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 «Основы алгоритмизации и программирования»**

Выполнил: студент учебной группы

ИСПк-205-52-00 Головёшкин Андрей Евгеньевич

Преподаватель:

Сергеева Елизавета Григорьевна

Киров

2024

**Вариант 2**

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

**Задание:**

1. Написать программу для работы со структурой данных "Кольцевой 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.

**Описание алгоритма:**

1. **Инициализация:**

* Программа запускается. Создаётся кольцевой список с фиксированным размером.
* Переменные инициализируются: список очищается, переменная option устанавливается в значение 1, флаг exitProgram – в false.

1. **Основной цикл работы:**

* Запускается цикл, который продолжается до тех пор, пока флаг exitProgram не станет true (то есть пока пользователь не нажмёт Esc).

1. **Отображение меню:**

* Вызывается процедура DisplayMenu, которая: Очищает экран и выводит меню с тремя пунктами: «Вставить элемент» «Удалить элемент» «Показать список» Подсвечивает (выделяет стрелочками) текущий выбранный пункт.
* Ожидает нажатия клавиши.
* Если нажата клавиша Esc, устанавливается флаг exitProgram и выход из процедуры.
* Если нажата стрелка вверх или вниз (обрабатываются через два вызова ReadKey), изменяется значение переменной option в пределах допустимого диапазона.
* Если нажата клавиша Enter, выбор подтверждается, и процедура завершается.

1. **Обработка выбранного пункта меню:**

* После завершения процедуры DisplayMenu в основной части программы производится проверка значения option:

Если option = 1 (Вставить элемент): Запрашивается у пользователя ввод числового значения. Вызывается процедура Insert для добавления элемента в кольцевой список.

Выводится сообщение об успешной вставке, затем ожидается нажатие Enter для продолжения. Если option = 2 (Удалить элемент): Вызывается процедура Delete для удаления элемента из кольцевого списка.

Выводится сообщение об успешном удалении, затем ожидается нажатие Enter для продолжения.

Если option = 3 (Показать список): Вызывается процедура Display, которая выводит текущее содержимое кольцевого списка.

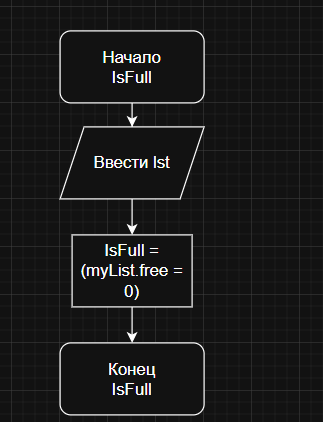
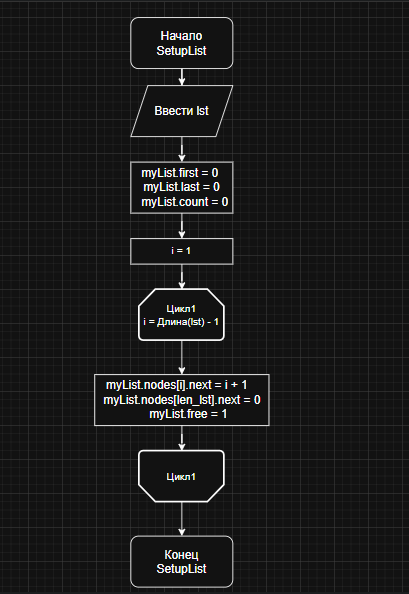
После отображения списка программа ожидает нажатие Enter для продолжения.

1. **Выход из программы:**

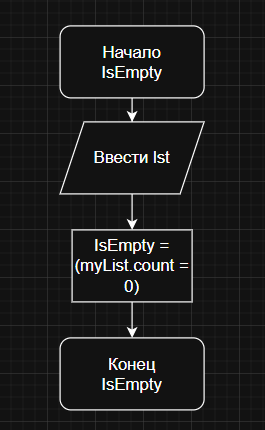
* Если в меню пользователь нажимает Esc, флаг exitProgram становится true, основной цикл завершается.
* Выводится сообщение «Выход...», и программа завершается.

**Схема алгоритма**

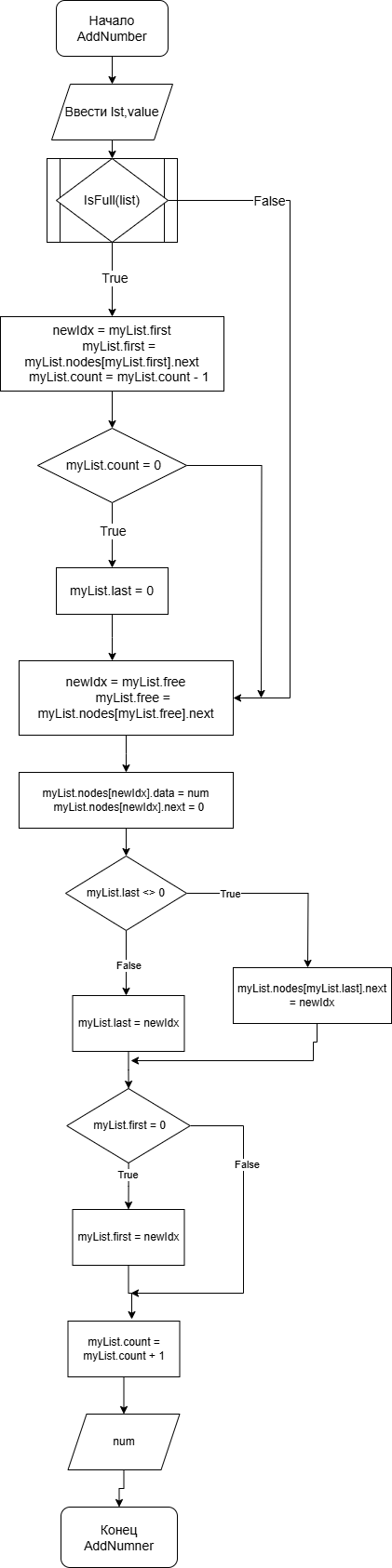
**Процедуры и функции:**

****

*Процедура 1 Функция 1*

**

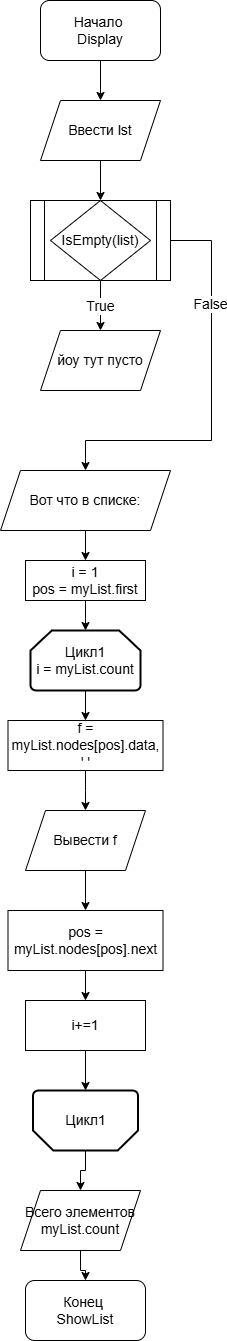
*Функция 2*

**

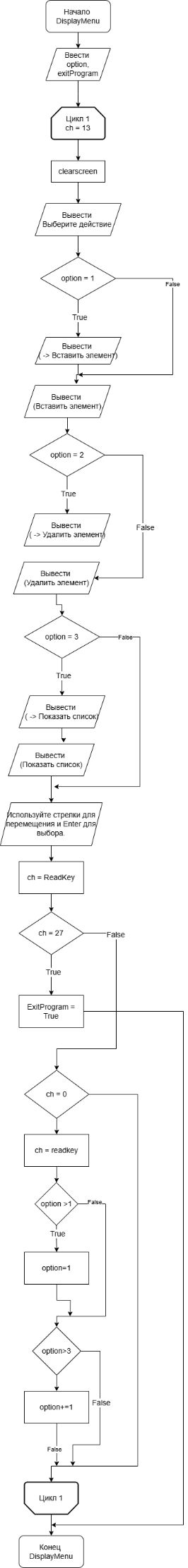
*Процедура 2*

**

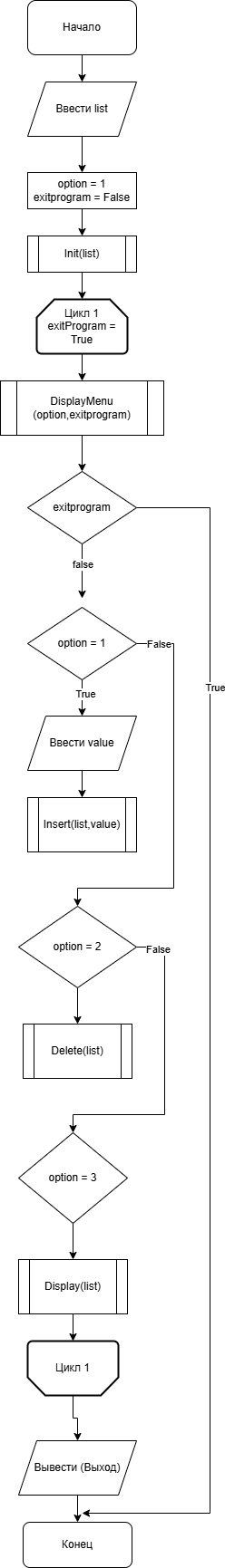
*Процедура 3*

**

*Процедура 5*

**

*Процедура 6*

**

*Программа 1*

**Код программы**

**program r3d1337;**

**uses crt;**

**const**

**len\_lst = 10;**

**type**

**Node = record**

**data: Integer;**

**next: Integer;**

**end;**

**RingList = record**

**nodes: array[1..len\_lst] of Node;**

**first, last, count: Integer;**

**free: Integer;**

**end;**

**procedure SetupList(var myList: RingList);**

**var**

**i: Integer;**

**begin**

**myList.first := 0;**

**myList.last := 0;**

**myList.count := 0;**

**for i := 1 to len\_lst - 1 do**

**myList.nodes[i].next := i + 1;**

**myList.nodes[len\_lst].next := 0;**

**myList.free := 1;**

**end;**

**function IsFull(myList: RingList): Boolean;**

**begin**

**IsFull := (myList.free = 0);**

**end;**

**function IsEmpty(myList: RingList): Boolean;**

**begin**

**IsEmpty := (myList.count = 0);**

**end;**

**procedure AddNumber(var myList: RingList; num: Integer);**

**var**

**newIdx: Integer;**

**begin**

**if IsFull(myList) then**

**begin**

**newIdx := myList.first;**

**myList.first := myList.nodes[myList.first].next;**

**myList.count := myList.count - 1;**

**if myList.count = 0 then**

**myList.last := 0;**

**end**

**else**

**begin**

**newIdx := myList.free;**

**myList.free := myList.nodes[myList.free].next;**

**end;**

**myList.nodes[newIdx].data := num;**

**myList.nodes[newIdx].next := 0;**

**if myList.last <> 0 then**

**myList.nodes[myList.last].next := newIdx;**

**myList.last := newIdx;**

**if myList.first = 0 then**

**myList.first := newIdx;**

**myList.count := myList.count + 1;**

**Writeln('Добавил число ', num);**

**end;**

**procedure RemoveNumber(var myList: RingList);**

**var**

**oldFirst: Integer;**

**begin**

**if IsEmpty(myList) then**

**begin**

**Writeln('а тут пусто,что удалять то');**

**Exit;**

**end;**

**oldFirst := myList.first;**

**myList.first := myList.nodes[myList.first].next;**

**myList.nodes[oldFirst].next := myList.free;**

**myList.free := oldFirst;**

**myList.count := myList.count - 1;**

**if myList.count = 0 then**

**myList.last := 0;**

**Writeln('удалил первый элемент');**

**end;**

**procedure ShowList(myList: RingList);**

**var**

**i, pos: Integer;**

**begin**

**if IsEmpty(myList) then**

**begin**

**Writeln('йоу тут пусто');**

**Exit;**

**end;**

**Write('Вот что в списке: ');**

**pos := myList.first;**

**for i := 1 to myList.count do**

**begin**

**Write(myList.nodes[pos].data, ' ');**

**pos := myList.nodes[pos].next;**

**end;**

**Writeln;**

**Writeln('Всего элементов: ', myList.count);**

**end;**

**procedure ShowMenu(var choice: Integer; var quit: Boolean);**

**var**

**key: Char;**

**begin**

**repeat**

**ClrScr;**

**Writeln('=== МОЙ КРУТОЙ КОЛЬЦЕВОЙ СПИСОК ===');**

**Writeln;**

**if choice = 1 then**

**Writeln('--> 1. Добавить число')**

**else**

**Writeln(' 1. Добавить число');**

**if choice = 2 then**

**Writeln('--> 2. Удалить число')**

**else**

**Writeln(' 2. Удалить число');**

**if choice = 3 then**

**Writeln('--> 3. Показать список')**

**else**

**Writeln(' 3. Показать список');**

**Writeln;**

**Writeln('ESC - Выход (но зачем?)');**

**key := ReadKey;**

**if key = #27 then**

**begin**

**quit := True;**

**Exit;**

**end;**

**if key = #0 then**

**begin**

**key := ReadKey;**

**case key of**

**#38: if choice > 1 then Dec(choice);**

**#40: if choice < 3 then Inc(choice);**

**end;**

**end;**

**until key = #13;**

**end;**

**var**

**myCoolList: RingList;**

**menuChoice, num: Integer;**

**exitProgram: Boolean;**

**begin**

**SetupList(myCoolList);**

**menuChoice := 1;**

**exitProgram := False;**

**repeat**

**ShowMenu(menuChoice, exitProgram);**

**if exitProgram then Break;**

**case menuChoice of**

**1: begin**

**Write('Введи число (только целое!): ');**

**Readln(num);**

**AddNumber(myCoolList, num);**

**Writeln('Нажми Enter чтобы продолжить...');**

**Readln;**

**end;**

**2: begin**

**RemoveNumber(myCoolList);**

**Writeln('Нажми Enter чтобы продолжить...');**

**Readln;**

**end;**

**3: begin**

**ShowList(myCoolList);**

**Writeln('Нажми Enter чтобы продолжить...');**

**Readln;**

**end;**

**end;**

**until exitProgram;**

**Writeln('На сегодня все');**

**end.**

**Вывод:**

Программа реализует взаимодействие с кольцевыми списками в Pascal через case-menu.