Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Вятский государственный университет»

Колледж ВятГУ

**ОТЧЕТ**

**ПО ДОМАШНЕЙ КОНТРОЛЬНОЙ РАБОТЕ №6**

**«Реализация элементарных структур данных на основе статической памяти»**

**ПО МДК 05.02 РАЗРАБОТКА КОДА ИНФОРМАЦИОННЫХ СИСТЕМ**

Выполнил: студент учебной группы

ИСПк-202-52-00

Домрачев Артём Андреевич

Преподаватель:

Сергеева Елизавета Григорьевна

Киров

2023

**Цель работы**

Изучение принципов работы с базовыми структурами данных, получение навыков организации case-меню.

**Описание алгоритма**

Двусвязный список представляет собой структуру данных, где элементы, называемые узлами, содержат полезную информацию, а также указатели на предыдущий и следующий узлы. Это позволяет двигаться по списку в обоих направлениях и иметь доступ к предыдущему и следующему элементам. Таким образом, двусвязный список обеспечивает более гибкую навигацию и операции с данными, чем односвязный список.

**Код программы**

**uses** Crt;

**const**

MAX\_NODES = 100;

**type**

TNode = **record**

data: integer;

next: integer;

prev: integer;

**end**;

**var**

head, tail: integer;

nodeArray: **array** [1..MAX\_NODES] **of** TNode;

nodeCounter: integer;

**procedure** InsertNode(**var** head, tail: integer; data: integer);

**var**

node: integer;

**begin**

Inc(nodeCounter);

node := nodeCounter;

nodeArray[node].data := data;

nodeArray[node].next := 0;

nodeArray[node].prev := 0;

**if** head = 0 **then**

**begin**

head := node;

tail := node;

**end**

**else**

**begin**

nodeArray[tail].next := node;

nodeArray[node].prev := tail;

tail := node;

**end**;

**end**;

**procedure** DeleteNode(**var** head, tail: integer; data: integer);

**var**

node, prevNode, nextNode: integer;

**begin**

node := head;

**while** (node <> 0) **and** (nodeArray[node].data <> data) **do**

**begin**

node := nodeArray[node].next;

**end**;

**if** node <> 0 **then**

**begin**

prevNode := nodeArray[node].prev;

nextNode := nodeArray[node].next;

**if** prevNode <> 0 **then**

**begin**

nodeArray[prevNode].next := nextNode;

**end**

**else**

**begin**

head := nextNode;

**end**;

**if** nextNode <> 0 **then**

**begin**

nodeArray[nextNode].prev := prevNode;

**end**

**else**

**begin**

tail := prevNode;

**end**;

Dec(nodeCounter);

**end**;

**end**;

**procedure** PrintList(head: integer);

**var**

node: integer;

**begin**

node := head;

**while** node <> 0 **do**

**begin**

Write(nodeArray[node].data, ' ');

node := nodeArray[node].next;

**end**;

Writeln;

**end**;

**var**

choice: integer;

data: integer;

**begin**

head := 0;

tail := 0;

nodeCounter := 0;

**repeat**

TextColor(red);

Writeln('1. Создать элемент списка');

Writeln('2. Удалить элемент из списка');

Writeln('3. Просмотреть список');

Writeln('4. Выход из меню');

Readln(choice);

**case** choice **of**

1:

**begin**

Write('Введите число: ');

Readln(data);

InsertNode(head, tail, data);

**end**;

2:

**begin**

Write('Введите число: ');

Readln(data);

DeleteNode(head, tail, data);

**end**;

3:

**begin**

Writeln('Список элементов:');

PrintList(head);

**end**;

**end**;

**until** (choice = 4);

**end**.

**Результат выполнения программы**

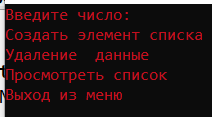
****

Рисунок 1 – Результат выполнения программы

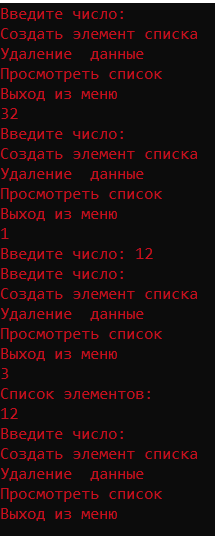


Рисунок 2 – Результат выполнения программы

**Вывод**

В ходе выполнения задания была реализована структура данных двусвязный список с использованием статической памяти. Было создано меню для добавления элементов в список, удаления элементов из списка и просмотра списка. Работа программы успешно завершена, и структура данных успешно протестирована на заданных операциях.