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 ДИСЦИПЛИНЕ «МДК 05.02 Разработка кода информационных систем»**

Выполнил: студент учебной группы

ИСПк-206-52-00

Табак Евгений Александрович

Преподаватель:

Сергеева Елизавета Григорьевна

Киров

2024

1. **Цель работы**

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

1. **Формулировка задания (с вариантом)**

Вариант: 8

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

**Описание алгоритма:**

Программа реализует циклический список через массив фиксированного размера, ограничивая количество элементов. В циклическом списке последний узел ссылается на первый, создавая замкнутую структуру.

Структура и Свойства:

TCircularList содержит массив Items элементов TItem (с данными Data и ссылкой Next на следующий элемент), индекс Head для начала списка и счетчик Size элементов.

Хранение и Связи:

Элементы связаны через индексы: Next каждого элемента указывает на следующий, формируя цикл от последнего к первому.

Основные Операции:

**Инициализация списка (InitializeList):** Устанавливает начальный размер списка в ноль и голову списка как отсутствующую. Это готовит структуру к добавлению элементов.

**Добавление элемента (AddItem):** Проверяет, есть ли место в списке. Если есть, то добавляет новый элемент в конец списка. Для первого элемента создает кольцо, указывая его Next на себя. Для последующих элементов устанавливает их Next на голову списка, поддерживая кольцевую структуру.

**Отображение списка (ShowList):** Проходит по элементам списка, начиная с головы и заканчивая, когда возвращается обратно к голове, демонстрируя кольцевую структуру. Печатает данные каждого элемента.

Кейс-Меню:

Предлагает пользователю три опции через простой текстовый интерфейс: добавить элемент, показать список, или выйти. Выбор реализован через бесконечный цикл и case, обеспечивая удобное управление.

**Схема алгоритма:**

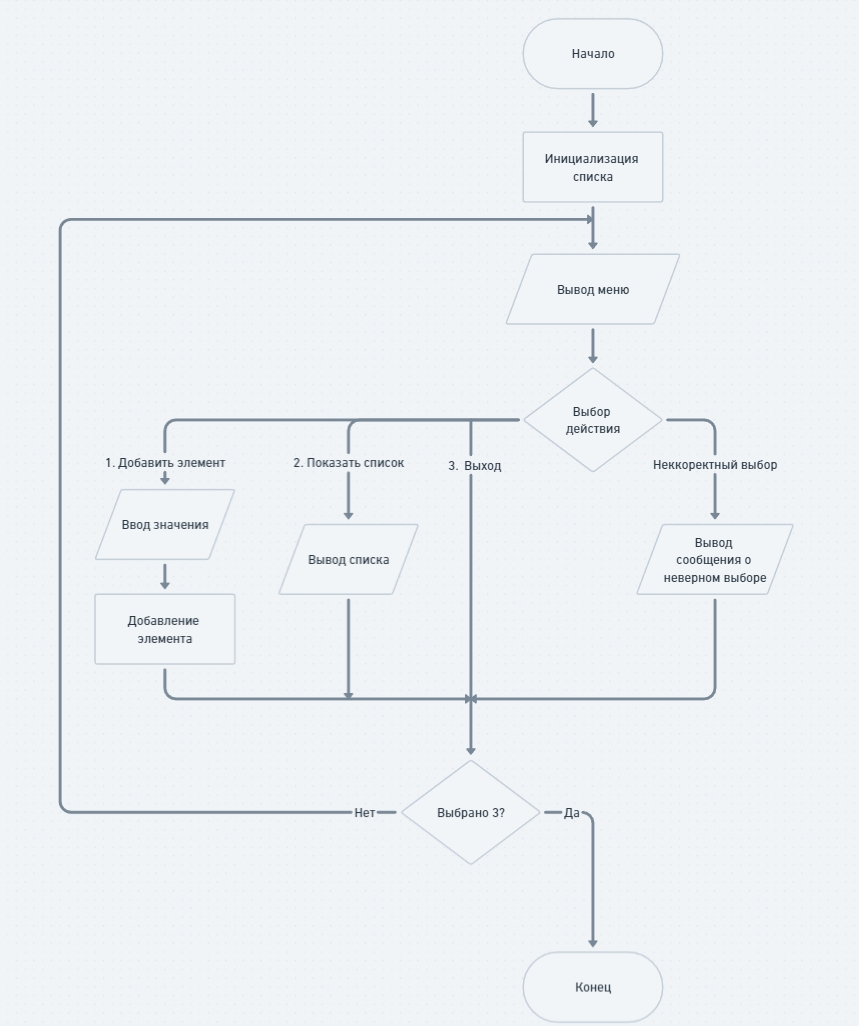


Рисунок 1 – Общая схема алгоритма программы.

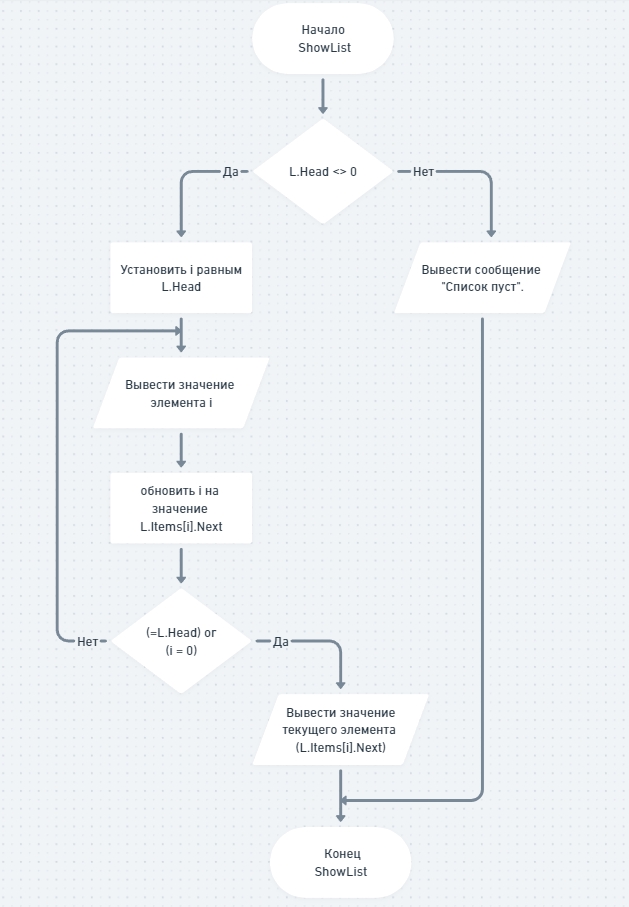
****

Рисунок 2 – Схема алгоритма процедуры

Рисунок 2 – Схема алгоритма процедуры ShowList.

**Изображение выглядит как текст, диаграмма, зарисовка, рисунок

Автоматически созданное описание**

Рисунок 3 – Схема алгоритма процедуры AddItem

**Код программы:**

**program** Dkr6;

**const**

MaxSize = 100;

**type**

TItem = **record**

Data: integer;

Next: integer;

**end**;

TCircularList = **record**

Items: **array**[1..MaxSize] **of** TItem;

Head: integer;

Size: integer;

**end**;

**var**

List: TCircularList;

Choice: integer;

**procedure** InitializeList(**var** L: TCircularList);

**begin**

L.Head := 0;

L.Size := 0;

**end**;

**procedure** AddItem(**var** L: TCircularList; Data: integer);

**var**

i, newIndex: integer;

**begin**

**if** L.Size < MaxSize **then**

**begin**

Inc(L.Size);

newIndex := L.Size;

L.Items[newIndex].Data := Data;

**if** L.Head = 0 **then**

**begin**

L.Head := newIndex;

L.Items[newIndex].Next := newIndex;

**end**

**else**

**begin**

i := L.Head;

**while** L.Items[i].Next <> L.Head **do**

i := L.Items[i].Next;

L.Items[i].Next := newIndex;

L.Items[newIndex].Next := L.Head;

**end**;

**end**

**else**

writeln('Список полон.');

**end**;

**procedure** ShowList(L: TCircularList);

**var**

i: integer;

**begin**

**if** L.Head <> 0 **then**

**begin**

i := L.Head;

**repeat**

write(L.Items[i].Data, ' ');

i := L.Items[i].Next;

**until** (i = L.Head) **or** (i = 0);

writeln;

**end**

**else**

writeln('Список пуст.');

**end**;

**begin**

InitializeList(List);

**repeat**

writeln('1. Добавить элемент');

writeln('2. Показать список');

writeln('3. Выход');

readln(Choice);

**case** Choice **of**

1: **begin**

writeln('Введите значение:');

readln(Choice);

AddItem(List, Choice);

**end**;

2: ShowList(List);

3: **break**;

**else**

writeln('Неверный выбор. Попробуйте снова.');

**end**;

**until** Choice = 3;

**end**.

**Результат выполнения программы:**

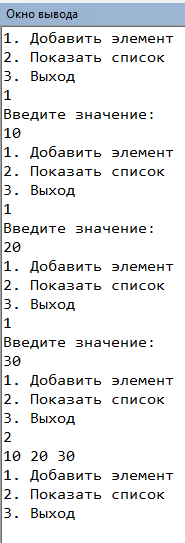
****

Рисунок 4 – Результат работы программы

**Вывод**

В ходе выполнения работы были изучены принципы работы с кольцевым односвязным списком и организация case-меню для управления структурой данных на основе статической памяти. Разработка программы позволила не только практически применить теоретические знания о структурах данных, но и улучшить навыки программирования и интерфейсной разработки. Результатом стало углубление понимания механизмов управления данными и взаимодействия с пользователем через наглядное case-меню.