

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. 9304

Рослова Л.С

Преподаватель

Филатов А.Ю

Санкт-Петербург

2020

Цель работы.

Изучить понятие рекурсии и методы её применения.

Научиться грамотно реализовывать программы с использованием рекурсии в языке C++.

Задание.

Вариант 12. Построить синтаксический анализатор для понятия скобки.

Скобки::= квадратные | круглые | фигурные

Квадратные::= [круглые фигурные] | +

Круглые::= (фигурные квадратные) | -

Фигурные::= {квадратные круглые} | 0

Выполнение работы.

Программа принимает в качестве аргумента текстовый файл, из которого считывается строка. Предварительно строка проверяется, при обнаружении лишних символов или несоответствии количества открывающих скобок к закрывающим — программа выдаст сообщение и завершит свою работу.

Анализ строки начинается с определения первой отпирающей скобы, после чего запускается одна из трех функции(*is_circle*, *is_square*, *is_figure*). Все три функции принимают в качестве аргументов: ссылку на исходную строку *myString*, позицию в строке вызывающей скобы *first* и позицию итератора *second*. Обход строки производится рекурсивно, условием выхода служит необходимая запирающая скоба или их некорректное расположение с последующим предупреждением и завершением программы.

Разработанный программный код см. в приложении А.

Тестирование.

Для проведения тестирования был написан bash-скрипт *./myScript*. Скрипт запускает программу с входными данными из папки *./Tests* и выводит результат в файлы *result*.

Результаты тестирования см. в приложении Б.

Выводы.

В процессе выполнения работы был изучен рекурсивный подход к решению задач. Была реализована программа на языке C++ с использованием рекурсивного алгоритма. Также был реализован *bash-script* помогающий тестировать программу.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: main.cpp

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

/*
    Все 3 функции принимают на вход:
    - Ссылку на строку с нашими данными
    - Позицию открывающей скобы в массиве
    - Ссылку на позицию итератора
*/

void is_circle(std::string&, size_t, size_t&);
void is_square(std::string&, size_t, size_t&);
void is_figured(std::string&, size_t, size_t&);

int main(int argc, char* argv[])
{
    setlocale(LC_ALL, "ru"); // Рашен лангуаге

    if(argc != 2) // Проверка кол-ва аргументов у программы
    {
        std::cout << "Неверное кол-во аргументов!\n";
    }

    std::ifstream test(argv[1]); // Файл на чтение

    if(!test)
    {
        std::cout << "Файл " << argv[1] << " не может быть открыт
на чтение!\n";
    }

    std::string myString;
    std::getline(test, myString); // Получаем в строку myString
данные из файла

/*
    Проверка корректности входных данных
*/

    size_t open = 0;
    size_t close = 0;

    for(int i = 0; i < myString.size(); i++){
        if(!(myString[i] == '(' || myString[i] == '[' ||
myString[i] == '{' ||
myString[i] == ')' || myString[i] == ']' ||
myString[i] == '}') ||
```

```

        myString[i] == '+' || myString[i] == '-' ||
myString[i] == '0'))
    {
        std::cout << "Некорректные символы в строке!" <<
std::endl;
        return 0;
    }
    if(myString[i] == '(' || myString[i] == '[' || myString[i]
== '{'){
        open++;
    }
    if(myString[i] == ')' || myString[i] == ']' || myString[i]
== '}'){
        close++;
    }
}

if(open != close){
    std::cout << "Кол-во открывающих скобок не совпадает с
закрывающими!" << std::endl;
    return 0;
}

std::cout << myString << " <- Начальная строка\n";

size_t second = 0;

if(myString[second] == '('){
    is_circle(myString, second, second);
}else if(myString[second] == '['){
    is_square(myString, second, second);
}else if(myString[second] == '{'){
    is_figured(myString, second, second);
}else{
    std::cout << "Кривое начало строки!" << std::endl;
    return 0;
}

std::cout << myString << " <-ЭТО СКОБКИ!" << std::endl;

return 0;
}

void is_square(std::string &s, size_t first, size_t &second)
{
    std::cout << "Открылись квадратные скобки!\n";
    //std::cout << "Открылись квадратные скобки!\n\tfirst = " <<
s[first] << "\n\tsecond = " << s[second+1] << '\n';

    second++;

    for(int i = 0; i < second; i++){
        std::cout << s[i];
    }
    std::cout << '\n';
}

```

```

        if(s[second] == '']){
            if(s[first] == '[' && second - first == 1){
                std::cout << "Не хватает -" << std::endl;
                exit(1);
            }
            std::cout << "\tЗакрылись скобки " << s[first] << ' ' <<
s[second] << '\n';
            second++;
            return;
        }

        if(s[first] == '+' && s[second] == '-'){
            std::cout << "\tЗакрылись скобки " << s[first] << ' ' <<
s[second] << '\n';
            second++;
            return;
        }else{
            if(s[second] == '-'){
                is_circle(s, second, second);
            }else{
                if(s[second] == '('){
                    is_circle(s, second, second);
                    if(s[second] == '{'){
                        is_figured(s, second, second);
                    }else if(s[first] == '+'){
                        std::cout << "\tЗакрылись скобки " << s[first]
<< ' ' << "()" << '\n';
                        return;
                    }else{
                        std::cout << "Не хватает {" << std::endl;
                        exit(1);
                    }
                }else{
                    std::cout << "Не хватает ( или -" << std::endl;
                    exit(1);
                }
            }
            second--;
            is_square(s, first, second);
        }
    }
}

```

```

void is_circle(std::string &s, size_t first, size_t &second)
{
    std::cout << "Открылись круглые скобки!\n";

    second++;

    for(int i = 0; i < second; i++){
        std::cout << s[i];
    }
    std::cout << '\n';

    if(s[second] == '']){
        if(s[first] == '(' && second - first == 1){

```

```

        std::cout << "Не хватает 0" << std::endl;
        exit(1);
    }
    std::cout << "\tЗакрылись скобки " << s[first] << ' ' <<
s[second] << '\n';
    second++;
    return;
}

if(s[first] == '-' && s[second] == '0'){
    std::cout << "\tЗакрылись скобки " << s[first] << ' ' <<
s[second] << '\n';
    second++;
    return;
}else{
    if(s[second] == '0'){
        is_figured(s, second, second);
    }else{
        if(s[second] == '{'){
            is_figured(s, second, second);
            if(s[second] == '['){
                is_square(s, second, second);
            }else if(s[first] == '-'){
                std::cout << "\tЗакрылись скобки " << s[first]
<< ' ' << "{}" << '\n';
                return;
            }else{
                std::cout << "Не хватает [" << std::endl;
                exit(1);
            }
        }else{
            std::cout << "Не хватает { или 0" << std::endl;
            exit(1);
        }
    }
    second--;
    is_circle(s, first, second);
}
}

```

```

void is_figured(std::string &s, size_t first, size_t &second)
{
    std::cout << "Открылись фигурные скобки!\n";

    second++;

    for(int i = 0; i < second; i++){
        std::cout << s[i];
    }
    std::cout << '\n';

    if(s[second] == '}'){
        if(s[first] == '{' && second - first == 1){
            std::cout << "Не хватает +" << std::endl;
            exit(1);
        }
    }
}

```

```

        std::cout << "\tЗакрылись скобки " << s[first] << ' ' <<
s[second] << '\n';
        second++;
        return;
    }

    if(s[first] == '0' && s[second] == '+'){
        std::cout << "\tЗакрылись скобки " << s[first] << ' ' <<
s[second] << '\n';
        second++;
        return;
    }else{
        if(s[second] == '+'){
            is_square(s, second, second);
        }else{
            if(s[second] == '['){
                is_square(s, second, second);
            }if(s[second] == '('){
                is_circle(s, second, second);
            }else if(s[first] == '0'){
                std::cout << "\tЗакрылись скобки " << s[first]
<< ' ' << "[" << '\n';
                return;
            }else{
                std::cout << "Не хватает (" << std::endl;
                exit(1);
            }
        }else{
            std::cout << "Не хватает [ или +" << std::endl;
            exit(1);
        }
    }
    second--;
    is_figured(s, first, second);
}
}

```

Название файла: Makefile

```

all: lab1

lab1: ./Source/main.cc
g++ ./Source/main.cc -o lab1

clean:
rm lab1

clean2:
rm ./*.txt

test:
chmod -x ./myScript

```


Название файла: ./myScript

```
#!/bin/bash
```

```
arg1=$(cat Tests/test1.txt)
echo "Test 1:"
echo "argument = $arg1"
touch result1.txt
./lab1 ./Tests/test1.txt > result1.txt
cat result1.txt
```

```
echo -e "\nTest 2:"
arg2=$(cat Tests/test2.txt)
echo "argument = $arg2"
touch result2.txt
./lab1 ./Tests/test2.txt > result2.txt
cat result2.txt
```

```
echo -e "\nTest 3:"
arg3=$(cat Tests/test3.txt)
echo "argument = $arg3"
touch result3.txt
./lab1 ./Tests/test3.txt > result3.txt
cat result3.txt
```

```
echo -e "\nTest 4:"
arg4=$(cat Tests/test4.txt)
echo "argument = $arg4"
touch result4.txt
./lab1 ./Tests/test4.txt > result4.txt
cat result4.txt
```

```
echo -e "\nTest 5:"
arg5=$(cat Tests/test5.txt)
echo "argument = $arg5"
touch result5.txt
./lab1 ./Tests/test5.txt > result5.txt
cat result5.txt
```

ПРИЛОЖЕНИЕ Б

ТЕСТИРОВАНИЕ ПРОГРАММЫ

Результаты тестирования представлены в табл.1

Таблица 1 – Результаты тестирования

№ п/п	Входные данные	Выходные данные	Комментарии
1.	1 ({+-}[-0])	({+-}[-0]) ← ЭТО СКОБКИ!	Корректная работа программы
2.	([{}])	([Не хватает (или -	После [должно идти (или -
3.	({}[])	({ Не хватает +	После { должен идти +
4.	Хоба! А тут нету скобок!	Некорректные символы в строке!	Отличные от скобок символы приводят к завершению программы
5.	1 ({+(0+)}[-0])	({+(0+)}[-0]) <- ЭТО СКОБКИ!	Корректная работа программы