Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Вятский государственный университет»

Колледж ВятГУ

**ОТЧЕТ**

**ПО ДКР №2**

**«Реализация элементарных структур данных на основе статической памяти»**

**«МДК 05.02 Разработка кода информационных систем информационных систем»**

Выполнил: студент учебной группы

ИСПк 206-52-00

Шарипов А.И

Преподаватель:

Сергеева Е.Г.

Киров

2024

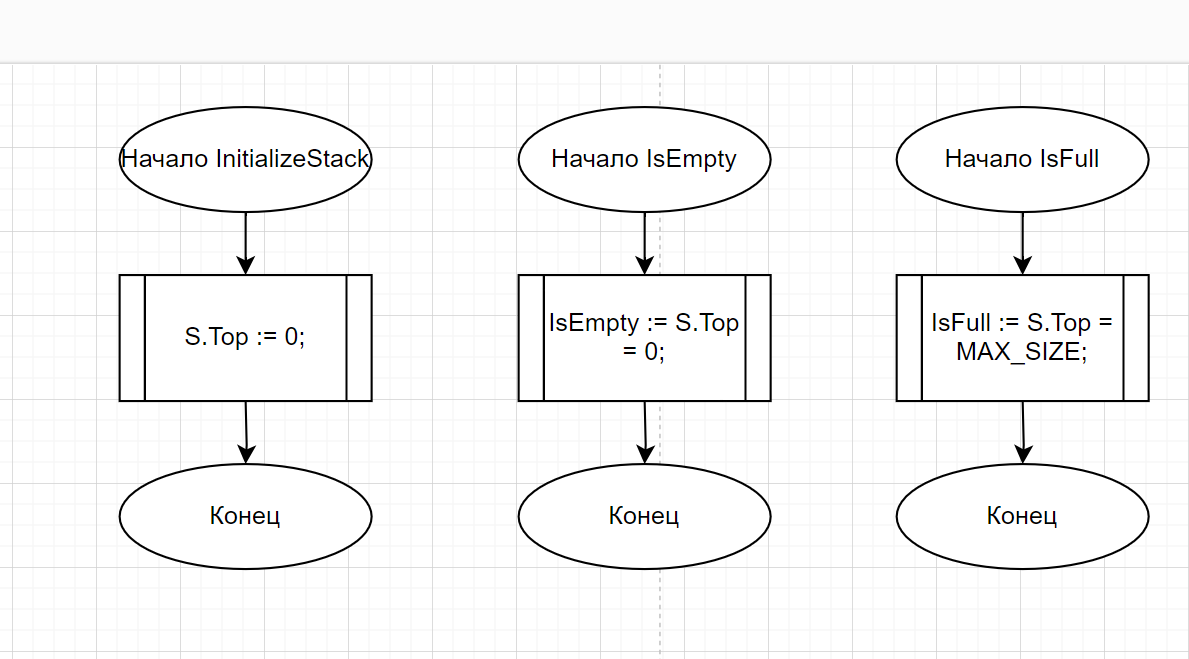
**Вариант 5.**

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

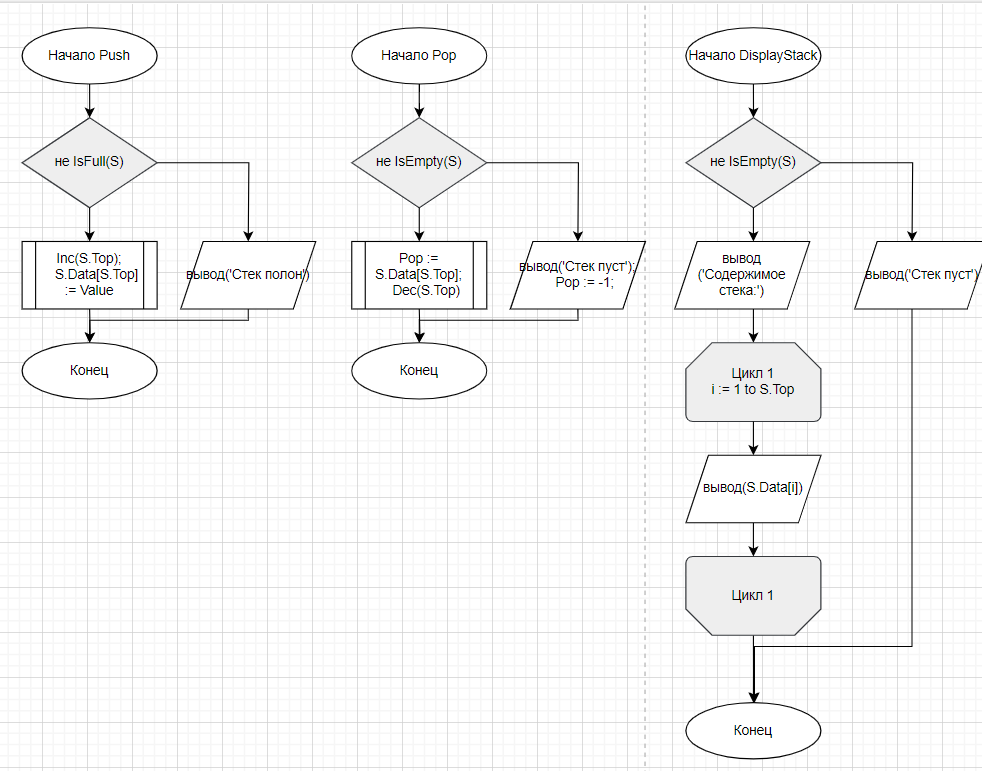
**Задание**:

1. Написать программу для работы со структурой данных «Стек».
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.

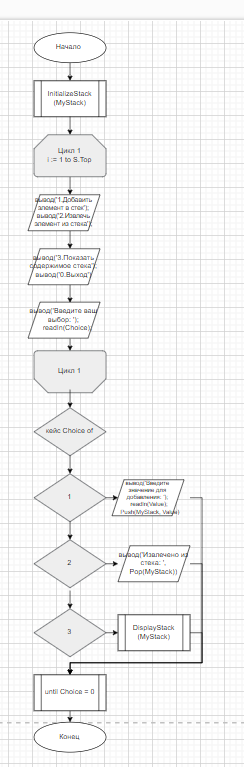
**Схема алгоритма:**

****

**Рисунок 1**

****

**Рисунок 2**



**Рисунок 3**

**Описание алгоритма**

1. В начале программы определены константа MAX\_SIZE (максимальный размер стека) и тип TStack, представляющий стек как запись с массивом целых чисел Data и переменной Top, указывающей на вершину стека.
2. Процедура InitializeStack инициализирует стек, устанавливая Top в 0.
3. Функции IsEmpty и IsFull проверяют, пуст ли стек или заполнен до максимального размера соответственно.
4. Процедура Push добавляет элемент в стек, увеличивая Top и помещая значение в массив Data, если стек не заполнен.
5. Функция Pop извлекает элемент из стека, возвращая его значение и уменьшая Top, если стек не пуст.
6. Процедура DisplayStack выводит содержимое стека, если он не пуст.
7. В основной части программы создается переменная MyStack типа TStack, инициализируется стек с помощью InitializeStack.
8. Затем осуществляется цикл с меню действий: добавление элемента в стек (ввод значения и вызов Push), извлечение элемента из стека (вывод извлеченного значения с помощью Pop), отображение содержимого стека (вывод с помощью DisplayStack).
9. Цикл продолжается до выбора пользователем выхода (Choice = 0).
10. Этот код демонстрирует базовые операции со стеком: добавление, извлечение и отображение элементов. Он позволяет понять принцип работы стека и его основные функции.

**Код программы:**

**program** StackDemo;

**const**

MAX\_SIZE = 10;

**type**

TStack = **record**

Data: **array**[1..MAX\_SIZE] **of** Integer;

Top: Integer;

**end**;

**procedure** InitializeStack(**var** S: TStack);

**begin**

S.Top := 0;

**end**;

**function** IsEmpty(**const** S: TStack): Boolean;

**begin**

IsEmpty := S.Top = 0;

**end**;

**function** IsFull(**const** S: TStack): Boolean;

**begin**

IsFull := S.Top = MAX\_SIZE;

**end**;

**procedure** Push(**var** S: TStack; Value: Integer);

**begin**

**if not** IsFull(S) **then**

**begin**

Inc(S.Top);

S.Data[S.Top] := Value;

**end**

**else**

writeln('Стек полон');

**end**;

**function** Pop(**var** S: TStack): Integer;

**begin**

**if not** IsEmpty(S) **then**

**begin**

Pop := S.Data[S.Top];

Dec(S.Top);

**end**

**else**

**begin**

writeln('Стек пуст');

Pop := -1;

**end**;

**end**;

**procedure** DisplayStack(**const** S: TStack);

**var**

i: Integer;

**begin**

**if not** IsEmpty(S) **then**

**begin**

writeln('Содержимое стека:');

**for** i := 1 **to** S.Top **do**

writeln(S.Data[i]);

**end**

**else**

writeln('Стек пуст');

**end**;

**var**

MyStack: TStack;

Choice, Value: Integer;

**begin**

InitializeStack(MyStack);

**repeat**

writeln('1. Добавить элемент в стек');

writeln('2. Извлечь элемент из стека');

writeln('3. Показать содержимое стека');

writeln('0. Выход');

write('Введите ваш выбор: ');

readln(Choice);

**case** Choice **of**

1:

**begin**

write('Введите значение для добавления: ');

readln(Value);

Push(MyStack, Value);

**end**;

2: writeln('Извлечено из стека: ', Pop(MyStack));

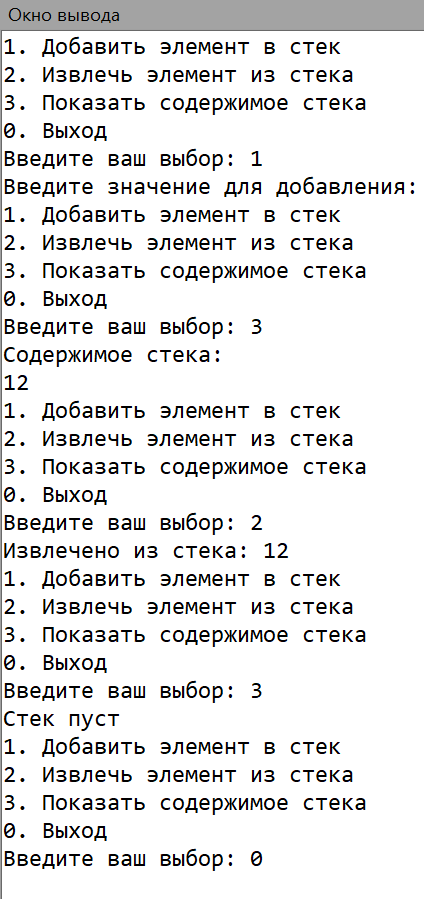
3: DisplayStack(MyStack);

**end**;

**until** Choice = 0;

**end**.

**Результат выполнения программы:**



**Вывод по работе:**

В результате выполнения данного задания, я получил навыки работы с базовыми структурами данных, получил навыки организации case-меню. Разработав программу для работы со структурой данных «Стек». Это позволило мне попрактиковаться в работе со структурами данных и изучить их основы. Одновременно я улучшил свои навыки программирования.

**Ответы:**

1. Структура Сте

Стек — это линейная структура данных, которая следует принципу Last-In-First-Out (LIFO). Это означает, что последний добавленный элемент (называемый вершиной) является первым, который будет удален.

Свойства стека:

LIFO (Last-In-First-Out): Элементы удаляются в обратном порядке их добавления.

Ограниченный размер: Размер стека ограничен доступной памятью.

Быстрые операции push и pop: Операции push и pop выполняются за постоянное время O(1).

2. Хранение стека с помощью массива

Стек можно хранить с помощью массива, где каждый элемент массива представляет собой узел стека. Узлы не связаны явным образом, а вершина стека отслеживается указателем на последний элемент массива.

3. Основные операции со стеком

Push: Добавляет новый элемент в вершину стека.

Pop: Удаляет и возвращает верхний элемент стека.

Peek: Возвращает верхний элемент стека, не удаляя его.

IsEmpty: Проверяет, пуст ли стек.

IsFull: Проверяет, заполнен ли стек.

Display: Выводит элементы стека на экран.

4. Кейс-меню

Кейс-меню — это структура управления, которая позволяет пользователю выбирать одну из нескольких опций. В случае со стеком кейс-меню можно использовать для предоставления пользователю следующих вариантов:

Push

Pop

Peek

IsEmpty

IsFull

Display

Exit