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

Кошкин К.Е.

Преподаватель:

Сергеева Е.Г.

Киров

2024

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

**Вариант – 18.**

**Задание**:

1. Написать программу для работы со структурой данных «Дек».
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.**

Дек (deque, double-ended queue) — универсальная структура данных, которая представляет собой последовательность элементов с двумя концами. Добавление и удаление элементов может происходить как в начало, так и в конец структуры.

Особенности структуры дек:

1. Вставка элемента возможна как в начало, так и в конец.
2. Удаление элемента возможно как в начале, так и в конце.
3. Доступ к первому и последнему элементу производится за константное время O(1).
4. Доступ к элементам в середине дека осуществляется за линейное время O(n), так как элементы хранятся последовательно.

Дек представляет собой смесь стека и очереди. Структура дек может реализовываться различными способами, например, с использованием двух стеков или двусвязного списка.

**Схема алгоритма.**

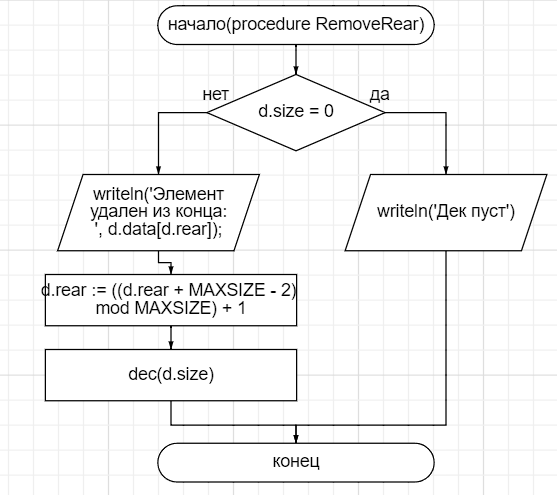
****

Рисунок 1. – procedure a

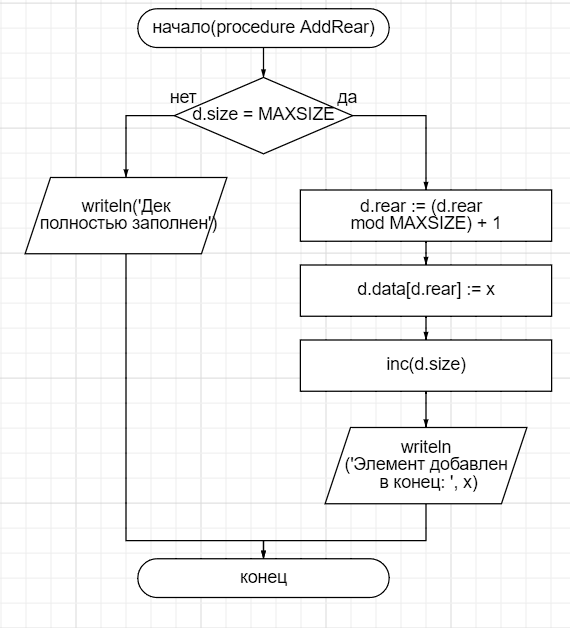
****

Рисунок 2. - procedure b

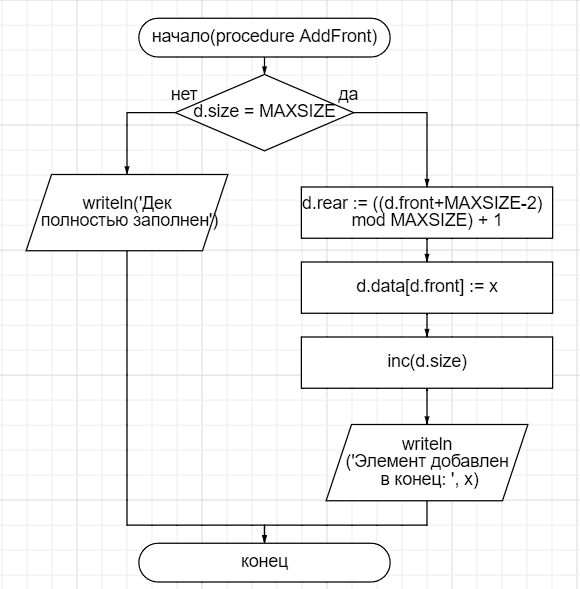
****

Рисунок 3. - procedure c

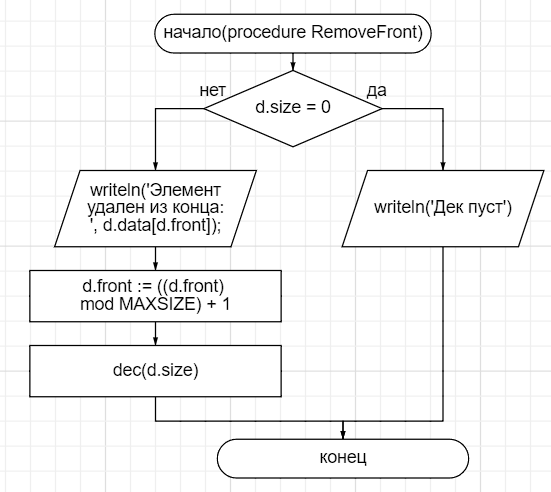
****

Рисунок 4. - procedure d

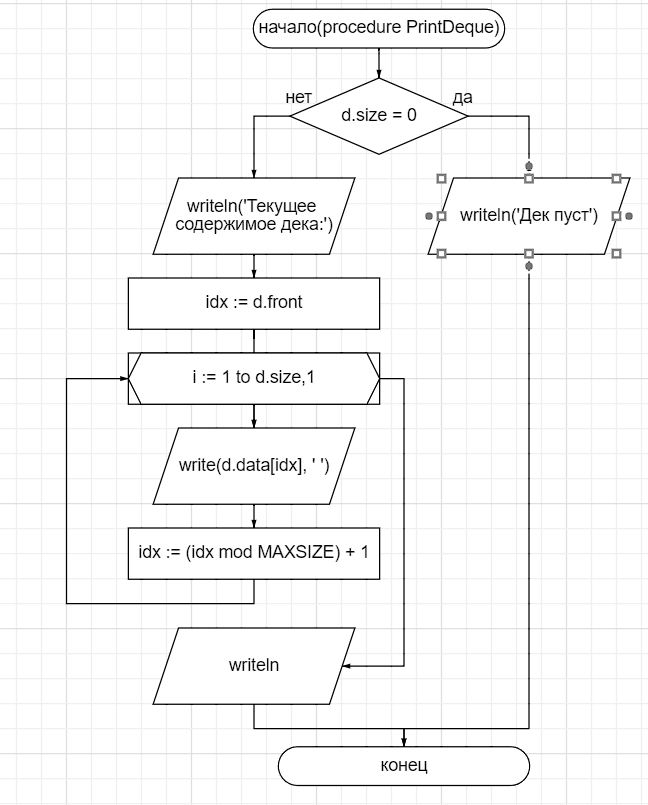
****

Рисунок 5. - procedure f

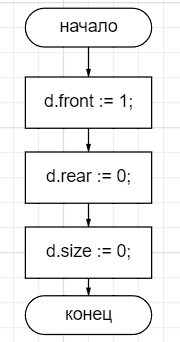
****

Рисунок 6. - Основная программа

**Код программы.**

**program** DequeExample;

**const**

MAXSIZE = 10; // Максимальный размер дека

**type**

Deque = **record**

data: **array**[1..MAXSIZE] **of** integer;

front: integer;

rear: integer;

size: integer;

**end**;

**var**

d: Deque;

**procedure** InitializeDeque(**var** d: Deque);

**begin**

d.front := 1;

d.rear := 0;

d.size := 0;

**end**;

**procedure** PrintDeque(**var** d: Deque);

**var**

i, idx: integer;

**begin**

**if** d.size = 0 **then**

writeln('Дек пуст')

**else**

**begin**

writeln('Текущее содержимое дека:');

idx := d.front;

**for** i := 1 **to** d.size **do**

**begin**

write(d.data[idx], ' ');

idx := (idx **mod** MAXSIZE) + 1;

**end**;

writeln;

**end**;

**end**;

**procedure** AddFront(**var** d: Deque; x: integer);

**begin**

**if** d.size = MAXSIZE **then**

writeln('Дек полностью заполнен')

**else**

**begin**

d.front := ((d.front + MAXSIZE - 2) **mod** MAXSIZE) + 1;

d.data[d.front] := x;

inc(d.size);

writeln('Элемент добавлен в начало: ', x);

**end**;

**end**;

**procedure** AddRear(**var** d: Deque; x: integer);

**begin**

**if** d.size = MAXSIZE **then**

writeln('Дек полностью заполнен')

**else**

**begin**

d.rear := (d.rear **mod** MAXSIZE) + 1;

d.data[d.rear] := x;

inc(d.size);

writeln('Элемент добавлен в конец: ', x);

**end**;

**end**;

**procedure** RemoveFront(**var** d: Deque);

**begin**

**if** d.size = 0 **then**

writeln('Дек пуст')

**else**

**begin**

writeln('Элемент удален из начала: ', d.data[d.front]);

d.front := (d.front **mod** MAXSIZE) + 1;

dec(d.size);

**end**;

**end**;

**procedure** RemoveRear(**var** d: Deque);

**begin**

**if** d.size = 0 **then**

writeln('Дек пуст')

**else**

**begin**

writeln('Элемент удален из конца: ', d.data[d.rear]);

d.rear := ((d.rear + MAXSIZE - 2) **mod** MAXSIZE) + 1;

dec(d.size);

**end**;

**end**;

**var**

choice, x: integer;

**begin**

InitializeDeque(d);

**repeat**

writeln('1. Добавить элемент в начало');

writeln('2. Добавить элемент в конец');

writeln('3. Удалить элемент из начала');

writeln('4. Удалить элемент из конца');

writeln('5. Показать дек');

writeln('0. Выход');

writeln('Выберите опцию:');

readln(choice);

**case** choice **of**

1: **begin**

writeln('Введите элемент:');

readln(x);

AddFront(d, x);

**end**;

2: **begin**

writeln('Введите элемент:');

readln(x);

AddRear(d, x);

**end**;

3: RemoveFront(d);

4: RemoveRear(d);

5: PrintDeque(d);

0: writeln('Выход из программы...');

**else**

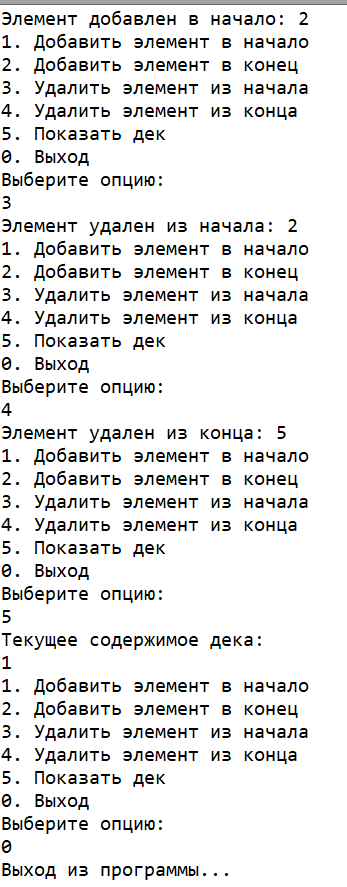
writeln('Неверный выбор, попробуйте снова.');

**end**;

**until** choice = 0;

**end**.

**Результат выполнения программы.**

****

**Вывод по работе.**

В ходе выполнения данной работы мной были закреплены знания об использовании case-меню, освежились в памяти массивы и работа с ними, а также были изучены принципы работы со структурой данных «Очередь» и изучено, что такое вообще «Дек». Данная тема будет полезна в дальнейшем для практики в программирования. Проблем во время выполнения работы никаких не возникло.