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-204-52-00

Беляев Данил Игоревич

Преподаватель:

Сергеева Елизавета Григорьевна

Киров

2023

**Цель работы**

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

**Скриншот задания**

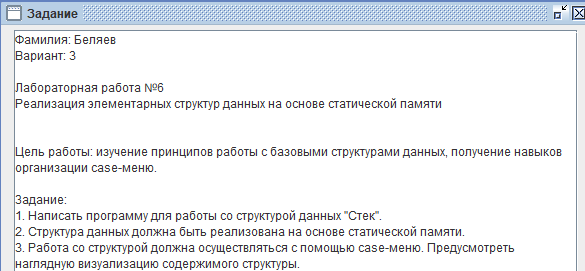


Рисунок 1 – скриншот задания

**Описание алгоритма**

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 визуализацию содержимого структуры.

Процедура Push (Добавление на вершину стека). Работает так: передвигаем указатель с последнего занятого элемента в массиве(верхнего в стеке) на первый свободный элемент массива (т.е. увеличиваем size на 1). Потом добавляем в эту пустую ячейку новый элемент. Таким образом size вновь указывает на верхний элемент стека.

Процедура Pop (Удаление верхнего элемента стека). Просто уменьшаем текущий размер, ничего не делая с последним элементом. Этот последний элемент будет уже как бы вне массива, поэтому мешать не будет.

Функция Top (Получения значения верхнего элемента стека). Т.к. мы знаем индекс элемента, который находится на вершине стека (этот индекс size), то просто возвращаем элемент с данным индексом.

Функция Exit – Выход, функция Print – Вывести введённые символы

**Код программы**

DKR6\_.pas:

**Program** dkr\_6;

**uses** crt;

**const**

MAX\_SIZE = 100;

**type**

Stackk = **array**[1..MAX\_SIZE] **of** integer;

**var**

stack: Stackk;

top: integer;

**procedure** push(value: integer);

**begin**

**if** top = MAX\_SIZE **then**

writeln('Массив полный')

**else**

**begin**

inc(top);

stack[top] := value;

**end**;

**end**;

**procedure** pop;

**begin**

**if** top = 0 **then**

writeln('Массив пустой')

**else**

**begin**

stack[top] := 0;

dec(top);

**end**;

**end**;

**procedure** printStack;

**var**

i: integer;

**begin**

writeln('Stack:');

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

write(stack[i], ' ');

writeln;

**end**;

**var**

choice, value: integer;

**begin**

top := 0;

**repeat**

writeln('################');

writeln('1. push');

writeln('2. pop');

writeln('3. print');

writeln('4. exit');

writeln('################');

write('Введите нужный параметр: ');

readln(choice);

**case** choice **of**

1:

**begin**

readln(value);

push(value);

**end**;

2: pop;

3: printStack;

4: **break**;

**else**

writeln('Invalid choice');

**end**;

**until** false;

**end**.

**Результат выполнения программы**

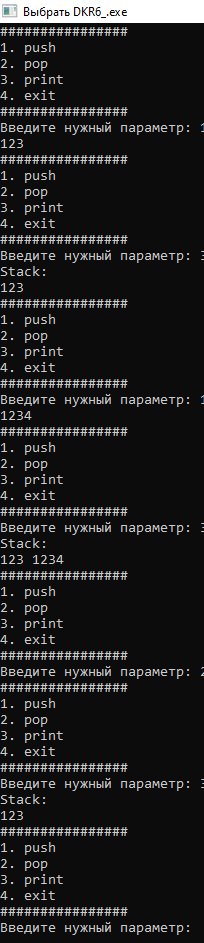


Рисунок 2 – Результат выполнения программы

**Вывод**

Была написана программа для для работы со структурой данных «Стек», также структура данных реализована на основе статической памяти.

Работа со структурой осуществляется с помощью case-меню. Предусмотрена визуализация содержимого структуры.