

**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

\_\_\_\_\_

Краев Д.В.

Преподаватель

\_\_\_\_\_

Фиалковский М.С.

Санкт-Петербург

2020

### **Цель работы.**

Узнать, что такое рекурсия, и научиться использовать ее на практике.

### **Задание.**

*Построить синтаксический анализатор для параметризованного понятия скобки( $T$ ), где  $T$ —заданное конечное множество, а круглые скобки «(»и «)» не являются терминальными символами, а отражают зависимость определяемого понятия от параметра  $T$ .*

*скобки( $T$ )::=элемент( $T$ )|список(скобки( $T$ ))*

*список( $E$ )::= $N$ |[ряд( $E$ )]*

*ряд( $E$ )::=элемент( $E$ )|элемент( $E$ )ряд( $E$ )*

### **Выполнение работы.**

#### **Функции**

Для выполнения работы были созданы 4 функции: isBrackets, isArray, isList и isElem.

#### **1)isBrackets**

IsBrackets принимает на вход экземпляр класса string, содержащий строку, которую нужно проверить. Функция при помощи совершает обход по данной строке, проверяет каждый символ с помощью функций isList и isElem. Если хотя бы 1 символ не пройдет проверку, то функция isBrackets вернут строку «incorrect», в ином случае строку «correct».

#### **2)isList**

IsList принимает на вход ссылку на итератор класса string. Если он находится на символе  $N$ , то функция возвращает true. Если итератор находится на символе  $[$ , то итератор переходит на следующий символ в строке и функция возвращает возвращаемое значение функции isArray, которой подали на вход итератор. Если итератор находится не на символе  $N$  и не на символе  $[$ , то функция возвращает false.

### **3)isArray**

isArray принимает на вход ссылку на итератор класса string. Функция совершает обход по строке до тех пор пока не встретит символ ] или символ конца строки. Каждый раз она проверяет с помощью функций isElem и isList. Если обе функции возвращают false, то функция isArray возвращает false, в ином случае продолжает обход. Если функция встретила символ конца строки, то она возвращает false. Если она встретила символ ] , и до этого проверила как минимум 1 символ, то она возвращает true.

### **4)isElem**

isElem примет на вход ссылку на итератор. И проверяет символ, на котором находится итератор, на вхождение в множество элементов. Если он не находится в данном множестве.

### **Работа программы**

Программа принимает на вход 1 аргумент командной строки, содержимое которого является строкой, которую нужно проверить. В функции main происходит вызов функции isBrackets, на вход которой подается 1 аргумент командной строки. Далее происходит вывод возвращаемого значения этой функции correct, если строка соответствует всем условиям, в ином случае incorrect.

### **Тестирование**

Тестирование проводится с помощью скрипта, написанном на языке Python. Скрипт использует библиотеки unittest и subprocess. Входные данные для скрипта берутся из файлов correct.txt и incorrect.txt, находящиеся Библиотека subprocess нужна для запуска программы с нужными входными данными, а библиотека unittest для проведения тестирования.

Запустить программу в командной строке можно при помощи команды «./lab <строка для проверки>». Скрипт можно запустить при помощи команды «make run\_tests». Результаты тестирования можно посмотреть в приложении В

## **Выводы**

Изучено понятие рекурсия. Написан синтаксически анализатор для понятия скобки( $T$ ) с помощью рекурсии. Использование рекурсии полностью оправдано, объекты типов, заданных условием задачи, могут содержать объекты такого же типа. Например:  $ряд(E) ::= элемент(E) | элемент(E)ряд(E)$ .

## ПРИЛОЖЕНИЕ А

### ИСХОДНЫЙ КОД ПРОГРАММЫ

**Файл: main.cpp**

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

#define ELEMS "<>{}()"

std::string isBrackets(const std::string &str);
bool isArray(std::string::const_iterator &it);
bool isList(std::string::const_iterator &it);
bool isElem(std::string::const_iterator &it);

std::string isBrackets(const std::string &str){
    if(str.size() == 0){
        return "incorrect";
    }
    for(std::string::const_iterator it = str.begin(); it != str.end(); it++){
        if(!isElem(it) && !isList(it)){
            return "incorrect";
        }
    }
    return "correct";
}

bool isArray(std::string::const_iterator &it){
```

```

int n = 0;
while(*it != '']){
    if(isElem(it) || isList(it)){
        it++;
    }else{
        return false;
    }
    if(*it == '\0'){
        return false;
    }
    n++;
}

if(n>=1){
    return true;
}else{
    return false;
}
}

bool isList(std::string::const_iterator &it){
    if(*it == 'N'){
        return true;
    }
    if(*it == '['){
        it++;
        return isArray(it);
    }
    return false;
}

```

```

bool isElem(std::string::const_iterator &it){
    std::string elems(ELEMS);
    for(int i = 0;i < elems.size();i++){
        if(*it == elems[i])
            return true;
    }
    return false;
}

```

```

int main(int argc, char* argv[]){
    std::cout << isBrackets(argv[1]) << '\n';
    return 0;
}

```





## ПРИЛОЖЕНИЕ В

### ТЕСТИРОВАНИЕ

Результаты тестирования представлены в таблице Б.1

Таблица Б.1 — Результаты тестирования

№ п/п	Входные данные	Выходные данные
1.	[[[]]]	incorrect
2.	<><>{}{}{}{}{}{}))((((	correct
3.	sofjsv3fj2230kcpwj2df2pof	incorrect
4.	<><><<<{[>]}	corect
5.	<><><<<{[>]}1	incorrect
6.	N	correct
7.	[[[[[[[[]]]]]]]	incorrect
8.	[[[<><>><{}]]]	correct
9.	[NNNNN]())({})<><>	correct
10.	Nadasdsad	incorrect