МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

НИЖЕГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ

УНИВЕРСИТЕТ им. Р.Е.АЛЕКСЕЕВА

Институт радиоэлектроники и информационных технологий

Кафедра информатики и систем управления

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Лабораторная работа №4\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(наименование темы проекта или работы)

ОТЧЁТ

По лабораторной работе

по дисциплине

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Алгоритмы и структуры данных\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(наименование дисциплины)

РУКОВОДИТЕЛЬ:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись) (фамилия, и.,о.)

СТУДЕНТ:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись) (фамилия, и.,о.)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(шифр группы)

Работа защищена «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_

С оценкой \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Нижний Новгород 2022

**Задание:**

Дано N-дерево. Найти поддерево, не включающее ни одну из заданных вершин.

**Список функций и структур данных с описанием:**

Структура NTree - структура данных содержащая значение корня дерева и указатели на сыновей и их количество

insertNode - функция создания и заполнения детей корня дерева

CreateNTree - Функция создания дерева

Delete - Функция удаления дерева

**Программный код:**

#include<iostream>

struct NTree

{

int data;

int N;

struct NTree \*\*children;

};

NTree \*insertNode(NTree \*parent, int num){

int N=num;

parent->children = new NTree\*[N];

for (int i=0; i<N; i++){

int buf=0;

std::cout<<"Введите "<< i+1 << "-го сына вершины "<<parent->data<<std::endl;

while(!(std::cin>>buf) || (std::cin.peek()!='\n')){

std::cin.clear();

while(std::cin.get()!='\n');

std::cout<<"Error! Введите число заново"<<std::endl;

}

parent->children[i] = new NTree;

parent->children[i]->data =buf;

if (buf==0){

std::cout<<"Ветвь окончена"<<std::endl;

}

if (parent->children[i]->data!=0){

parent->children[i]->N = N;

parent->children[i] = insertNode(parent->children[i],N);

}

}

return parent;

}

NTree \*CreateNTree(int num){

NTree \*source= new NTree;

source->N = num;

int first=0;

std::cout<<"Для ввода вершины введите его значение, потом его сыновей слева направо"<<std::endl;

std::cout<<"Введите первую вершину дерева"<<std::endl;

while(!(std::cin>>first) || (std::cin.peek()!='\n')){

std::cin.clear();

while(std::cin.get()!='\n');

std::cout<<"Error! Введите число заново";

}

source->data = first;

if(num<=0){

std::cout<<"error!"<<std::endl;

return source;

};

source = insertNode(source,num);

return source;

}

NTree \*Delete(NTree \*curNode,int num,int ch){

int N = num;

if(curNode->data!=ch){

std::cout<<curNode->data<<";";

for(int i=0;i<N;i++){

if(curNode->children[i]->data!=0){

Delete(curNode->children[i],num,ch);

}

}

}

return curNode;

}

int main()

{

char logo[] =

"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n"

"\* Nizhny Novgorog State Technical University \*\n"

"\* Study work number 4. Task number 1. \*\n"

"\* Performed student 21-IVT-2 Kitov A. A. \*\n"

"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n";

std::cout<< logo <<std::endl;

setlocale(LC\_ALL, "Russian");

int cg=0;

std::cout<<"Введите колчество n веток корня: ";

while(!(std::cin>>cg) || (std::cin.peek()!='\n')){

std::cin.clear();

while(std::cin.get()!='\n');

std::cout<<"Error! Введите число заново";

}

NTree \*tree = CreateNTree(cg);

int chislo=0;

std::cout<<"Введите исключаемое число: ";

while(!(std::cin>>chislo) || (std::cin.peek()!='\n')){

std::cin.clear();

while(std::cin.get()!='\n');

std::cout<<"Error! Введите число заново";

}

tree=Delete(tree,cg,chislo);

std::cout<<std::endl;

return 0;

}

**Результаты работы программы:**









