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»**

**«МДК 05. 02 Разработка кода информационных систем»**

Выполнил: студент учебной группы

ИСПк-206-52-00

Никулин Е.В.

Преподаватель:

Сергеева Е.Г.

Киров

2024

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

**Задание**:

1. Написать программу для работы со структурой данных “Кольцевой двусвязный список”.
2. Структура данных должна быть реализована на основе статистической памяти.
3. Работа со структурой должна осуществляться с помощью case-меню. Предусмотреть наглядную визуализацию содержимого структуры.

**Описание алгоритма**

**InitializeList:**

* Инициализируется двусвязный кольцевой список, заполняя массив list записями, каждая из которых содержит значение data, ссылку next на следующий элемент и ссылку prev на предыдущий.
* head и tail устанавливаются в 0, что означает, что список пуст.
* free устанавливается в 1, что означает, что первый элемент массива list доступен для использования.

**CreateNode:**

* Создает новый узел в списке, используя свободный элемент, указанный в free.
* Если свободных элементов нет (free = 0), то выводится сообщение об отсутствии памяти и программа завершается.
* Значение data нового узла устанавливается в переданное значение.
* next и prev указывают на себя, так как узел пока не вставлен в список.

**InsertFront:**

* Вставляет новый узел в начало списка.
* Если список пуст (head = 0), то новый узел становится head и tail.
* Если список не пуст, то новый узел становится head, а next старого head указывает на новый узел, а prev нового узла указывает на старый head.

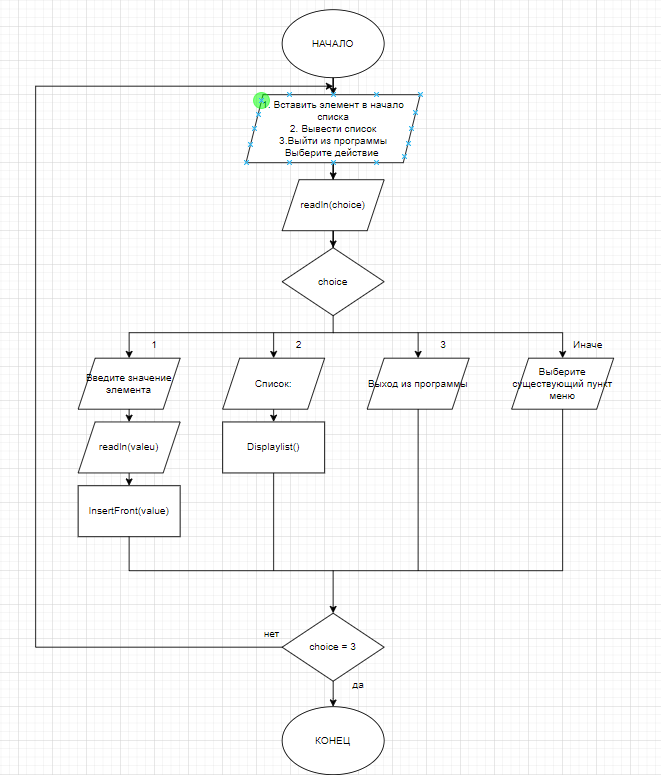
**DisplayList:**

* Выводит содержимое списка, начиная с head и перебирая узлы до тех пор, пока не будет достигнут head еще раз.
* Выводит значение data каждого узла, если оно не равно 0.

**Menu:**

* Отображает меню с вариантами действий: вставка элемента в начало списка, вывод списка и выход из программы.
* Получает выбор пользователя и выполняет соответствующее действие, пока пользователь не выберет вариант "3" для выхода из программы.

**Схема алгоритма**

****

**Код программы**

**program** Кольцевойдвусвязныйсписок;

**const**

MAX\_SIZE = 10;

**type**

Node = **record**

data: integer;

next, prev: integer;

**end**;

**var**

list: **array** [1..MAX\_SIZE] **of** Node;

head, tail, free: integer;

**procedure** InitializeList();

**var**

i: integer;

**begin**

**for** i := 1 **to** MAX\_SIZE **do**

**begin**

list[i].next := i + 1;

list[i].prev := i - 1;

**end**;

list[MAX\_SIZE].next := 1;

list[1].prev := MAX\_SIZE;

head := 0;

free := 1;

**end**;

**function** CreateNode(value: integer): integer;//добавить элемент в список

**var**

newNode: integer;

**begin**

**if** free = 0 **then**

**begin**

writeln('Не хватает памяти');

Halt;

**end**;

newNode := free;

free := list[free].next;

list[newNode].data := value;

CreateNode := newNode;

**end**;

**procedure** InsertFront(value: integer);

**var**

newNode: integer;

**begin**

newNode := CreateNode(value);

**if** head = 0 **then**

**begin**

head := newNode;

tail := newNode;

**end**

**else**

**begin**

list[newNode].next := head;

list[head].prev := newNode;

head := newNode;

**end**;

**end**;

**procedure** DisplayList(); //показать список

**var**

currentNode: integer;

**begin**

**if** head = 0 **then**

**begin**

writeln('Список пуст');

**Exit**;

**end**;

currentNode := head;

**repeat**

**if** (list[currentNode].data <> 0) **then**

writeln('Data: ', list[currentNode].data);

currentNode := list[currentNode].next;

**until** currentNode = head;

**end**;

**procedure** Menu(); //case menu

**var**

choice, value: integer;

**begin**

**repeat**

writeln('1. Вставить элемент в начало списка');

writeln('2. Вывести список');

writeln('3. Выйти из программы');

writeln('Выберите действие: ');

readln(choice);

**case** choice **of**

1: **begin**

writeln('Введите значение для нового элемента: ');

readln(value);

InsertFront(value);

**end**;

2: **begin**

writeln('Список:');

DisplayList();

**end**;

3: writeln('Выход из программы');

**else**

writeln('Неверный ввод. Пожалуйста, выберите существующий пункт меню.');

**end**;

**until** choice = 0;

**end**;

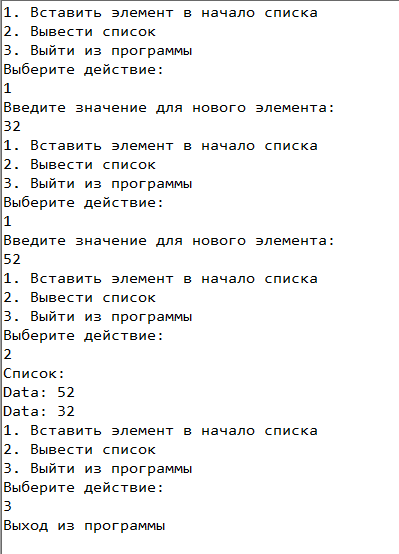
**begin**

InitializeList();

Menu();

**end**.

**Результаты выполнения программы**

****

**Ответы на вопросы**

.

.

.

.

.

.