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-203-52-00

Корюков Павел Дмитриевич

Преподаватель:

Сергеева Елизавета Григорьевна

Киров

2023

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

2.Формулировка задания:

Вариант 14  
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**

ClrScr;

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

Объявить переменные Head, Tail как тип Integer и инициализировать их значением 0.

Объявить массив NodeArray размером MAX\_NODES (максимальное количество узлов) элементов типа TNode.

Объявить переменную NodeCounter типа Integer и инициализировать ее значением 0.

Создать процедуру InsertNode со следующими параметрами: var Head, Tail: Integer; Data: Integer.

В процедуре InsertNode увеличить NodeCounter на 1, создать новый узел Node и присвоить ему данные Data, установить свойства Next и Prev нового узла на 0.

Проверить, если Head равен 0, то установить Head и Tail равными Node, иначе присвоить Node следующий за Tail и обновить свойства Tail.

Создать процедуру DeleteNode со следующими параметрами: var Head, Tail: Integer; Data: Integer.

В процедуре DeleteNode инициализировать узел Node значением Head и проверить его на равенство 0 и Data.

Если Node не равен 0, то найти предыдущий и следующий узлы, обновить их свойства Next и Prev, обновить Head и Tail, уменьшить NodeCounter.

Создать процедуру PrintList с параметром Head типа Integer.

В процедуре PrintList инициализировать узел Node значением Head и выводить данные узлов по порядку до тех пор, пока Node не станет равным 0.

В основном блоке программы инициализировать переменные Choice и Data типа Integer.

В цикле повторять запрос ввода пользователя и выполнение определенных действий в зависимости от выбора:

при выборе 1 запрашивать данные и вызывать процедуру InsertNode;

при выборе 2 запрашивать данные и вызывать процедуру DeleteNode;

при выборе 3 вызывать процедуру PrintList;

при выборе 4 завершать цикл.

Завершить выполнение программы.

5. Результаты выполнения программы



Рисунок 1 - Результаты программы (1)

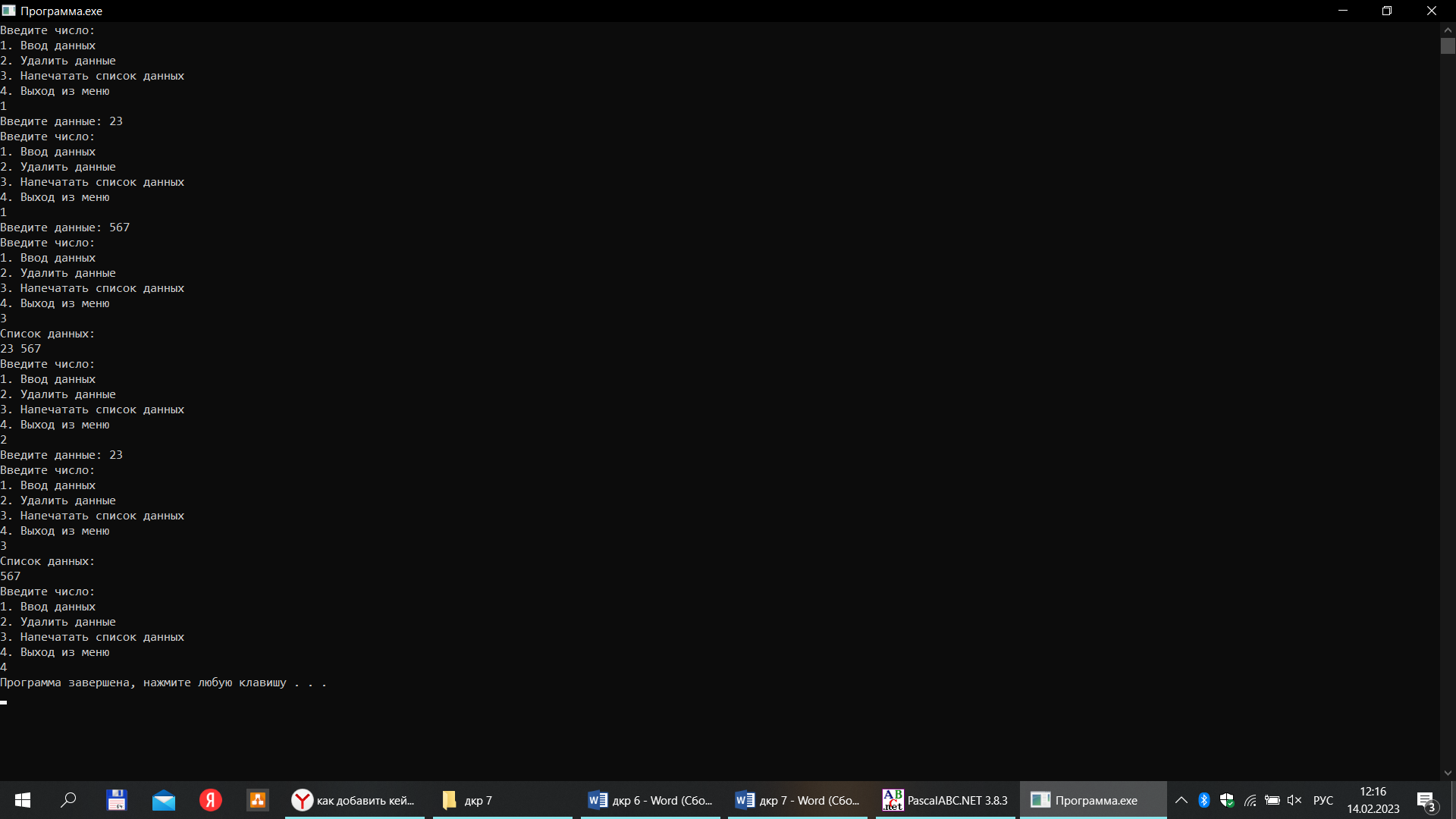


Рисунок 2 - Результаты программы (2)

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.