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: студент учебной группы

ИСПк-205-52-00

Арбузов Иван

Преподаватель:

Сергеева Елизавета Григорьевна

Киров

2024

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

**Задание (Вариант 1)**

1. ﻿﻿﻿ Написать программу для работы со структурой данных "Двусвязный список”.
2. Структура данных должна быть реализована на основе статической памяти
3. Работа со структурой должна осуществляться с помощью сазе-меню. Предусмотреть наглядную визуализацию содержимого структуры.

Описание алгоритма:

Этот алгоритм реализует двусвязный список (Doubly Linked List) на языке программирования Pascal.

В начале программы определены два типа: NodePtr, который является указателем на узел списка, и Node, который представляет структуру узла списка с полями для данных, указателей на предыдущий и следующий узлы.

Затем определены глобальные переменные head, tail и current, которые будут использоваться для управления списком.

Процедура CreateNode(data: Integer) создает новый узел с заданными данными. Создается новый узел newNode и заполняются его поля: данные и указатели на предыдущий и следующий узлы. Если список пустой (head равен nil), то новый узел становится как головой, так и хвостом списка. В противном случае, новый узел добавляется в конец списка, обновляя указатель на следующий узел у текущего хвоста списка и указатель на предыдущий у нового узла, после чего хвост списка обновляется на новый узел.

Процедура DisplayList() выводит все данные узлов списка, начиная с головы и переходя к следующему узлу через указатель на следующий узел, пока не достигнет конца списка.

Процедура Menu() представляет меню с тремя пунктами: добавить узел, отобразить список и выйти. В зависимости от выбора пользователя выполняются соответствующие действия.

Основная часть программы начинается с инициализации переменных head и tail как nil, после чего вызывается процедура Menu() для взаимодействия с пользователем.

Пользователь может добавлять узлы с данными и отображать список, пока не выберет пункт выхода из программы.

Схема алгоритма:

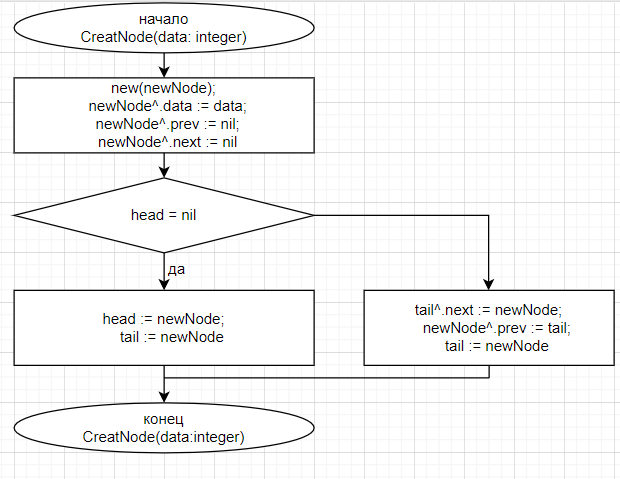


Рисунок 1 – схема алгоритма процедуры CreatNode

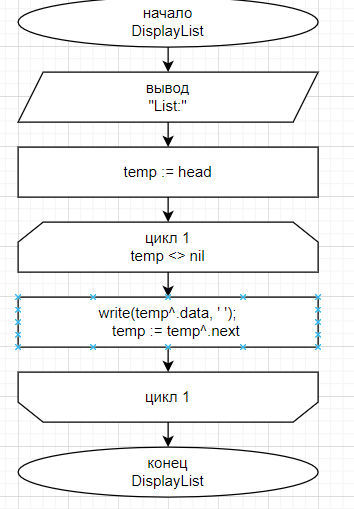


Рисунок 2 – схема алгоритма процедуры DisplayList

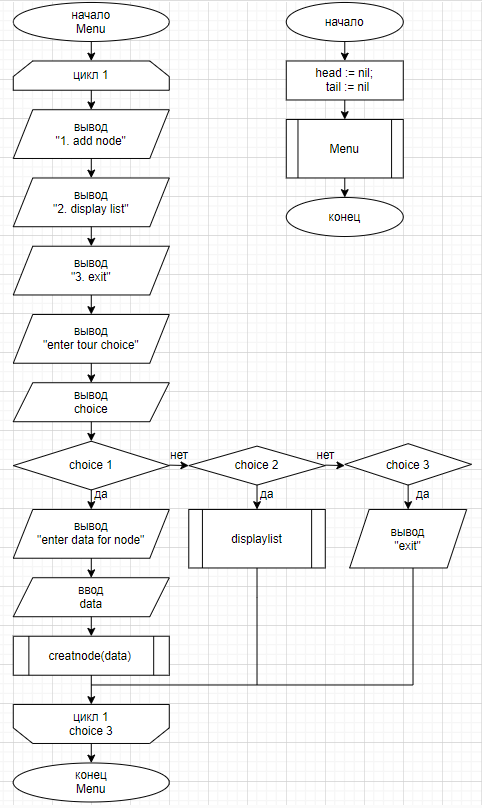


Рисунок 3 – схема алгоритма процедуры меню и процедуры

Код программы:

**program** DoublyLinkedList;

**type**

NodePtr = ^Node;

Node = **record**

data: Integer;

prev, next: NodePtr;

**end**;

**var**

head, tail, current: NodePtr;

**procedure** CreateNode(data: Integer);

**var**

newNode: NodePtr;

**begin**

**new**(newNode);

newNode^.data := data;

newNode^.prev := nil;

newNode^.next := nil;

**if** head = nil **then**

**begin**

head := newNode;

tail := newNode;

**end**

**else**

**begin**

tail^.next := newNode;

newNode^.prev := tail;

tail := newNode;

**end**;

**end**;

**procedure** DisplayList();

**var**

temp: NodePtr;

**begin**

temp := head;

write('List: ');

**while** temp <> nil **do**

**begin**

write(temp^.data, ' ');

temp := temp^.next;

**end**;

writeln;

**end**;

**procedure** Menu();

**var**

choice, data: Integer;

**begin**

**repeat**

writeln('1. Add node');

writeln('2. Display list');

writeln('3. Exit');

write('Enter your choice: ');

readln(choice);

**case** choice **of**

1: **begin**

write('Enter data for node: ');

readln(data);

CreateNode(data);

**end**;

2: DisplayList();

**end**;

**until** choice = 3;

**end**;

**begin**

head := nil;

tail := nil;

Menu();

**end**.

Результаты выполнения работы:

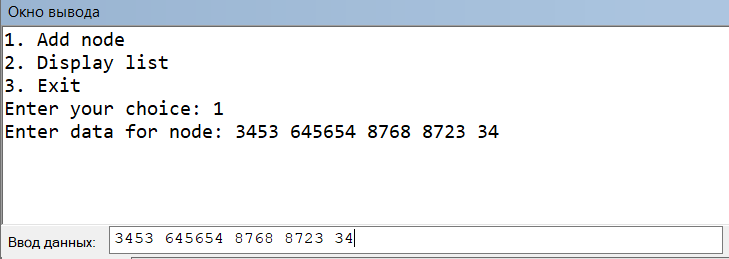


Рисунок 4 – результат выполнения программы при выборе 1 пункта

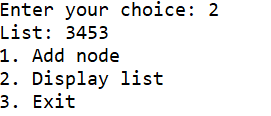


Рисунок 5 – результат при выборе 2 пункта

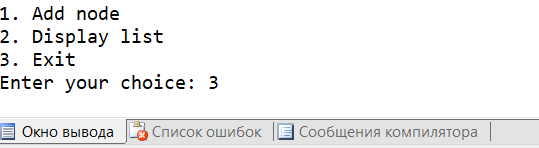


Рисунок 6 – вывод при выполнении 3 пункта

Вывод: в результате изучения принципов работы с базовыми структурами данных и приобретения навыков организации базы данных-меню, студент приобретет необходимые навыки и знания для эффективного управления информацией, обработки данных и построения сложных запросов. Такие навыки могут быть полезными как при работе с базами данных в рамках профессиональной деятельности, так и при разработке программного обеспечения, ведении статистических исследований и анализе данных. В целом, изучение данных принципов позволит студенту более эффективно и качественно работать с информацией, улучшить свои навыки программирования и аналитические способности.

Эти навыки также могут быть полезны при создании веб-сайтов, приложений и других проектов, где требуется работа с данными. Понимание базовых структур данных позволит более эффективно реализовывать системы хранения, обработки и анализа информации, что повысит производительность и качество разрабатываемых продуктов. В общем, приобретение навыков работы с базовыми структурами данных и организации баз данных-меню имеет большое значение для развития профессиональных компетенций в области информационных технологий и программирования.