Министерство образования и науки Российской Федерации  
Федеральное‌ ‌государственное‌ ‌бюджетное‌ ‌образовательное‌ ‌учреждение‌

высшего‌ ‌образования‌

**«Пермский национальный исследовательский  
политехнический университет»**

Кафедра «Информационные технологии и автоматизированные системы»

**О Т Ч Ё Т**

**по лабораторной работе №11**

Дисциплина: основы алгоритмизации и программирования

Тема: Информационные динамические структуры

Выполнил работу

студент группы ПРТ-21-1б

Пикулев М.А.

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

Проверила

Доцент кафедры ИТАС

к.т.н. Полякова О. А.

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

Пермь 2022

**Цель работы**

Знакомство с динамическими информационными

структурами на примере одно- и двунаправленных списков.

**Постановка задачи**

(Вариант 1)

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

**Анализ задачи**

Записи в линейном списке содержат ключевое поле типа int. Сформировать однонаправленный список. Удалить из него элемент с заданным номером, добавить элемент с заданным номером;

1. В ходе работы были использованы односвязный список и функции.

**Решение**

#include <iostream>

using namespace std;

struct Node

{

int x;

Node\* Next;

};

typedef Node\* PNode;

void Add(int data, PNode& Head, PNode& LastNode)

{

PNode Temp;

if (Head == NULL)

{

Head = new Node;

LastNode = Head;

Head->Next = NULL;

}

else

{

Temp = new Node;

LastNode->Next = Temp;

LastNode = Temp;

LastNode->Next = NULL;

}

LastNode->x = data;

}

void Show(PNode Head)

{

while (Head != NULL)

{

cout << Head->x << " ";

Head = Head->Next;

}

}

void DeleteNode(PNode& Head, PNode OldNode)

{

PNode q = Head;

if (Head == OldNode)

Head = OldNode->Next;

else

{

while (q && q->Next != OldNode)

q = q->Next;

if (q == NULL) return;

q->Next = OldNode->Next;

}

delete OldNode;

}

void AddAfter(int data, PNode p, PNode NewNode)

{

NewNode->Next = p->Next;

NewNode->x = data;

p->Next = NewNode;

}

void main()

{

PNode Head, LastNode;

int n;

int k;

Head = NULL;

LastNode = NULL;

cout << "vvedite skolko elemetov v spiske\n";

cin >> k;

cout << "vvedite spisok \n";

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

{

cin >> n;

Add(n, Head, LastNode);

}

cout << "spisok: ";

Show(Head);

cout << endl;

cout << "vvedite nomer elementa kotorii udalit\n";

int x = -1;

while (x > k || x <= 0)

{

cout << "vvedite nomer ot 1 do " << k << endl;

cin >> x;

}

PNode q = Head;

for (int i = 0; i < x - 1; i++)

q = q->Next;

DeleteNode(Head, q);

Show(Head);

cout << "\nvvedite pod kakim nomerom dobavit element\n";

int y = -1;

while (y > k || y <= 0)

{

cout << "vvedite nomer ot 1 do " << k << endl;

cin >> y;

}

PNode NewNode = new Node;

int g;

cout << "vvedite znachenie elementa\n";

cin >> g;

q = Head;

if (y - 2 < 0)

{

NewNode->Next = Head;

NewNode->x = g;

Head = NewNode;

}

else

{

for (int i = 0; i < y - 2; i++)

q = q->Next;

AddAfter(g, q, NewNode);

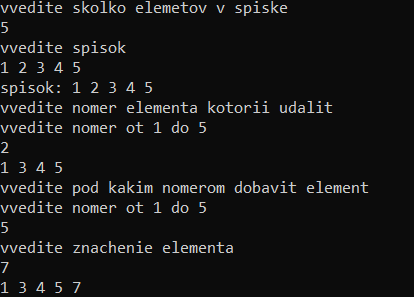
}

Show(Head);

DeleteNode(Head, q);

}

**Скриншоты результата работы**

****