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 ДИСЦИПЛИНЕ «МДК 05.02 РАЗРАБОТКА КОДА ИНФОРМАЦИОННЫХ СИСТЕМ»**

Выполнил: студент учебной группы

ИСПк-205-52-00

Панов Антон Андреевич

Преподаватель:

Сергеева Елизавета Григорьевна

Киров,

2023

1. Цель домашней контрольной работы.

Изучение принципов работы с базовыми структурами данных, получение навыков организации case-меню.

1. Формулировка задания (с вариантом)

Вариант 4.

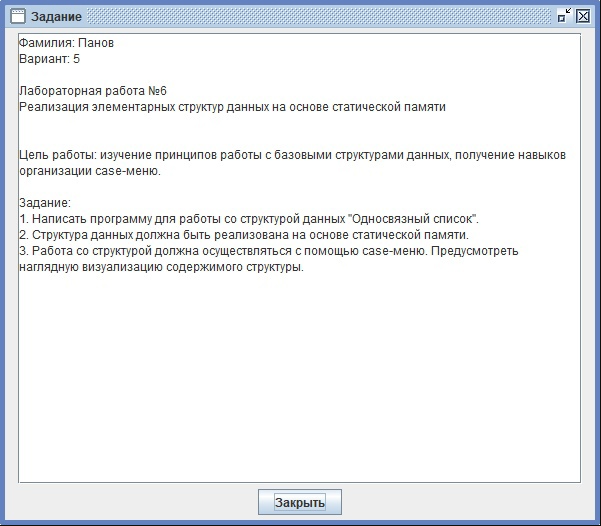


Рисунок 1 – Скриншот задания

1. Описание алгоритма

Массив структур Node, где каждый элемент содержит информацию о значении и индексе следующего элемента списка. Головной элемент списка хранится в переменной Head, а индекс первого свободного элемента списка хранится в переменной Free. InitializeList: процедура инициализирует список. Она заполняет массив List и переменные Head и Free значениями по умолчанию.

IsListEmpty: функция, которая возвращает true, если список пуст, и false, если список не пуст.

IsListFull: функция, которая возвращает true, если список переполнен, и false, если в списке есть свободное место.

GetFreeNode: функция, которая возвращает индекс свободного элемента списка.

ReleaseNode: процедура, которая освобождает элемент списка и делает его доступным для последующего использования.

AddToList: процедура, которая добавляет новый элемент со значением Data в начало списка.

RemoveFromList: процедура, которая удаляет первый элемент списка.

PrintList: процедура, которая выводит значения элементов списка на экран.

1. Код программы

**const**

MAX\_SIZE = 100;

**type**

Node = **record**

Data: Integer;

Next: Integer;

**end**;

**var**

List: **array**[1..MAX\_SIZE] **of** Node;

Head: Integer;

Free: Integer;

**procedure** InitializeList;

**var**

i: Integer;

**begin**

**for** i := 1 **to** MAX\_SIZE - 1 **do**

List[i].Next := i + 1;

List[MAX\_SIZE].Next := 0;

Head := 0;

Free := 1;

**end**;

**function** IsListEmpty: Boolean;

**begin**

Result := Head = 0;

**end**;

**function** IsListFull: Boolean;

**begin**

Result := Free = 0;

**end**;

**function** GetFreeNode: Integer;

**var**

NodeIndex: Integer;

**begin**

NodeIndex := Free;

Free := List[Free].Next;

List[NodeIndex].Next := 0;

Result := NodeIndex;

**end**;

**procedure** ReleaseNode(NodeIndex: Integer);

**begin**

List[NodeIndex].Next := Free;

Free := NodeIndex;

**end**;

**procedure** AddToList(Data: Integer);

**var**

NodeIndex: Integer;

**begin**

**if** IsListFull **then**

**begin**

writeln('Список переполнен');

**Exit**;

**end**;

NodeIndex := GetFreeNode;

List[NodeIndex].Data := Data;

**if** IsListEmpty **then**

Head := NodeIndex

**else**

List[NodeIndex].Next := Head;

Head := NodeIndex;

**end**;

**procedure** RemoveFromList;

**var**

NodeIndex: Integer;

**begin**

**if** IsListEmpty **then**

**begin**

writeln('Список пуст');

**Exit**;

**end**;

NodeIndex := Head;

Head := List[Head].Next;

ReleaseNode(NodeIndex);

**end**;

**procedure** PrintList;

**var**

NodeIndex: Integer;

**begin**

**if** IsListEmpty **then**

**begin**

writeln('Список пуст');

**Exit**;

**end**;

NodeIndex := Head;

**while** NodeIndex <> 0 **do**

**begin**

write(List[NodeIndex].Data, ' ');

NodeIndex := List[NodeIndex].Next;

**end**;

writeln;

**end**;

**var**

Choice, Value: Integer;

**begin**

InitializeList;

**repeat**

writeln('1. Добавить элемент');

writeln('2. Удалить элемент');

writeln('3. Вывести список');

writeln('4. Выход');

write('Выберите пункт: ');

readln(Choice);

**case** Choice **of**

1:

**begin**

write('Введите значение для добавления: ');

readln(Value);

AddToList(Value);

**end**;

2: RemoveFromList;

3: PrintList;

4: **break**;

**else**

writeln('Неккоректно');

**end**;

**until** False;

**end**.

1. Результат выполнения программы

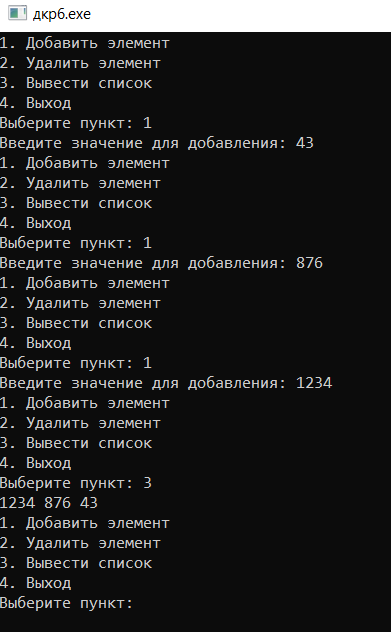


Рисунок 2 – Результат выполнения программы

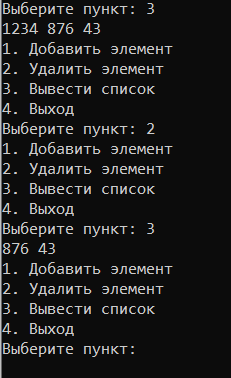


Рисунок 3 – Результат выполнения программы

1. Вывод

В ходе выполнения домашней контрольной работы №6 были изучены структуры данных. В первом этапе работы было выяснено понятие и виды структур данных. Во втором этапе изучены их реализация. Третий этап заключался в написании кода и его улучшении. Трудностей в работе не возникало.