Министерство образования РФ

Пермский национальный исследовательский политехнический университет

Кафедра ИТАС

ЛАБОРАТОРНАЯ РАБОТА № 11

ПО ОСНОВАМ АЛГОРИТМИЗАЦИИ ЗА I СЕМЕСТР

Вариант 15

Выполнил студент:

Сташков Арсений Дмитриевич

Группа РИС-20-1бз

Шифр 20-ЭТФ-644

Проверила:

Доцент кафедры ИТАС

Полякова Ольга Андреевна

ПЕРМЬ 2020

# Лабораторная работа №11 "Информационные динамические структуры"

**Вариант №15**

**Цель:** Знакомство с динамическими информационными структурами на примере одно- и двунаправленных списков.

**Задача:** Написать программу, в которой создаются динамические структуры и выполнить их обработку в соответствии со своим вариантом.

15.

Записи в линейном списке содержат ключевое поле типа \*char(строка символов). Сформировать двунаправленный список. Удалить К элементов из конца списка. Добавить элемент после элемента с заданным ключом.

**Решение:**

#include <iostream>

using namespace std;

// Заготовка двунаправленного списка

struct Data2

{

int a; // данные

};

struct list

{

Data2 d;

list\* prev; // указатель на предшествующий элемент

list\* next; // указатель на последующий элемент

};

// Вывод двунаправленного списка на экран

void PrintDoublyLinkedList(list\* Start)

{

setlocale(LC\_ALL, "Rus");

list\* p = Start;

cout << "\nНаш двунаправленный список:\n\n";

while (p)

{

cout << p->d.a << " ";

p = p->next;

}

cout << "\n";

}

// Удаление элементов списка с конца

void Clear(list\*\* Start)

{

if (\*Start == 0)

{

return;

}

list\* p = \*Start;

list\* t;

int x;

cout << "\nСколько значений Вы хотели бы удалить?\n";

cin >> x;

while (x > 10)

{

cout << "\nНельзя ввести значение, большее, чем 10!\n";

cin >> x;

}

// Пробегаем список до нужного элемента

for (int i = 0; i < 10-x; i++)

{

t = p;

p = p->next;

}

// Начиная с нужного элемента удаляем элементы до конца списка

for (int i = 0; i < x; i++)

{

t = p;

p = p->next;

t->d.a = NULL;

}

PrintDoublyLinkedList(\*Start);

}

// Наполнение двунаправленного списка данными

void DoublyLinkedList()

{

list\* Start = NULL; // Начало списка

list\* End = NULL; // Конец списка

// Создаём первый элемент

list\* t = new list;

t->d.a = rand() % 10;

t->prev = NULL;

t->next = NULL;

// Настроим на него оба указателя

Start = t;

End = t;

// Создаем еще 9 элементов

int x = 1;

while (x < 10)

{

t->next = new list;

list\* p = t;

t = t->next;

t->prev = p;

t->d.a = rand() % 10;

t->next = NULL;

End = t;

x++;

}

PrintDoublyLinkedList(Start);

Clear(&Start);

}

int main()

{

DoublyLinkedList();

return 0;

}

