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-202-52-00

Кротова Дарья Павловна

Преподаватель:

Сергеева Елизавета Григорьевна

Киров

2023

1. Цель контрольной работы

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

1. Формулировка задания (вариант 9)

Задание 1. Написать программу для работы со структурой данных «Односвязный список».

Задание 2. Структура данных должна быть реализована на основе статической памяти.

Задание 3. Работа со структурой должна осуществляться с помощью case-меню. Предусмотреть наглядную визуализацию содержимого структуры.

1. Описание алгоритма

Односвязный список – это динамическая структура данных, элементы которой содержат ссылку на следующий элемент. Последний элемент имеет в качестве ссылки NULL. Для доступа к списку используется указатель на первый элемент.

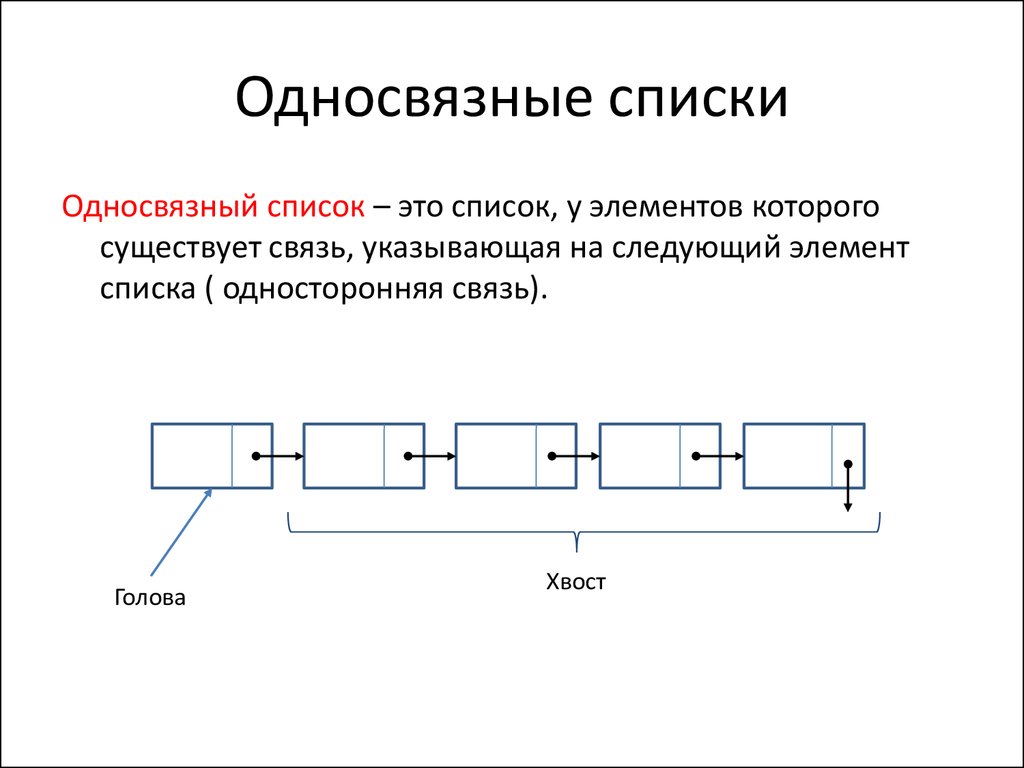


Рисунок 1 – Односвязный список. Логическая структура.

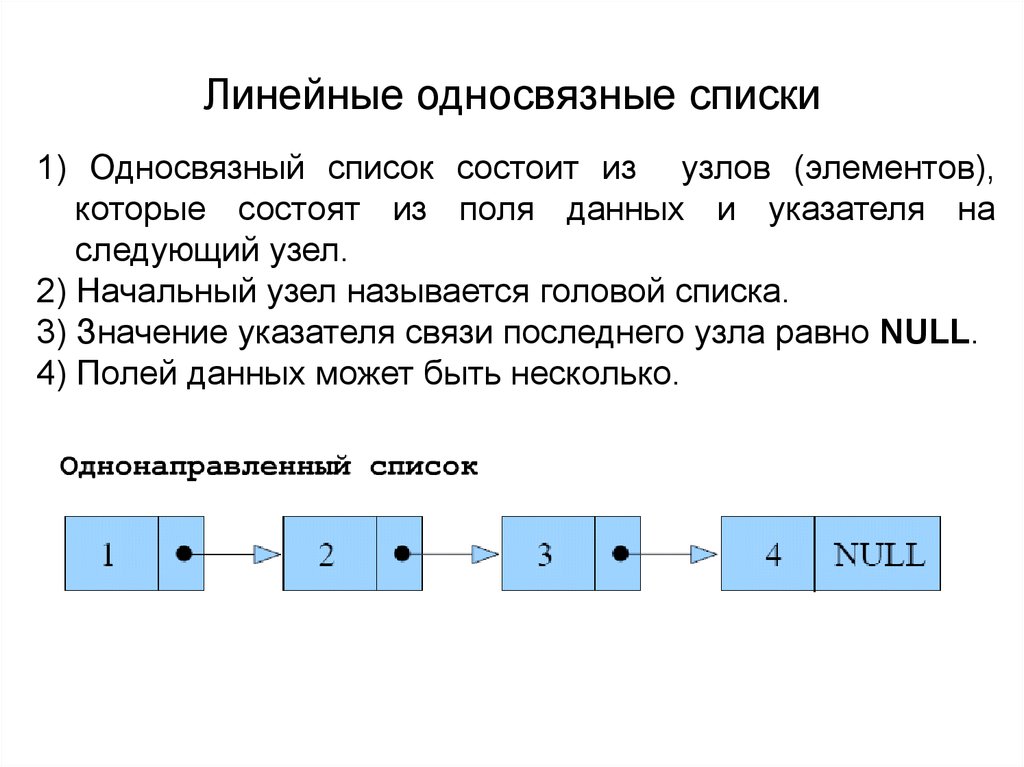


Рисунок 2 – Линейность односвязного списка.

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. Результаты выполнения программы:

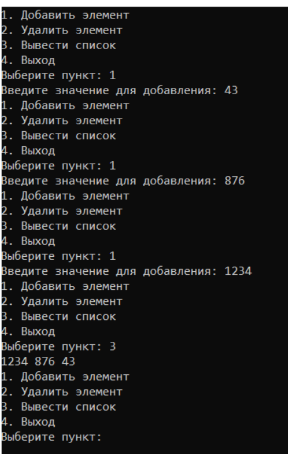


Рисунок 3 – Результат выполнения программы

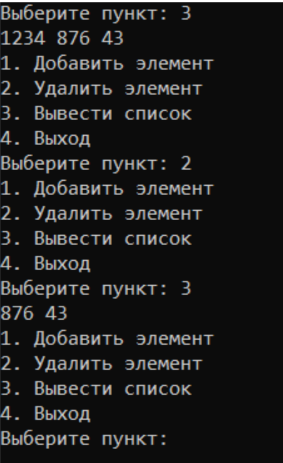


Рисунок 4 – Результат выполнения программы

1. Вывод

В ходе выполнения домашней контрольной работы №6 структура данных «Односвязного списка» реализована на основе статической памяти и работы со структурой осуществлена с помощью case-меню и предусмотрена наглядная визуализации содержимого структуры.

Данная работа выполнена в полном объеме, цель домашней контрольной работы №6 была достигнута, так же были получены новые знания и умения.