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 МДК 05.02 РАЗРАБОТКА КОДА ИНФОРМАЦИОННЫХ СИСТЕМ**

Выполнил: студент учебной группы

ИСПк-203-52-00

Цветкова Кристина Андреевна

Преподаватель:

Сергеева Елизавета Григорьевна

Киров

2024

1. **Цель домашней контрольной работы:**

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

**2. Формулировка задания**

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.

**3. Результат работы программы**

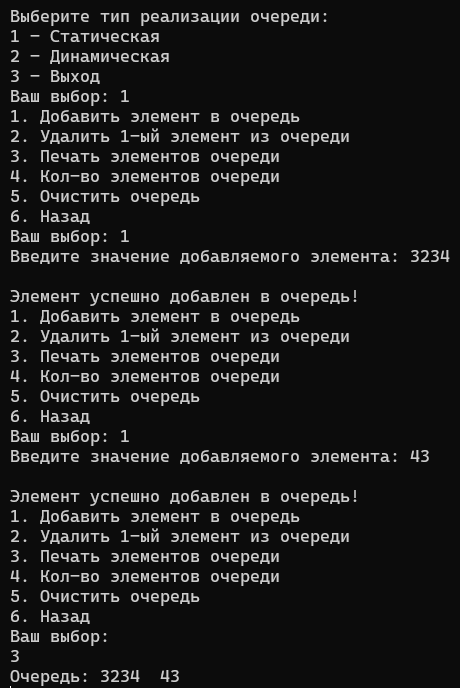
****

Рисунок 1 – Печать элементов очереди

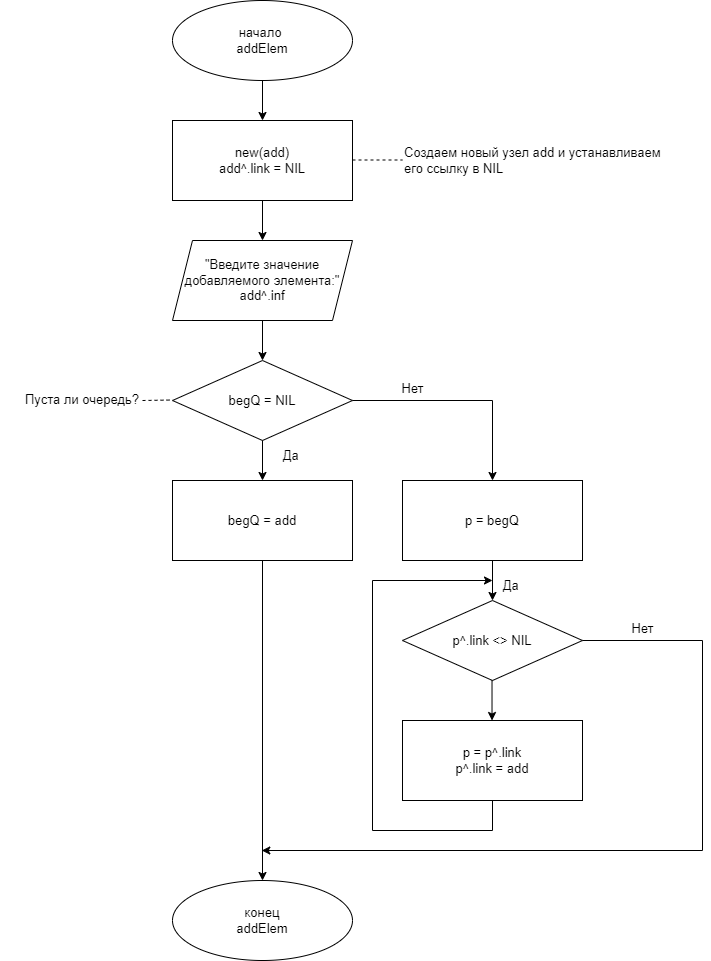


Рисунок 2 – Процедура добавления элемента

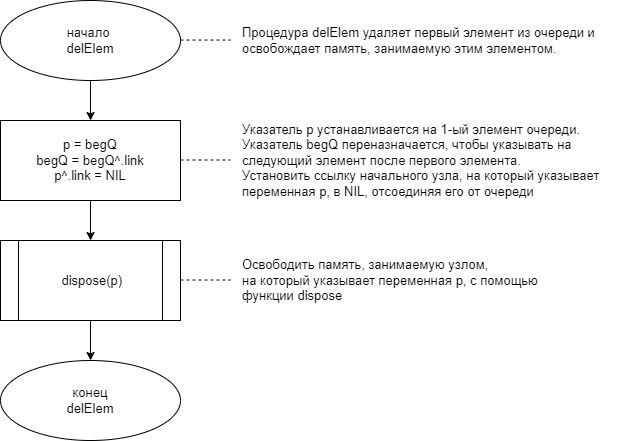
****

Рисунок 3 – Процедура удаления элемента

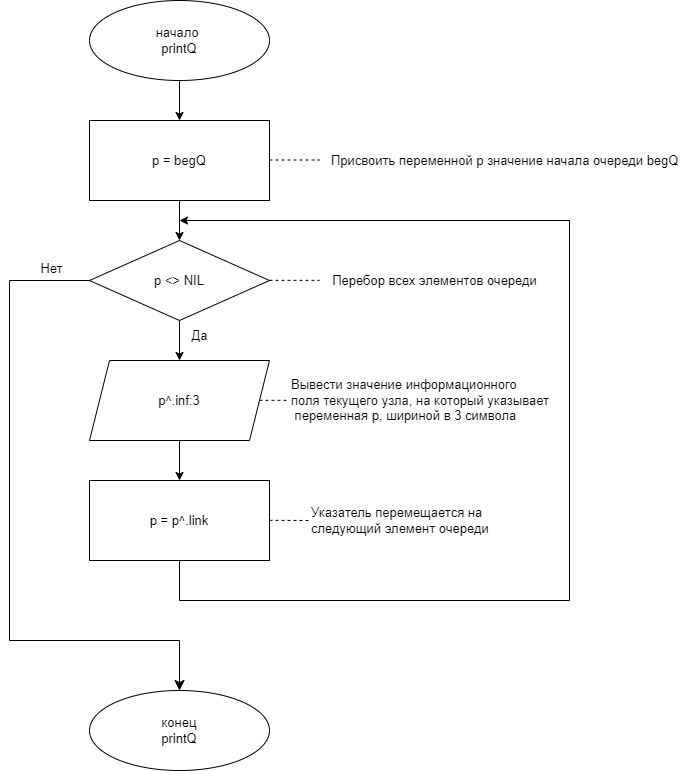


Рисунок 4 – Процедура вывода всех элементов очереди от начала к концу

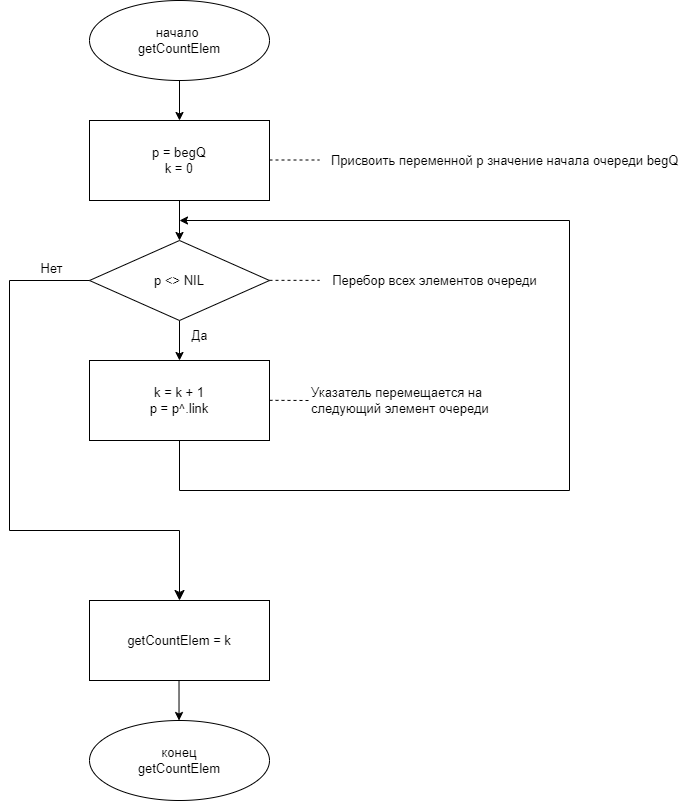


Рисунок 5 – Функция подсчета количества элементов в очереди

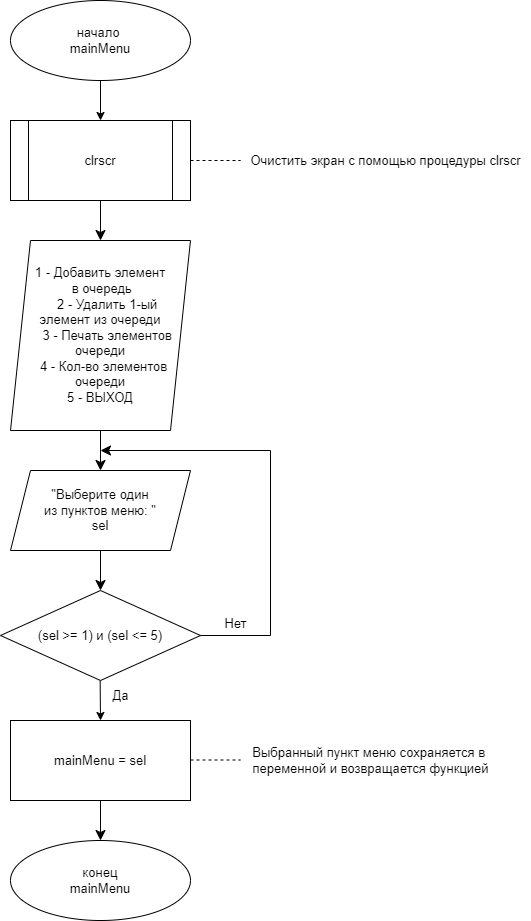


Рисунок 6 – Меню

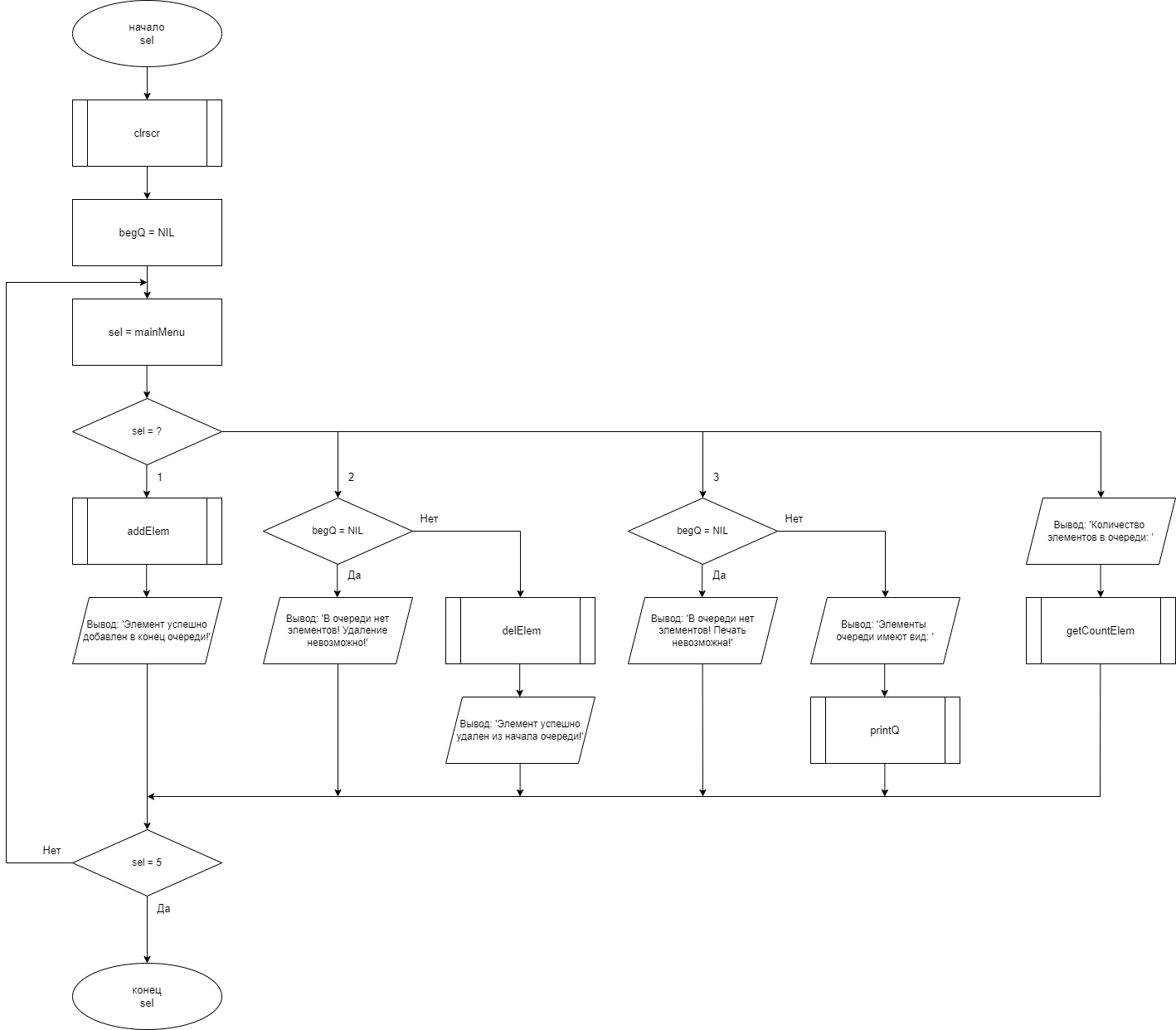


Рисунок 7 – Меню выбора

**4.Код программы**

**program** dkr\_6;

**uses** crt;

**type**

Tptr = ^Telem;

Telem = **record**

inf: integer;

link: Tptr;

**end**;

**var**

begQ: Tptr;

count: integer; // Количество элементов в очереди

sel, sel1: integer;

**procedure** Clear(**var** p: Tptr);

**var**

temp: Tptr;

**begin**

**while** p <> NIL **do**

**begin**

temp := p^.link;

dispose(p);

p := temp;

**end**;

begQ := NIL;

count := 0; // Обнуление счетчика элементов

**end**;

**function** getCountElem: Integer;

**begin**

getCountElem := count;

**end**;

**procedure** addElem;

**var**

add: Tptr;

p: Tptr;

value: string;

**begin**

**new**(add);

add^.link := NIL;

write('Введите значение добавляемого элемента: ');

readln(value);

add^.inf := StrToInt(value); // преобразуем строку в целое число

**if** (begQ = NIL) **then**

begQ := add

**else**

**begin**

p := begQ;

**while** (p^.link <> NIL) **do**

p := p^.link;

p^.link := add;

**end**;

inc(count);

**end**;

**procedure** delElem;

**var**

p: Tptr;

**begin**

**if** (begQ = NIL) **then**

writeln('В очереди нет элементов! Удаление невозможно!')

**else**

**begin**

p := begQ;

begQ := begQ^.link;

p^.link := NIL;

dispose(p);

dec(count);

**end**;

**end**;

**procedure** printQ;

**var**

p: Tptr;

**begin**

**if** (begQ = NIL) **then**

writeln('В очереди нет элементов! Печать невозможна!')

**else**

**begin**

p := begQ;

write('Очередь: ');

**while** (p <> NIL) **do**

**begin**

write(p^.inf:3, ' ');

p := p^.link;

**end**;

**end**;

**end**;

**function** mainMenu: integer;

**var**

choice: integer;

**begin**

**repeat**

clrscr;

writeln('Выберите тип реализации очереди:');

writeln('1 - Статическая');

writeln('2 - Динамическая');

writeln('3 - Выход');

write('Ваш выбор: ');

readln(sel1);

**case** sel1 **of**

1:

**repeat**

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

writeln('2. Удалить 1-ый элемент из очереди');

writeln('3. Печать элементов очереди');

writeln('4. Кол-во элементов очереди');

writeln('5. Очистить очередь');

writeln('6. Назад');

write('Ваш выбор: ');

readln(choice);

**case** choice **of**

1:

**begin**

addElem;

writeln;

writeln('Элемент успешно добавлен в очередь!');

readkey;

**end**;

2:

**begin**

delElem;

writeln;

writeln('Элемент успешно удален из начала очереди!');

readkey;

**end**;

3:

**begin**

printQ;

writeln;

readkey;

**end**;

4:

**begin**

writeln('Количество элементов в очереди: ', getCountElem);

readkey;

**end**;

5:

**begin**

Clear(begQ);

writeln;

writeln('Очередь очищена!');

readkey;

**end**;

**end**;

**until** choice = 6;

2:

**repeat**

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

writeln('2. Удалить 1-ый элемент из очереди');

writeln('3. Печать элементов очереди');

writeln('4. Кол-во элементов очереди');

writeln('5. Очистить очередь');

writeln('6. Назад');

write('Ваш выбор: ');

readln(choice);

**case** choice **of**

1:

**begin**

addElem;

writeln;

writeln('Элемент успешно добавлен в очередь!');

readkey;

**end**;

2:

**begin**

delElem;

writeln;

writeln('Элемент успешно удален из начала очереди!');

readkey;

**end**;

3:

**begin**

printQ;

writeln;

readkey;

**end**;

4:

**begin**

writeln('Количество элементов в очереди: ', getCountElem);

readkey;

**end**;

5:

**begin**

Clear(begQ);

writeln;

writeln('Очередь очищена!');

readkey;

**end**;

**end**;

**until** choice = 6;

**end**;

**until** sel1 = 3;

mainMenu := 0;

**end**;

**begin**

mainMenu;

**end**.

1. **Вывод**

В ходе выполнения домашней контрольной работы номер 6 были получены базовые навыки реализации консольных приложений на основе case-меню и работы с статическими структурами данных. Работа была разделена на несколько этапов: цель, формулировка задания, написание кода программы, результат выполнения программы, вывод. Код программы был написан на языке программирования Pascal. Делая вывод, стоит отметить, что в ходе работы все цели были достигнуты.

В процессе выполнения работы, для создания блок схемы мы познакомились с программой Draw io. Draw.io — инструмент для создания диаграмм, схем алгоритмов, интеллект-карт, бизнес-макетов, отношений сущностей, программных блоков и другого. Draw.io обладает богатым набором функций для визуализации большинства задач пользователя.