Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Вятский государственный университет»

Колледж ВятГУ

**ОТЧЕТ**

**ПО ДОМАШНЕЙ КОНТРОЛЬНОЙ РАБОТЕ №7**

**«РЕАЛИЗАЦИЯ ЭЛЕМЕНТАРНЫХ СТРУКТУР ДАННЫХ НА ОСНОВЕ СТАТИЧЕСКОЙ ПАМЯТИ»**

**ПО**

**МДК 05.02 РАЗРАБОТКА КОДА ИНФОРМАЦИОННЫХ СИСТЕМ**

Выполнил: студент учебной группы

ИСПк-205-52-00

Семанов Даниил Олегович

Преподаватель:

Сергеева Елизавета Григорьевна

Киров 2024

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

**Задание (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 сазе-меню. Предусмотреть наглядную визуализацию содержимого структуры.

**Описание алгоритма**

1. Создание процедуры AddNode(data: integer), которая принимает целочисленный параметр data.

2. Объявление указателя на структуру Node newNode.

3. Выделение памяти под новый узел с помощью функции New и присвоение указателю newNode адреса этой памяти.

4. Присвоение значению data поля data нового узла.

5. Присвоение значения nil поля prev нового узла.

6. Присвоение значения nil поля next нового узла.

7. Проверка наличия элементов в списке: если head = nil, то устанавливаем head и tail на новый узел.

8. Если список не пустой, то присваиваем полю next последнего узла tail значение указателя на новый узел.

Затем устанавливаем поле prev нового узла на указатель на последний узел tail, и обновляем указатель tail на новый узел.

9. Создание процедуры DisplayList для отображения элементов двусвязного списка.

10. Объявление указателя current на структуру Node и присвоение ему значения head.

11. Вывод сообщения о начале списка.

12. Цикличный проход по списку: пока current не равен nil, выводим значение поля data текущего узла и сдвигаем указатель current на следующий узел.

13. После завершения цикла выводим пустую строку для разделения вывода.

Начало программы.

14. Объявление переменных 'choice' и 'data' типа integer.

15. Начало процедуры 'Menu'.

16. Вывод на экран меню с вариантами действий: добавить узел, отобразить список, выйти.

17. Запрос выбора пользователя.

18. Считывание выбора пользователя в переменную 'choice'.

19. Использование конструкции case для выполнения действия в зависимости от выбора пользователя.

20. При выборе '1' пользователю предлагается ввести данные для нового узла.

21. Считывание данных в переменную 'data'.

22. Вызов процедуры AddNode для добавления узла с введенными данными.

23. При выборе '2' вызывается процедура DisplayList для отображения списка.

24. При выборе '3' выводится сообщение о завершении программы.

25. В случае некорректного выбора выводится сообщение с предложением выбрать еще раз.

26. Процедура 'Menu' повторяется до тех пор, пока выбор пользователя не равен '3'.

27. Конец процедуры 'Menu'.

28. Инициализация головы и хвоста списка как nil.

29. Вызов процедуры 'Menu'.

30. Конец программы.

**Схема алгоритма с комментариями**

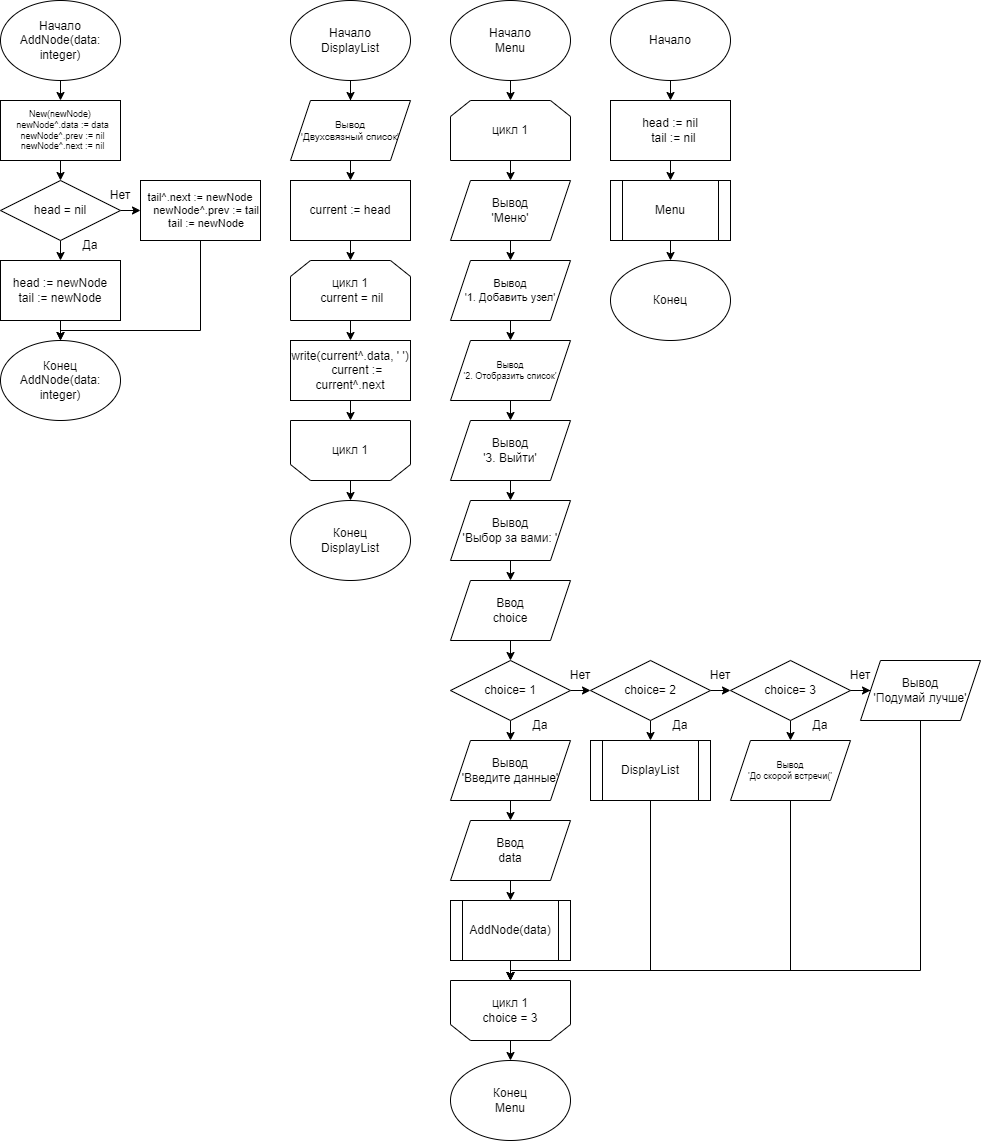
****

Рис.1 Схема алгоритма procedure AddNote.

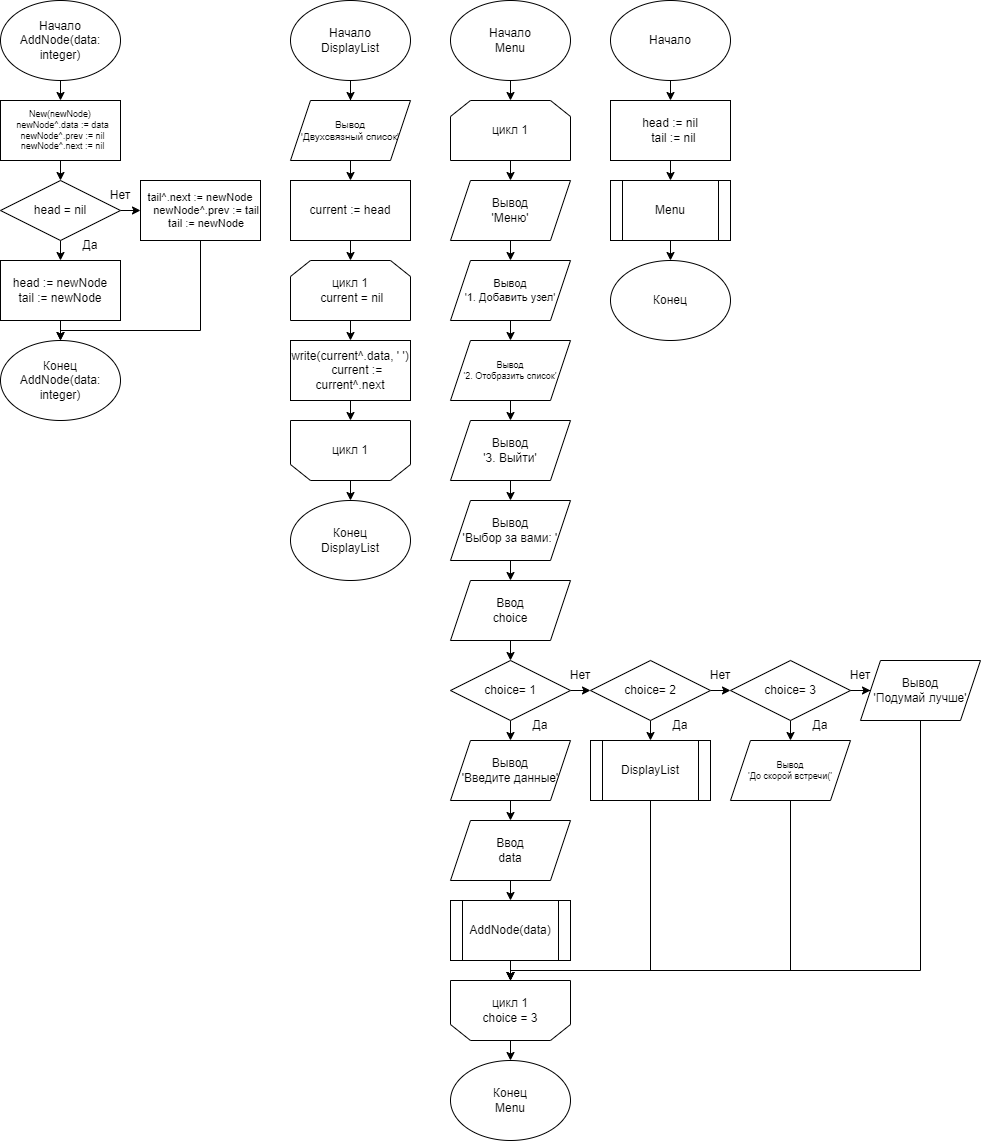
****

Рис.2 Схема алгоритма procedure DisplayList.



Рис.3 Схема алгоритма procedure Menu и самой программы.

**Код программы**

**program** dkr6;

**type**

Node = **record**

data: integer;

prev, next: ^Node;

**end**;

**var**

head, tail: ^Node;

**procedure** AddNode(data: integer);

**var**

newNode: ^Node;

**begin**

**New**(newNode);

newNode^.data := data;

newNode^.prev := nil;

newNode^.next := nil;

**if** head = nil **then**

**begin**

head := newNode;

tail := newNode;

**end**

**else**

**begin**

tail^.next := newNode;

newNode^.prev := tail;

tail := newNode;

**end**;

**end**;

**procedure** DisplayList;

**var**

current: ^Node;

**begin**

writeln('Двусвязный список:');

current := head;

**while** current <> nil **do**

**begin**

write(current^.data, ' ');

current := current^.next;

**end**;

writeln;

**end**;

**procedure** Menu;

**var**

choice, data: integer;

**begin**

**repeat**

writeln('Меню:');

writeln('1. Добавить узел');

writeln('2. Отобразить список');

writeln('3. Выйти');

write('Выбор за вами: ');

readln(choice);

**case** choice **of**

1: **begin**

write('Введите данные: ');

readln(data);

AddNode(data);

**end**;

2: DisplayList;

3: writeln('До скорой встречи(');

**else** writeln('Подумай лучше)');

**end**;

**until** choice = 3;

**end**;

**begin**

head := nil;

tail := nil;

Menu;

**end**.

**Результат выполнения программы**

1. Выбираем пункт 1 и вводим данные.

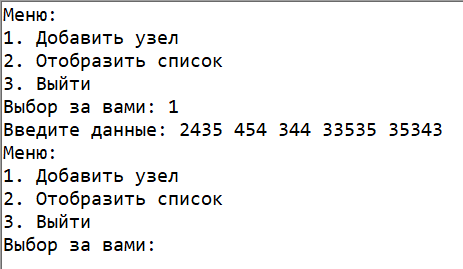


Рис.4 Результат выполнение программы при выборе 1 пункта.

1. Выбираем пункт 2.

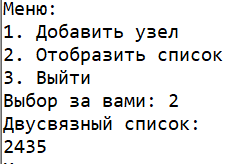


Рис.5 Результат выполнение программы при выборе 2 пункта.

1. Выбираем пункт 3.

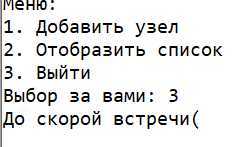


Рис.6 Результат выполнение программы при выборе 3 пункта.

1. Вводим любой другой пункт.

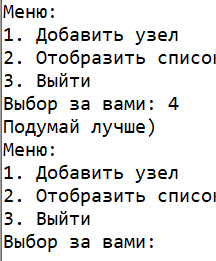


Рис.7 Результат выполнение программы.

Вывод: в результате изучения принципов работы с базовыми структурами данных и приобретения навыков организации базы данных-меню, студент приобретет необходимые навыки и знания для эффективного управления информацией, обработки данных и построения сложных запросов. Такие навыки могут быть полезными как при работе с базами данных в рамках профессиональной деятельности, так и при разработке программного обеспечения, ведении статистических исследований и анализе данных. В целом, изучение данных принципов позволит студенту более эффективно и качественно работать с информацией, улучшить свои навыки программирования и аналитические способности.

Эти навыки также могут быть полезны при создании веб-сайтов, приложений и других проектов, где требуется работа с данными. Понимание базовых структур данных позволит более эффективно реализовывать системы хранения, обработки и анализа информации, что повысит производительность и качество разрабатываемых продуктов. В общем, приобретение навыков работы с базовыми структурами данных и организации баз данных-меню имеет большое значение для развития профессиональных компетенций в области информационных технологий и программирования.