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 СТАТИСТИЧЕСКОЙ ПАМЯТИ»**

**ПО МДК 05.02 РАЗРАБОТКА КОДА ИНФОРМАЦИОННЫХ СИСТЕМ**

Выполнил: студент учебной группы

ИСПк- 101-51-00

Перминов Артём Валерьевич

Преподаватель:

Сергеева Елизавета Григорьевна

Киров

2023

**Цель работы:**

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

**Задания:**

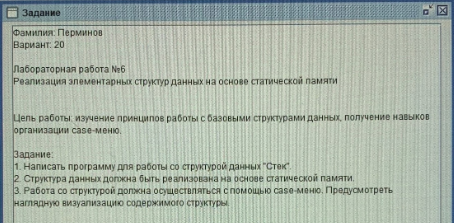
****

Рисунок 1 – Скриншот задания

**Описание алгоритма**:

Стек — абстрактный тип данных, представляющий собой список элементов, организованных по принципу LIFO (англ. last in — first out, «последним пришёл — первым вышел»). Это динамическая структура данных, в которой добавление и удаление элементов доступно только с одного конца( с последнего элемента).

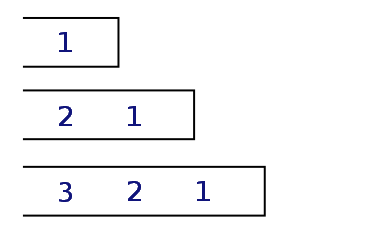


Рисунок 2 – описание алгоритма.

**Код программы**:

**program** StackDemo;

**type**

stack = **record**

Data: **array**[1..100] **of** integer;

Top: integer;

**end**;

**procedure** InitStack(**var** S: Stack);

**begin**

S.Top:=0;

**end**;

**function** IsEmpty(S:Stack): boolean;

**begin**

**if** S.Top = 0 **then**

IsEmpty:= true

**else**

IsEmpty:= false;

**end**;

**function** IsFull(S:Stack): boolean;

**begin**

**if** S.Top = 100 **then**

IsFull:= true

**else**

IsFull:= false;

**end**;

**procedure** Push(**var** S: Stack; x:integer);

**begin**

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

S.Top:= S.Top +1;

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

**end**

**else begin**

writeln('Stack overflow');

**end**;

**end**;

**function** Pop(**var** S:Stack):integer;

**var**

x: integer;

**begin**

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

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

S.Top:= S.Top - 1;

Pop:= x;

**end**

**else begin**

writeln('Stack underflow');

Pop:= 0;

**end**;

**end**;

**procedure** PrintStack(S: Stack);

**var**

i:integer;

**begin**

writeln('Stack contents:');

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

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

writeln(S.Data[i]);

**end**

**else begin**

writeln('Stack is empty');

**end**;

**end**;

**var**

S: Stack;

choice, x:integer;

**begin**

initStack(S);

**repeat**

writeln('Stack menu:');

writeln('1.Push');

writeln('2.Pop');

writeln('3.Print');

writeln('4.Exit');

write('Введите свой выбор:');

readln(choice);

**case** choice **of**

1:**Begin**

write('Введите элемент для нажатия:');

readln(x);

Push(S,x);

**end**;

2:**begin**

x:= Pop(S);

writeln('Вышедший элемент:',x);

**end**;

3:**begin**

PrintStack(S);

**end**;

4:**begin**

writeln('Выход из программы');

**end**;

**else begin**

writeln('Неверный выбор');

**end**;

**end**;

**until** choice =4;

**end**.

**Результат выполнения программы:**

****

Рисунок 3 – case-меню программы

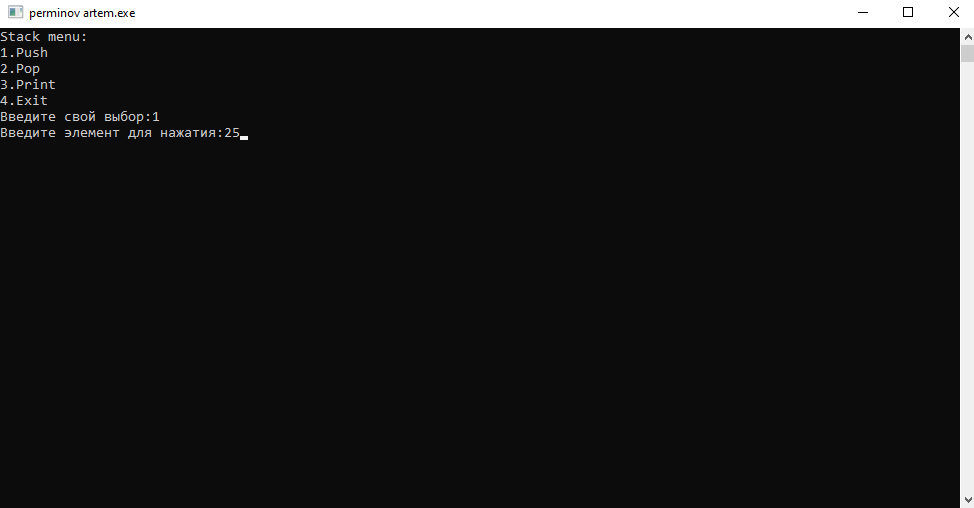


Рисунок 4 – добавление элемента в очередь

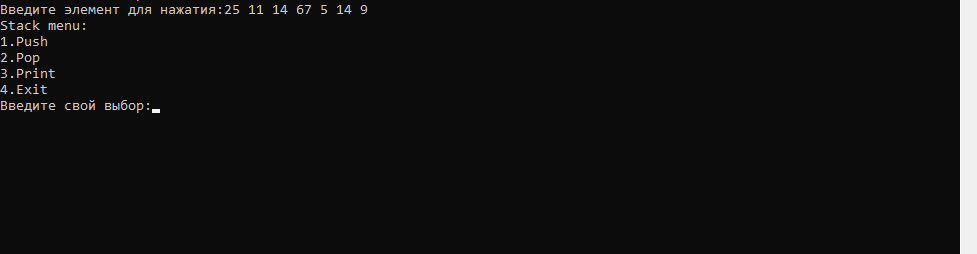


Рисунок 5 – Визуализация содержимого структуры



Рисунок 6 – Удаление элемента

**Вывод:**

В ходе выполнения данной домашней контрольной работы, были получены знания об базовой структуре данных. Был получен опыт в реализации динамической структуры данных “Стек”. Также в ходе домашней контрольной работы пригодились полученные ранее знания об организации case-меню.

В ходе выполнения домашней контрольной работы №6 возникли некоторые трудности. Но были успешно решены повторением пройденного материала.

В итоге были изучены принципы реализации элементарных структур данных на основе статической памяти, и пригодились знания о организации case-меню