Министерство образования Республики Беларусь

Учреждение образования

“Брестский государственный университет”

Кафедра ИИТ

Лабораторная работа №14

По дисциплине “Основы алгоритмизации и программирования”

Тема: “Динамические структуры: списки и деревья”

Вариант №11

Выполнил:

Студент 1ого курса

Группы ПО 7

Лобан К.Ю.

Проверила:

Войцехович О.Ю.

Брест 2021

**Цель работы**: Приобретение навыков работы с динамической памятью и указателями на С/C++. Изучение принципов работы с динамическими структурами данных: списками и деревьями.

**Задание**: Написать программу, обеспечивающую работу с однонаправленным нециклическим списком: добавление/удаление элементов в голову, просмотр списка, инверсию списка.

**Код программы**:

#include <iostream>

using namespace std;

struct list

{

int index;

int value;

list\* next = NULL;

};

void add(list\*& head);

list\* del(list\* head);

void display(list\* head);

void reverse(list\*& head);

int main()

{

setlocale(LC\_ALL, "ru");

list\* core = NULL;

while (true)

{

cout << "Меню: " << endl;

cout << "1. Добавить" << endl;

cout << "2. Удалить" << endl;

cout << "3. Отобразить" << endl;

cout << "4. Перевернуть" << endl;

cout << "0. Выход" << endl;

cout << "Ваш выбор: ";

int command = -1;

cin >> command;

switch (command)

{

case 1:

add(core);

break;

case 2:

core = del(core);

break;

case 3:

display(core);

break;

case 4:

reverse(core);

break;

case 0:

exit(0);

break;

default:

cout << "\nНет такой команды" << endl;

break;

}

system("pause");

system("cls");

}

}

void add(list\*& head)

{

list\* start = head;

cout << "Значение: ";

int value = -1;

cin >> value;

cout << endl;

if (head == NULL)

{

head = new list;

head->value = value;

head->index = 0;

return;

}

while (head->next != NULL)

head = head->next;

head->next = new list;

head->next->value = value;

head->next->index = head->index + 1;

head = start;

return;

}

list\* del(list\* head)

{

list\* start = head;

if (head == NULL)

{

cout << "Список пуст" << endl;

return NULL;

}

cout << "Значение: ";

int value = -1;

cin >> value;

cout << endl;

list\* prev = NULL;

do

{

if (head->value == value)

{

cout << "Индекс искомого элемента: " << head->index << endl;

if (prev == NULL)

{

list\* new\_head = head->next;

delete head;

return new\_head;

}

else

{

prev->next = head->next;

}

return start;

}

prev = head;

head = head->next;

} while (head != NULL);

cout << "Такой элемент не найден" << endl;

return start;

}

void display(list\* head)

{

while (head != NULL)

{

cout << head->value << " ";

head = head->next;

}

cout << endl;

return;

}

void reverse(list\*& head)

{

if (head == NULL)

{

cout << "Список пуст" << endl;

return;

}

list\* new\_list = NULL;

while (head != NULL)

{

list\* prev\_item = new list;

prev\_item->value = head->value;

prev\_item->index = head->index;

prev\_item->next = new\_list;

new\_list = prev\_item;

head = head->next;

}

head = new\_list;

return;

}

**Вывод**: Приобрел навыки работы с динамической памятью и указателями на С/C++. Изучил принципов работы с динамическими структурами данных: списками и деревьями.