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: Евенок Маша

ИСПк-203-52-00

Колледж ВятГУ

Преподаватель:

Сергеева Елизавета Григорьевна

Киров

2023

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

2. Скриншот задания с вариантом:

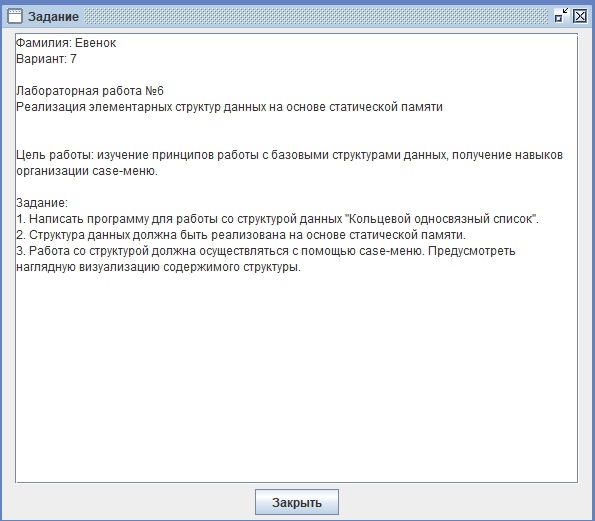


Рисунок 1. Скриншот задания

3. Описание алгоритма:

Может быть организован на основе как односвязного, так и двусвязного списков. При этом в односвязном списке указатель последнего элемента должен указывать на первый элемент; в двухсвязном списке в первом и последнем элементах соответствующие указатели переопределяются следующим образом:

4. Код программы:

**uses** crt;

**const**

MAX\_SIZE = 5;

**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('List is empty')

**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** printList;

**var**

i: integer;

**begin**

writeln('List:');

i := head;

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

**begin**

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

i := lit[i].next;

**end**;

write(lit[i].data);

writeln;

**end**;

**procedure** printList1;

**var**

i: integer;

**begin**

writeln('Список:');

i := head;

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

**begin**

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

i := lit[i].next;

**end**;

write(lit[i].data);

writeln;

**end**;

**var**

choice: integer;

a: integer;

**begin**

head := 1;

tail := 1;

**repeat**

writeln('1 - Создать элемент списка ');

writeln('2 - Удалить элемент из списка');

writeln('3- Просмотреть список');

writeln('4 - Выйти');

write('Выберте: ');

readln(choice);

**case** choice **of**

1: **begin**

write('Введите значение: ');

readln(a);

addNode(a);

{ClrScr;}

**end**;

2: **begin** deleteNode; {ClrScr;} **end**;

3: printList;

4: **exit**;

**else** writeln('Неправильно);

**end**;

**until** choice = 4;

**end**.

5. Результат выполнения работы

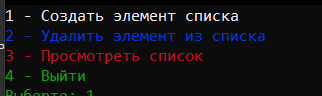


Рисунок 1. Case-меню

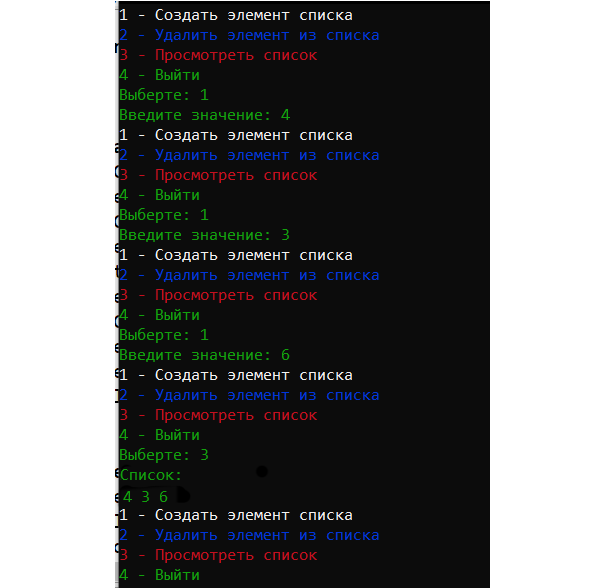


Рисунок 2. Список

6. Вывод:

В ходе выполнение данной работы мы поверили работу с case-меню, а также изучили принципы работы с базовыми структурами данных, и получили навыков организации case-меню. Были некоторые проблемы, но мы быстро находили решение, нужно было всего лишь получше разобраться и проблемы пропали. Я очень старалась ^-^