**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**

Выполнила: студент учебной группы

ИСПк-206-52-00

Чистоусов Егор Романович

Преподаватель:

Сергеева Елизавета Григорьевна

Киров

2024

**Цель работы**

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

**Формулировка задания**

Вариант 1

1. Написать программу для работы со структурой данных «Двусвязный список».
2. Структура данных должна быть реализованная на основе статической памяти.

**Описание алгоритма**

1. В начале определяется тип записи TNode, представляющий узел списка с тремя полями: data (данные узла), next (указатель на следующий узел) и prev (указатель на предыдущий узел).
2. Затем объявляются переменные head и tail для хранения указателей на начало и конец списка, а также массив nodeArray для хранения узлов.
3. Процедура InsertNode используется для добавления нового узла в список. Она создает новый узел, и если список пустой, устанавливает head и tail на этот узел. В противном случае новый узел добавляется в конец списка.
4. Процедура DeleteNode удаляет узел с определенными данными из списка. Она находит узел по данным, перенаправляет указатели соседних узлов и освобождает память, уменьшая счетчик узлов.
5. Процедура PrintList используется для вывода данных всех узлов списка, начиная с указанного узла head.
6. В основной части программы пользователю предлагается выбирать действия: ввод данных, удаление данных, печать списка данных или выход из меню. В зависимости от выбора вызываются соответствующие процедуры.
7. Программа продолжает работу в цикле до тех пор, пока пользователь не выберет выход из меню (вводит 4).
8. Этот код реализует базовые операции с двусвязным списком: добавление, удаление и печать данных.

**Код программы**

**uses** Crt;

**type**

TNode = **record**

data, next, prev: integer;

**end**;

**var**

head, tail: integer;

nodeArray: **array** [1..100] **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);

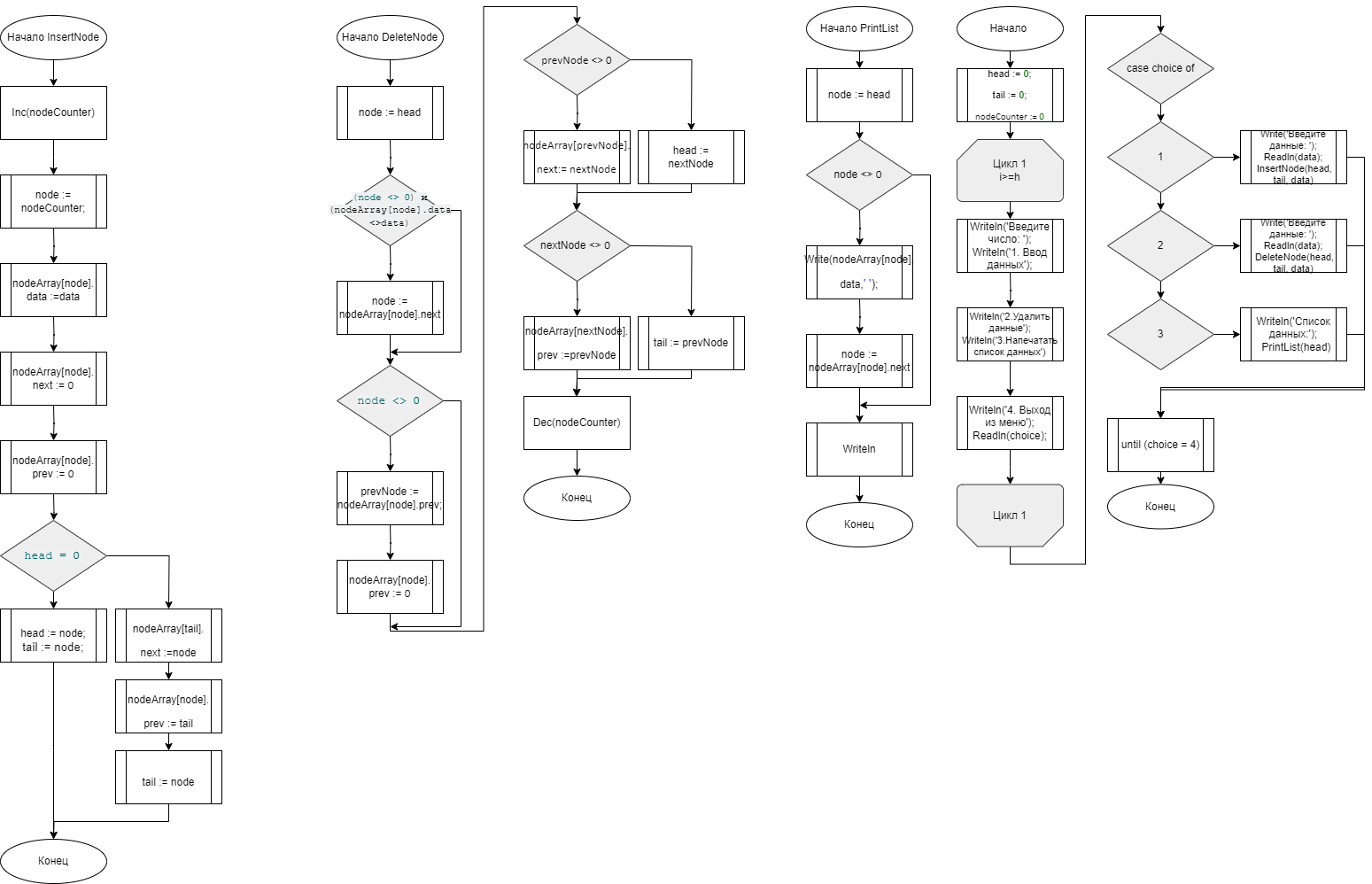
**end**;

**end**;

**until** (choice = 4);

**end**.

**Схема Алгоритма**



**Вывод**

В результате выполнения данного задания, я получил навыки работы с базовыми структурами данных, получил навыки организации case-меню.

Разработав программу для работы со структурой данных «Двусвязный список», которая позволяет добавлять данные, удалять и показывать. Это позволило мне попрактиковаться в работе со структурами данных изучить их основы. Одновременно я улучшил свои навыки программирования.