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»**

Выполнил: студент учебной группы

ИСПк-202-52-00

Порубов Кирилл Александрович

Преподаватель:

Сергеева Елизавета Григорьевна

Киров

2023

1.Цель работы: изучение принципов работы с базовыми структурами данных, получение навыков организации case-меню.

2.Формулировка задания (с вариантом):

Вариант 10

1) Написать программу для работы со структурой данных "Двусвязный список".  
2) Структура данных должна быть реализована на основе статической памяти.  
3) Работа со структурой должна осуществляться с помощью case-меню. Предусмотреть наглядную визуализацию содержимого структуры.

3. 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**

Writeln('Введите число: ');

Writeln('1. Ввод данных');

Writeln('2. Удалить данные');

Writeln('3. Напечатать список данных');

Writeln('4. Выход из меню');

Readln(choice);

**case** choice **of**

1: **begin**

Write('Введите данные: ');

Readln(data);

InsertNode(head, tail, data);

**end**;

2: **begin**

Write('Введите данные: ');

Readln(data);

DeleteNode(head, tail, data);

**end**;

3: **begin**

Writeln('Список данных:');

PrintList(head);

**end**;

**end**;

**until** (choice = 4);

**end**.

4. Алгоритм построения программы

Эта программа представляет собой реализацию двусвязного списка, в котором каждый элемент содержит данные и ссылки на предыдущий и следующий элементы. Она предоставляет три операции: добавление элемента в список, удаление элемента из списка и печать списка. Каждый элемент списка представляется структурой TNode, а все элементы хранятся в массиве nodeArray. Переменные head и tail содержат ссылки на первый и последний элементы списка, а nodeCounter используется для отслеживания количества элементов в списке. Код программы содержит цикл, который выводит меню с тремя операциями, позволяет пользователю выбрать операцию и взаимодействовать со списком до тех пор, пока пользователь не выберет выход из меню.

5. Результаты выполнения программы



Рисунок 1 - Результаты программы (1)

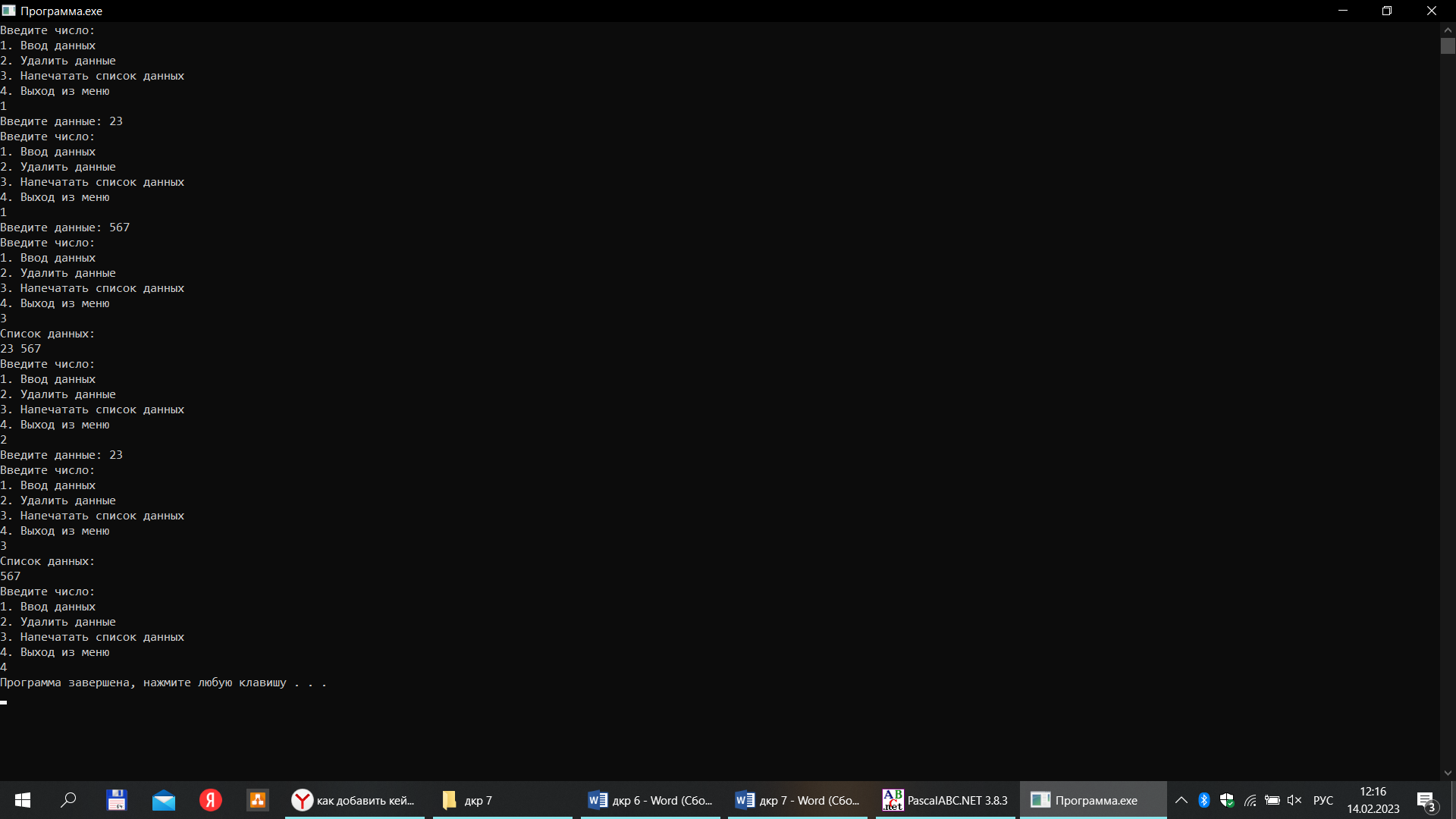


Рисунок 2 - Результаты программы (2)

6. Вывод

В ходе выполнения домашней контрольной работы №6 «реализация элементарных структур данных на основе статической памяти». Изучили принципы работы с базовыми структурами данных, получили навыки организации case-меню.

Данная программа реализует двусвязный список (двунаправленный список). Пользователь может выбрать действие из меню: вставка данных, удаление данных, вывод списка данных или выход из меню. Программа использует статическую память для создания новых узлов в списке. Удаление узла также реализовано с помощью статической памяти. Вывод списка данных происходит циклически, проходя по всем узлам списка и выводя значение каждого узла.

С работой справились в полном объёме. Конечно возникали трудности, однако, их удалось решить. Получили новые знания, умения, опыт. В работе помогали электронные ресурсы, знания, полученные во время занятий благодаря преподавателю и знания, полученные в процессе саморазвития.