namespace std;

int main(){
    unsigned int h;
    bool m;
    bool res;//Для записи результата
    bool cur;//Промежуточная переменная для выполнения логической операции
    int is_okey = 0;//Переменная для определения правильности выполнения программы
    int flag=0;//Для ф-ии "rec", отслеживает какой раз я захожу в функцию
    int choise = 0; //Для выбора потока данных
    string s2;
    cout << "Введите 1, если хотите считать выражение с терминала" << endl;
    cout << "Введите 2, если хотите считать выражение из файла" << endl;
    cout << "Введите 3, если хотите выйти из программы" << endl;
    cin >> choise;
    cin.ignore();
    switch (choise){
        case 1:
        {
            cout << "Введите логическое выражение: ";
            //is_okey=1;
            stringbuf exp;//буфер под строку, введенную из терминала
            string temp_str;
            getline(cin, temp_str);
            if(check(temp_str, m, h)>0){
                is_okey=1;
                istream is_str(&exp);
                exp.str(temp_str);
                res = rec(is_str, cur, flag);
            }
        }
        break;
        case 2:
        {
            filebuf file;//буфер под строку из файла
            file.open("input.txt", ios::in);
            istream files(&file);
            string s;
            getline ( files, s );
            cout << s<<endl;
            files.seekg(0);
            if(check(s, m, h)>0){
                is_okey=1;
            }
        }
    }
}
```

```

        res = rec(files, cur, flag);
    }
    file.close();
}
break;
case 3:
{
    cout << "Вы вышли из программы" << endl;
}
break;
default:
{
    cout << "Вы ввели не то значение, попробуйте заново" << endl;
}
break;
}

if(is_okey){
    cout << "Ответ: ";
    if(res)
        cout << "true" << endl;
    else
        cout << "false" << endl;
}
else
    cout << "Логическое выражение введено неправильно" << endl;

return 0;
}

```

Файл func.cpp:

```

#include "func.hpp"
/* for function "oper"
    1 = "or"
    0 = "and"
*/
int oper(istream &files, int op){ //Функция для определения вводимой операции
    char str[10];
    files>>str;
    if (strcmp(str,"and")==0)
        return 0;
    else
        return 1;
}

bool write(istream &files, bool cur){ //Функция для определения вводимого значение
    char str[10];
    if(files>>str){
        if (strcmp(str,"true")==0){
            cur = true;
        }
    }
}

```

```

        if (strcmp(str,"false")==0)
            cur = false;
        if (strcmp(str,"not")==0){
            files>>str;
            if (strcmp(str,"true")==0)
                cur = false;
            if (strcmp(str,"false")==0)
                cur = true;
        }
    }
    return cur;
}

bool rec(istream &files, bool cur, int flag){    //Рекурсивная функция для
определения результата логического выражения
    if (flag==1){
        cur = write(files, cur);
        return cur;
    }
    if (flag==0){
        cur = write(files, cur);
        int op;
        op = oper(files, op);
        flag=1;
        if (op)
            return (cur || rec(files, cur, flag));
        else
            return (cur && rec(files, cur, flag));
    }
}

bool check(string str, bool flag=false, unsigned int k=0){
    if (k<str.length()){
        if(!flag && str.substr(k, str.find(' ', k)-k)=="not")
            flag=check(str, false, k+4);
        else if(!flag && str.substr(k, str.find(' ', k)-k)=="true")
            flag=check(str, true, k+5);
        else if(!flag && str.substr(k, str.find(' ', k)-k)=="false")
            flag=check(str, true, k+6);
        else if(flag && str.substr(k, str.find(' ', k)-k)=="and")
            flag=check(str, false, k+4);
        else if(flag && str.substr(k, str.find(' ', k)-k)=="or")
            flag=check(str, false, k+3);
        else
            return false;
    }
    return flag;
}

```

Файл func.hpp:

```

#include <iostream>
#include <fstream>
#include <istream>

```

```
#include <sstream>
#include <cstring>
#pragma once
using namespace std;
int oper(istream &files, int op);
bool write(istream &files, bool cur);
bool rec(istream &files, bool cur, int flag);
bool check(string str, bool flag, unsigned int k);
```

Файл Makefile:

```
all: run

run: main.o func.o func.hpp
    g++ main.o func.o -o run
main.o: main.cpp func.hpp
    g++ -c main.cpp
func.o: func.cpp func.hpp
    g++ -c func.cpp
clean: rm -rf *.o
```