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 «ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ»**

Выполнил: студент учебной группы

ИСПк-205-52-00

Останин Кирилл Игоревич

Преподаватель:

Сергеева Елизавета Григорьевна

Киров

2024

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

**Задание**: **Вариант 11**

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

**Описание алгоритма**

Шаг 1: Определение структуры данных

- Создайте структуру данных для одного элемента списка, включающую поле для данных и указатель на следующий элемент.

- Определите массив элементов, который будет представлять собой статический односвязный список.

- Создайте указатель на начало списка.

Шаг 2: Инициализация списка

- Напишите процедуру инициализации списка, в которой заполните каждый элемент списка значениями по умолчанию и правильно свяжите их указателями.

- Установите указатель на начало списка на первый элемент.

Шаг 3: Реализация case-меню

- Создайте процедуру для отображения содержимого списка, которая будет последовательно проходить по элементам списка и выводить их данные.

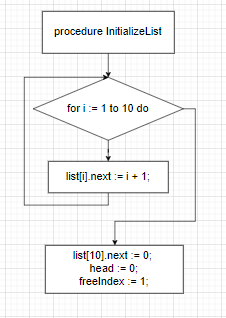
- Создайте case-меню, в котором предусмотрите выбор различных действий с односвязным списком, таких как вывод списка, добавление элемента, удаление элемента и т.д.

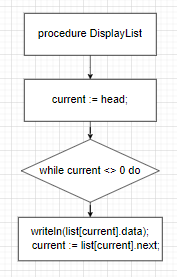
Шаг 4: Главная программа

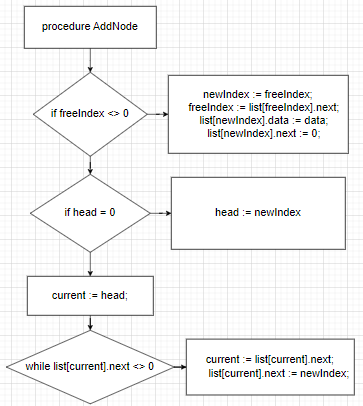
- В главной программе инициализируйте список с помощью созданной процедуры инициализации.

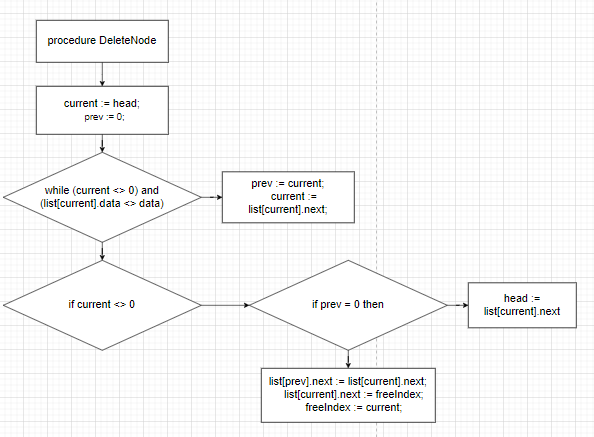
- Вызовите case-меню для работы с односвязным списком.

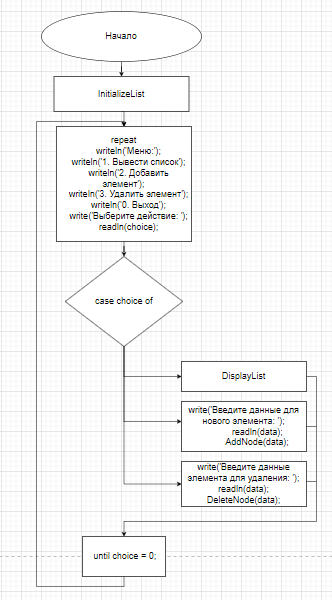
**Схема алгоритма с комментариями:**











**Код программы:**

**type**

Node = **record**

data: integer;

next: integer;

**end**;

**var**

list: **array**[1..10] **of** Node;

head, freeIndex: integer;

**procedure** InitializeList;

**var**

i: integer;

**begin**

**for** i := 1 **to** 10 **do**

**begin**

list[i].next := i + 1;

**end**;

list[10].next := 0; // Последний элемент списка

head := 0; // Голова списка

freeIndex := 1; // Начальный свободный индекс

**end**;

**procedure** DisplayList;

**var**

current: integer;

**begin**

current := head;

writeln('Содержимое списка:');

**while** current <> 0 **do**

**begin**

writeln(list[current].data);

current := list[current].next;

**end**;

**end**;

**procedure** AddNode(data: integer);

**var**

newIndex, current: integer;

**begin**

**if** freeIndex <> 0 **then**

**begin**

newIndex := freeIndex;

freeIndex := list[freeIndex].next;

list[newIndex].data := data;

list[newIndex].next := 0;

**if** head = 0 **then**

head := newIndex

**else**

**begin**

current := head;

**while** list[current].next <> 0 **do**

current := list[current].next;

list[current].next := newIndex;

**end**;

**end**

**else**

writeln('Список полон, невозможно добавить новый элемент.');

**end**;

**procedure** DeleteNode(data: integer);

**var**

current, prev: integer;

**begin**

current := head;

prev := 0;

**while** (current <> 0) **and** (list[current].data <> data) **do**

**begin**

prev := current;

current := list[current].next;

**end**;

**if** current <> 0 **then**

**begin**

**if** prev = 0 **then**

head := list[current].next

**else**

list[prev].next := list[current].next;

list[current].next := freeIndex;

freeIndex := current;

**end**

**else**

writeln('Элемент не найден.');

**end**;

**var**

choice, data: integer;

**begin**

InitializeList;

**repeat**

writeln('Меню:');

writeln('1. Вывести список');

writeln('2. Добавить элемент');

writeln('3. Удалить элемент');

writeln('0. Выход');

write('Выберите действие: ');

readln(choice);

**case** choice **of**

1: DisplayList;

2: **begin**

write('Введите данные для нового элемента: ');

readln(data);

AddNode(data);

**end**;

3: **begin**

write('Введите данные элемента для удаления: ');

readln(data);

DeleteNode(data);

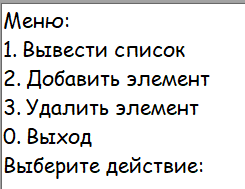
**end**;

**end**;

**until** choice = 0;

**end**.

**Результат выполнения программы:**



**Вывод:**

В ходе выполнения контрольной домашней работы были улучшены навыки в организации времени и постановке приоритетов выполнения задач.

Несмотря на некоторые сложности с оформлением и ограничением по времени, задание было выполнено. Получен положительный опыт и знания. Эта работа помогла развить навыки работы в условиях ограниченного времени и глубже понять изучаемый материал.

Кроме этого, стало ясно, что важно создавать чёткий план работы перед началом выполнения задания. Это позволяет более организованно подходить к решению задач и не тратить время на ненужные действия.

В целом, контрольная работа была полезным занятием, которое помогло мне развить навыки решения задач в ограниченное время и углубить знания в предмете.