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-204-52-00

Корзунин Евгений Сергеевич

Преподаватель:

Сергеева Елизавета Григорьевна

Киров

2023

В отчете должны отображаться:

* Цель работы
* Формулировка задания (с вариантом)
* Описание алгоритма
* Схема алгоритма с комментариями
* Код программы
* Результат выполнения программы
* Вывод

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

Вариант: 10.

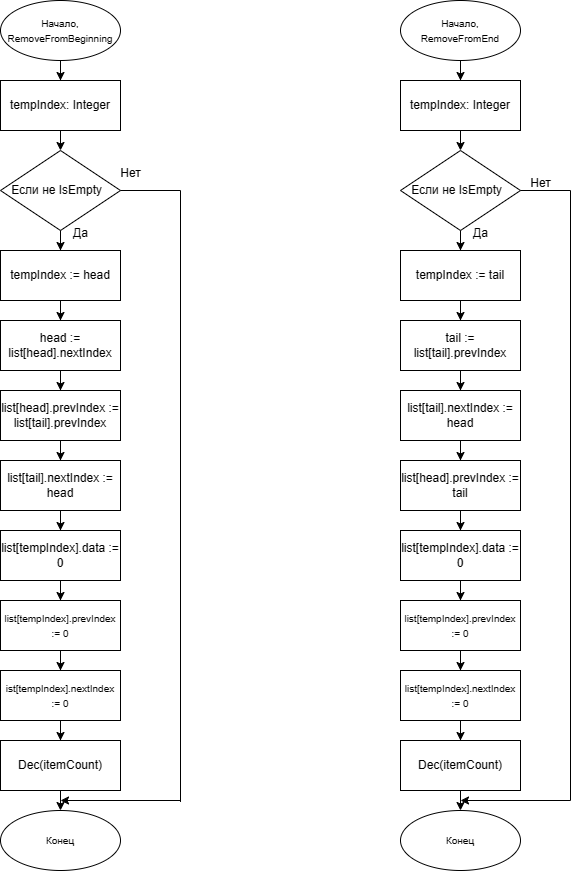
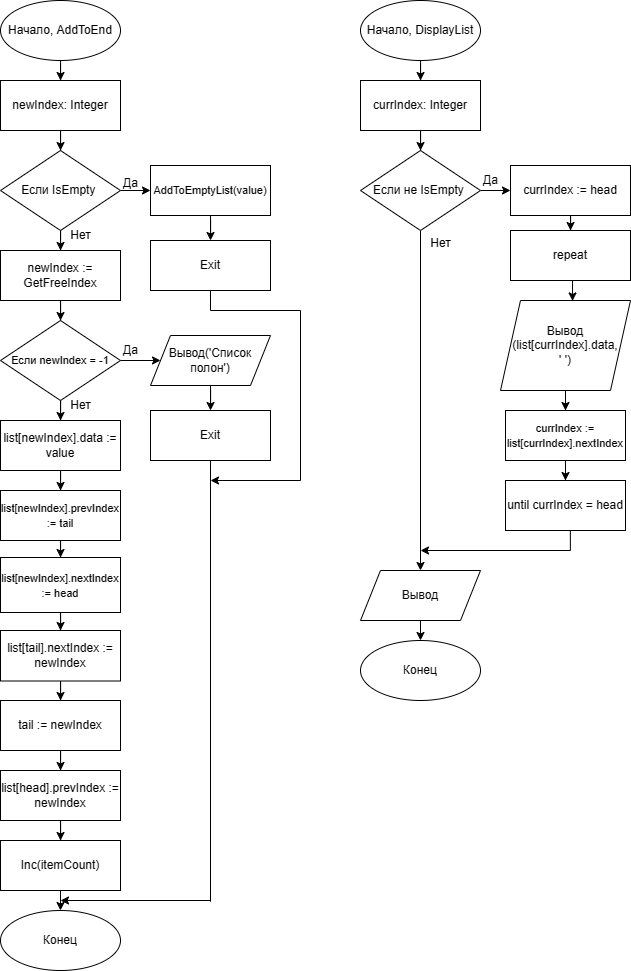
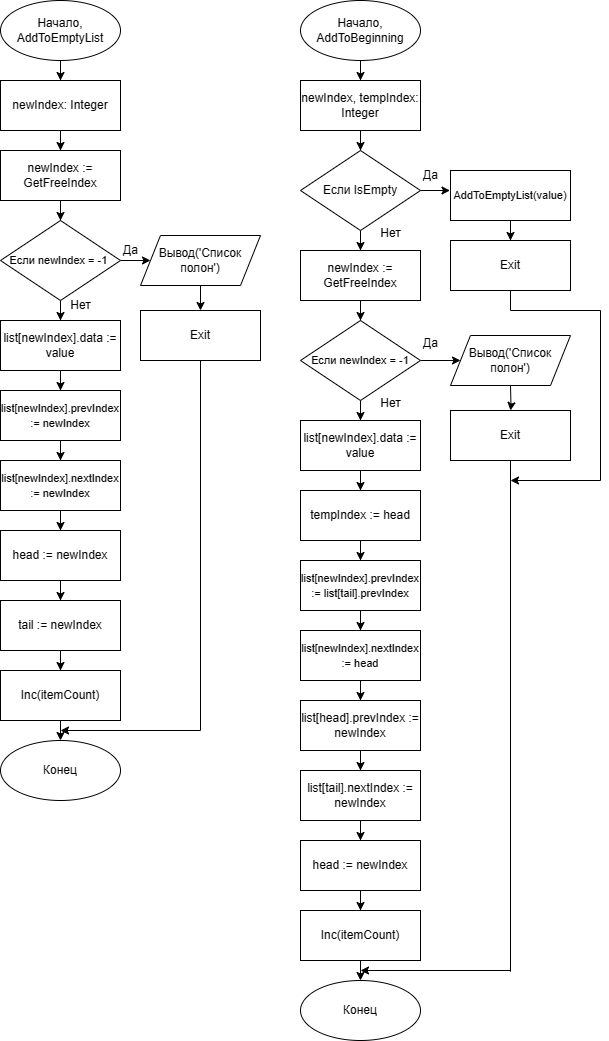
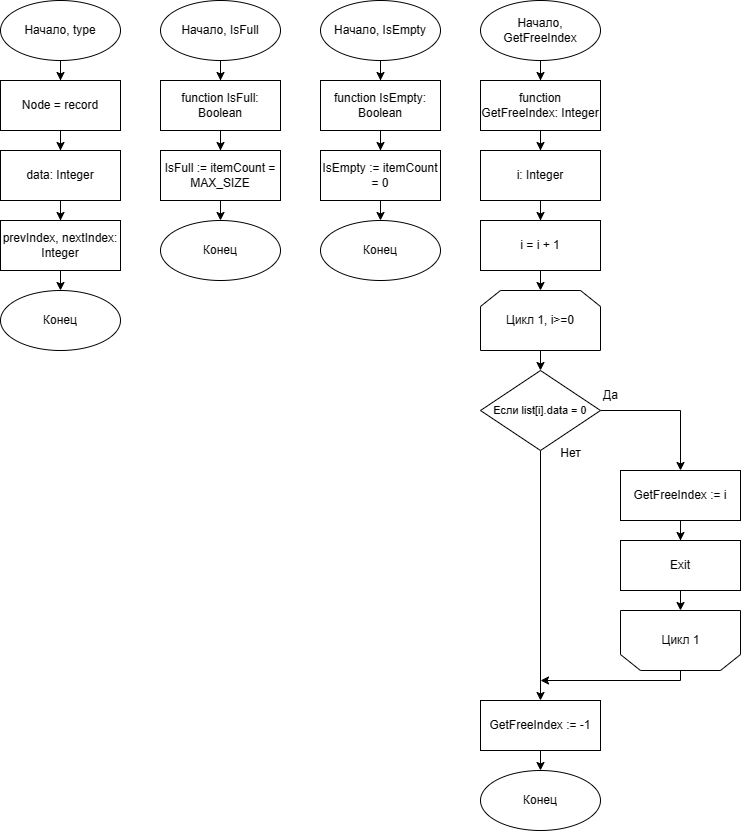
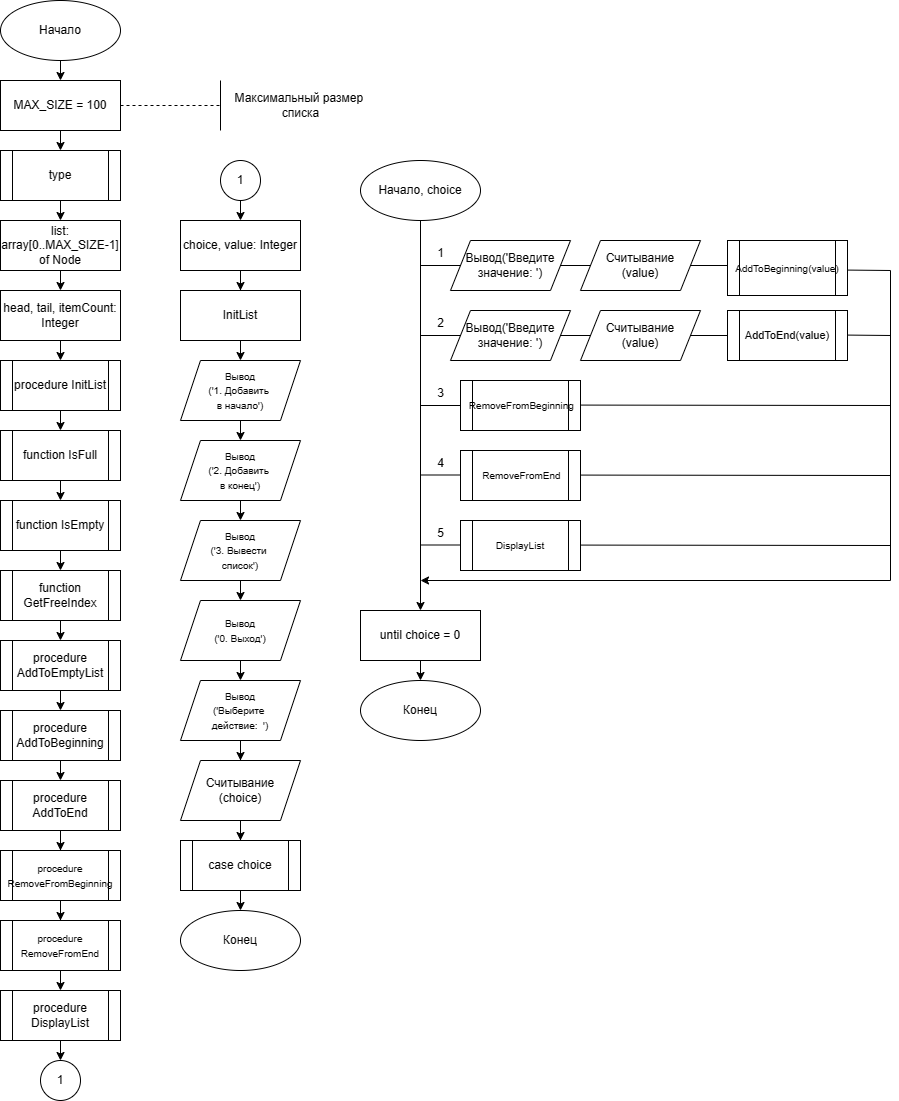
Задание:

1. Написать программу для работы со структурой данных «Кольцевой двусвязный 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:

Алгоритм состоит из 10 различных частей: функций, процедур и кейса.

Алгоритм:



Код программы:

**const**

MAX\_SIZE = 100; // Максимальный размер списка

**type**

Node = **record**

data: Integer;

prevIndex, nextIndex: Integer;

**end**;

**var**

list: **array**[0..MAX\_SIZE-1] **of** Node;

head, tail, itemCount: Integer;

**procedure** InitList;

**begin**

head := -1;

tail := -1;

itemCount := 0;

**end**;

**function** IsFull: Boolean;

**begin**

IsFull := itemCount = MAX\_SIZE;

**end**;

**function** IsEmpty: Boolean;

**begin**

IsEmpty := itemCount = 0;

**end**;

**function** GetFreeIndex: Integer;

**var**

i: Integer;

**begin**

**for** i := 0 **to** MAX\_SIZE - 1 **do**

**begin**

**if** list[i].data = 0 **then**

**begin**

GetFreeIndex := i;

**Exit**;

**end**;

**end**;

GetFreeIndex := -1; // Нет свободных индексов

**end**;

**procedure** AddToEmptyList(value: Integer);

**var**

newIndex: Integer;

**begin**

newIndex := GetFreeIndex;

**if** newIndex = -1 **then**

**begin**

WriteLn('Список полон');

**Exit**;

**end**;

list[newIndex].data := value;

list[newIndex].prevIndex := newIndex;

list[newIndex].nextIndex := newIndex;

head := newIndex;

tail := newIndex;

Inc(itemCount);

**end**;

**procedure** AddToBeginning(value: Integer);

**var**

newIndex, tempIndex: Integer;

**begin**

**if** IsEmpty **then**

**begin**

AddToEmptyList(value);

**Exit**;

**end**;

newIndex := GetFreeIndex;

**if** newIndex = -1 **then**

**begin**

WriteLn('Список полон');

**Exit**;

**end**;

list[newIndex].data := value;

tempIndex := head;

list[newIndex].prevIndex := list[tail].prevIndex;

list[newIndex].nextIndex := head;

list[head].prevIndex := newIndex;

list[tail].nextIndex := newIndex;

head := newIndex;

Inc(itemCount);

**end**;

**procedure** AddToEnd(value: Integer);

**var**

newIndex: Integer;

**begin**

**if** IsEmpty **then**

**begin**

AddToEmptyList(value);

**Exit**;

**end**;

newIndex := GetFreeIndex;

**if** newIndex = -1 **then**

**begin**

WriteLn('Список полон');

**Exit**;

**end**;

list[newIndex].data := value;

list[newIndex].prevIndex := tail;

list[newIndex].nextIndex := head;

list[tail].nextIndex := newIndex;

tail := newIndex;

list[head].prevIndex := newIndex;

Inc(itemCount);

**end**;

**procedure** RemoveFromBeginning;

**var**

tempIndex: Integer;

**begin**

**if not** IsEmpty **then**

**begin**

tempIndex := head;

head := list[head].nextIndex;

list[head].prevIndex := list[tail].prevIndex;

list[tail].nextIndex := head;

list[tempIndex].data := 0;

list[tempIndex].prevIndex := 0;

list[tempIndex].nextIndex := 0;

Dec(itemCount);

**end**;

**end**;

**procedure** RemoveFromEnd;

**var**

tempIndex: Integer;

**begin**

**if not** IsEmpty **then**

**begin**

tempIndex := tail;

tail := list[tail].prevIndex;

list[tail].nextIndex := head;

list[head].prevIndex := tail;

list[tempIndex].data := 0;

list[tempIndex].prevIndex := 0;

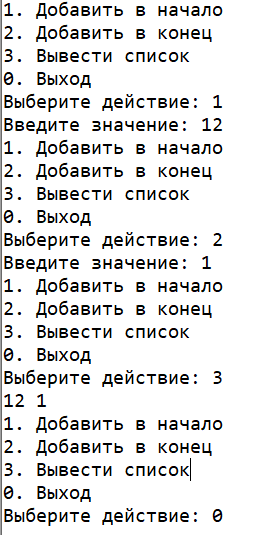
list[tempIndex].nextIndex := 0;

Dec(itemCount);

**end**;

**end**;

Результат выполнения программы:



Вывод.

Мы изучили принципы работы с базовыми структурами данных, получили навыки организации case-меню, построение функций, процедур. Научились создавать программы со структурой данных «Кольцевой двусвязный список» на основе статической памяти, а также предусмотрели управление программой при помощи case – меню.