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**

Выполнила: студентка учебной группы ИСПк-203-52-00

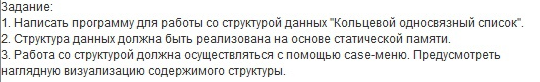
Гюльмамедова Лэйла Эльмир-кызы

Преподаватель:

Сергеева Елизавета Григорьевна

Киров

2023

1. Цель работы: изучить принципы работы с базовыми структурами данных, получение навыков организации case-меню.
2. 
3. Описание алгоритма

В программе есть 3 функции. Первая функция deleteNode служит для удаления узла в списке. Следящая функция addNode служит для добавления узла. AddNode работает по следующему алгоритму: если список переполнен то вызывается функция deleteNode, переходим на один элемент назад в списке и вызываем addNode. Иначе мы переходим к концу списка и добавляем то число, которое мы вели с клавиатуры. И последняя функция это printList1 она служит для вывода всего списка.

В самой же программе сделан case-меню.

1. Код программы

**uses** crt;

**const**

MAX\_SIZE = 6;

**type**

List = **array**[1..MAX\_SIZE] **of record**

next: integer;

data: integer;

**end**;

**var**

lit: List;

head, tail: integer;

**procedure** deleteNode;

**begin**

**if** head = tail **then**

writeln('Список пуст')

**else**

**begin**

lit[head].data := 0;

head := lit[head].next;

**end**;

**end**;

**procedure** addNode(a: integer);

**var**

i: integer;

**begin**

**if** (tail + 1) **mod** MAX\_SIZE = head **then begin**

deleteNode;

i := i - 1;

addNode(a);

**end**

**else**

**begin**

i := tail + 1;

**if** i = MAX\_SIZE **then**

**begin**

deleteNode;

i := 1;

**end**;

lit[i].data := a;

lit[tail].next := i;

tail := i;

**end**;

**end**;

**procedure** printList1;

**var**

i: integer;

**begin**

writeln('Кольцевой список:');

i := head;

**while** i <> tail **do**

**begin**

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

i := lit[i].next;

**end**;

write(lit[i].data);

writeln;

readln;

**end**;

**var**

choice: integer;

value: integer;

**begin**

head := 1;

tail := 1;

**repeat**

writeln('1. Добавка элемента');

writeln('2. Удаление элемента');

writeln('3. Показать список');

writeln('4. Выход');

write('Выберете вариант: ');

readln(choice);

**case** choice **of**

1:

**begin**

write('Ведите число: ');

readln(value);

addNode(value);

**end**;

2: deleteNode;

3: printList1;

4:;

**else** writeln('Не правильно выбран вариант');

**end**;

clrscr();

**until** choice = 4;

**end**.

1. Результаты выполнения программы

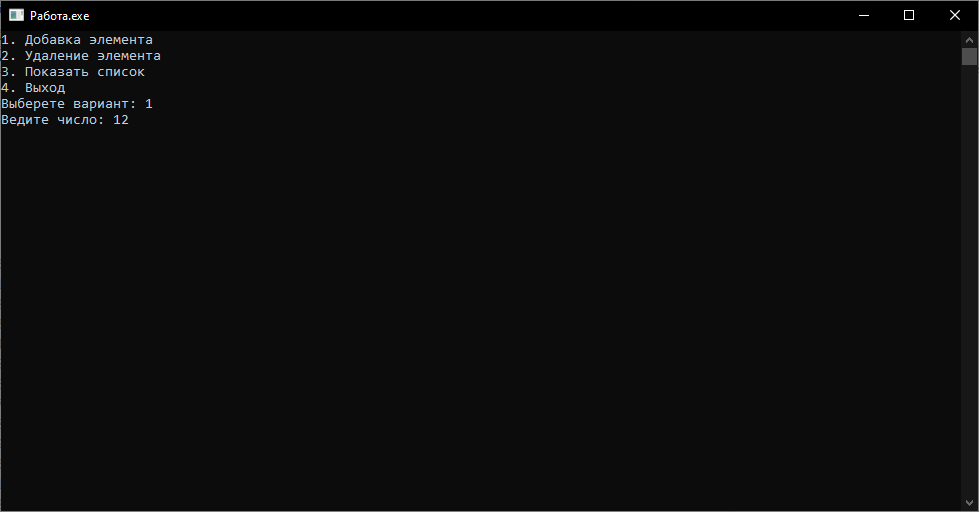


Рисунок 1 – Результат выполнения программы

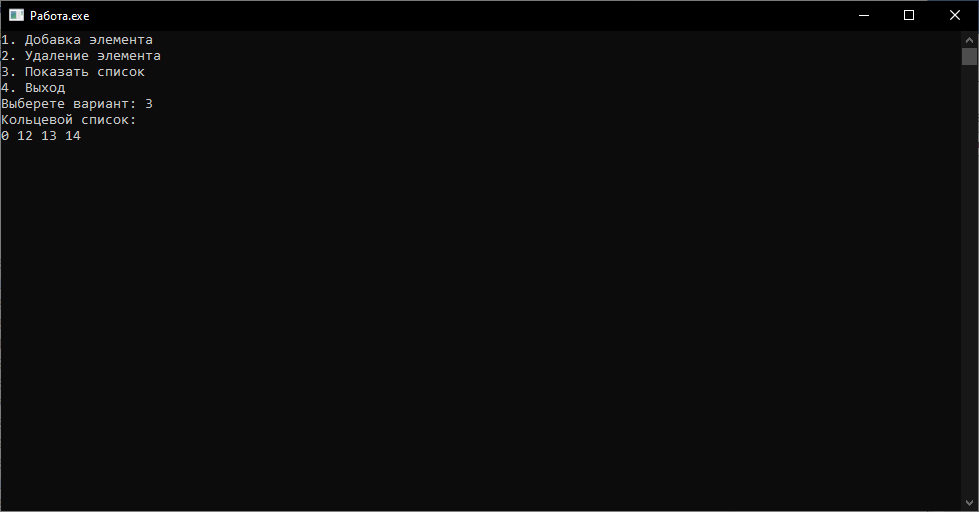


Рисунок 2 – Результат выполнения программы

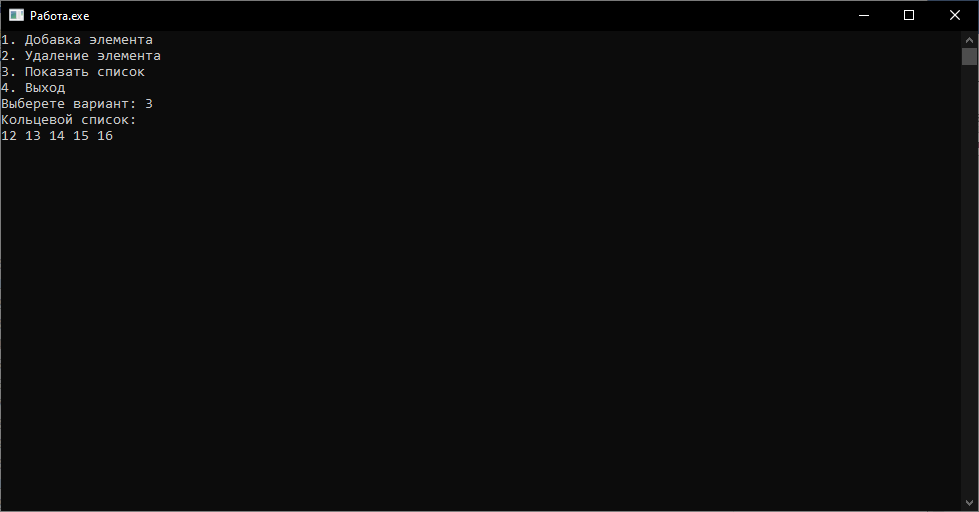


Рисунок 3 – Результат выполнения программы

1. Вывод

В этой домашней контрольной работе научилась работать с базовыми структурами данных и укрепила навыки организации case-меню.

Программа начинается с того что в списке обнуляется все для того чтоб не возникли ошибки в коде. После чего вызывается цикл repeat в котором реализован case-меню. В case-меню есть 4 варианта: 1 вариант для создания узла в списке, 2 вариант для удаления узла в списке, 3 вариант для просмотра всего списка, 4 вариант для выхода из цикла и в общем из программы. Если пользователь выберет неправильный вариант, то на экран выведется надпись: «Не правильно выбран вариант».