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 НА ОСНОВЕ СТАТИЧЕСКОЙ ПАМЯТИ»**

**ПО МДК 05.02**

**«РАЗРАБОТКА КОДА ИНФОРМАЦИОННЫХ СИСТЕМ»**

Выполнила: студент учебной группы

ИСПк-202-52-00

Савватеев Аркадий Сергеевич

Преподаватель:

Сергеева Елизавета Григорьевна

Киров

2025

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

**1.Формулировка задания 1. Вариант «76»**

**Постановка задачи**

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

**2.Описание алгоритма**

1. Создаем основную программу, в которой будет создаваться структура «очередь», а также все процедуры для добавления, удаления и просмотра содержимого.

2. создаем case-меню, в котором будет представлен выбор функций.

**3.Схема алгоритма**

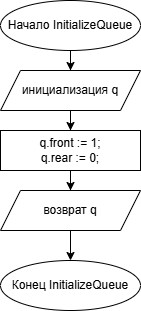


Рисунок 1.1 – схема алгоритма

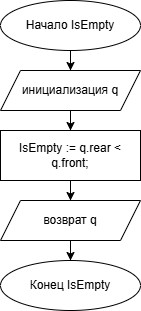


Рисунок 2 – схема алгоритма

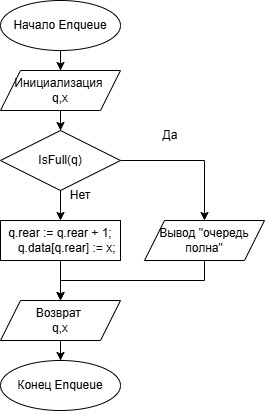


Рисунок 3 – схема алгоритма

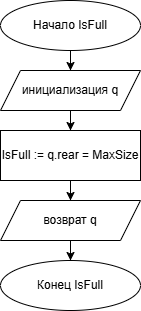


Рисунок 4 – схема алгоритма

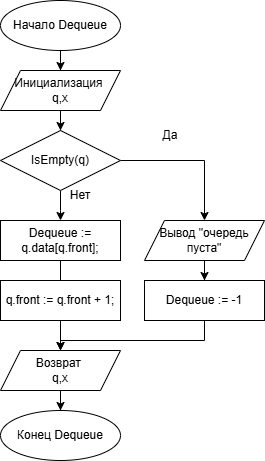


Рисунок 5 – схема алгоритма

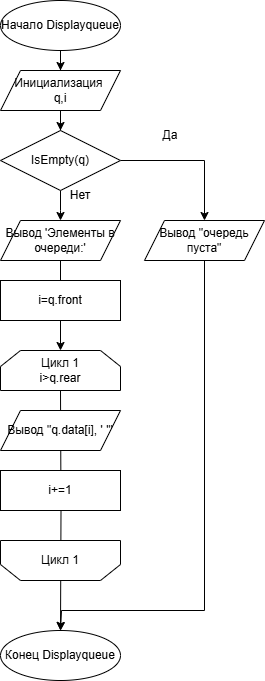


Рисунок 6 – схема алгоритма

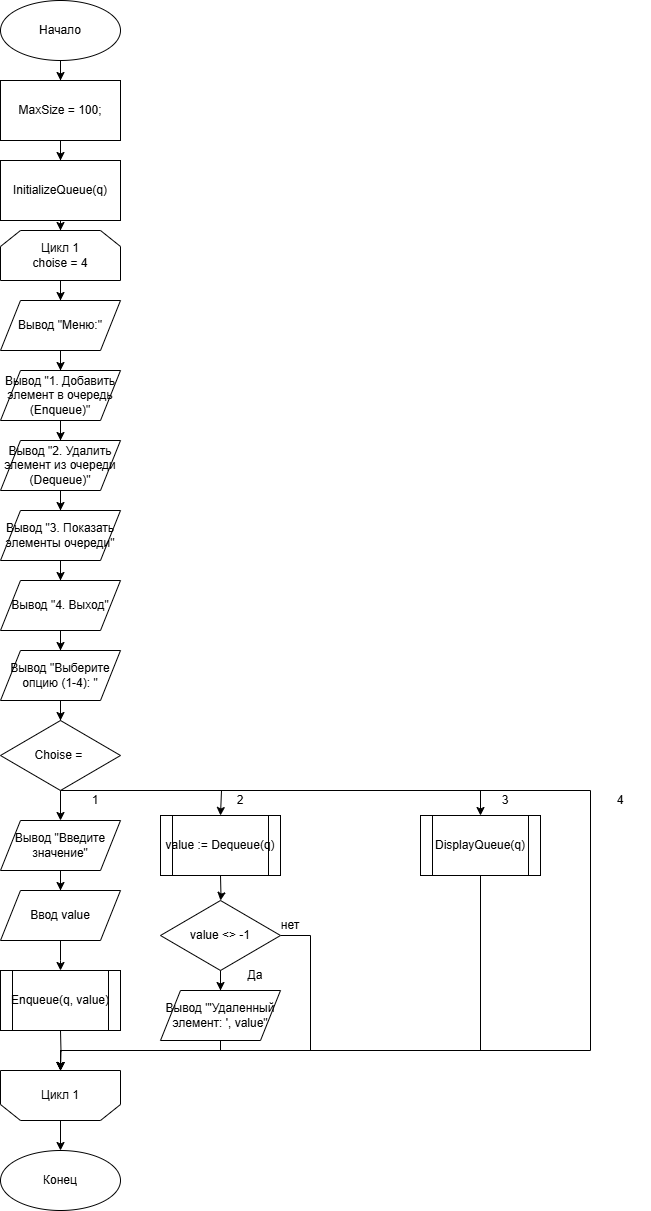


Рисунок 7 – схема алгоритма

**4.Код основной программы**

**program** QueueExample;

**const**

MaxSize = 100; // Максимальный размер очереди

**type**

Queue = **record**

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

front, rear: integer;

**end**;

**var**

q: Queue;

**procedure** InitializeQueue(**var** q: Queue);

**begin**

q.front := 1;

q.rear := 0;

**end**;

**function** IsEmpty(**var** q: Queue): boolean;

**begin**

IsEmpty := q.rear < q.front;

**end**;

**function** IsFull(**var** q: Queue): boolean;

**begin**

IsFull := q.rear = MaxSize;

**end**;

**procedure** Enqueue(**var** q: Queue; x: integer);

**begin**

**if** IsFull(q) **then**

writeln('Очередь полна, невозможно добавить элемент.')

**else**

**begin**

q.rear := q.rear + 1;

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

**end**;

**end**;

**function** Dequeue(**var** q: Queue): integer;

**begin**

**if** IsEmpty(q) **then**

**begin**

writeln('Очередь пуста, невозможно удалить элемент.');

Dequeue := -1;

**end**

**else**

**begin**

Dequeue := q.data[q.front];

q.front := q.front + 1;

**end**;

**end**;

**procedure** DisplayQueue(**var** q: Queue);

**var**

i: integer;

**begin**

**if** IsEmpty(q) **then**

writeln('Очередь пуста.')

**else**

**begin**

writeln('Элементы в очереди:');

**for** i := q.front **to** q.rear **do**

write(q.data[i], ' ');

writeln;

**end**;

**end**;

**var**

choice, value: integer;

**begin**

InitializeQueue(q);

**repeat**

writeln('Меню:');

writeln('1. Добавить элемент в очередь (Enqueue)');

writeln('2. Удалить элемент из очереди (Dequeue)');

writeln('3. Показать элементы очереди');

writeln('4. Выход');

writeln('Выберите опцию (1-4): ');

readln(choice);

**case** choice **of**

1: **begin**

writeln('Введите значение для добавления в очередь:');

readln(value);

Enqueue(q, value);

**end**;

2: **begin**

value := Dequeue(q);

**if** value <> -1 **then**

writeln('Удаленный элемент: ', value);

**end**;

3: DisplayQueue(q);

**end**;

**until** choice = 4;

**end**.

**5.Результат выполнения программы**

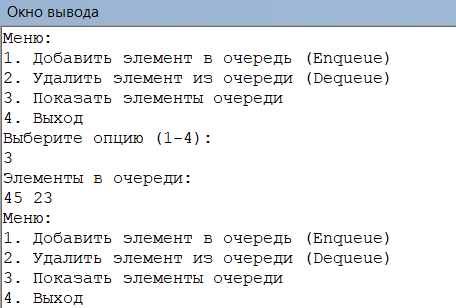
****

Рисунок 8 – пример выполнения программы

1. **Вывод**

В ходе работы мы освоили принципы работы со структурами данных; получили базовые умения взаимодействия с ними.

Для создания алгоритмических схем было изучено правильное построение письменного алгоритма для достижения верного решения задачи, а также и само построение схем в программе «draw.io».

В конечном итоге при помощи полученных знаний и исправленных ошибок у нас получилось добиться поставленной цели – решить задачу при помощи кода с условиями и циклами, а также узнать о языке программирования Pascal новые знания.