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-203-52-00

Сюткин Андрей Андреевич

Преподаватель:

Сергеева Елизавета Григорьевна

Киров

2023

1. **Цель домашней контрольной работы:**

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

1. **Скриншот формулировки задания**

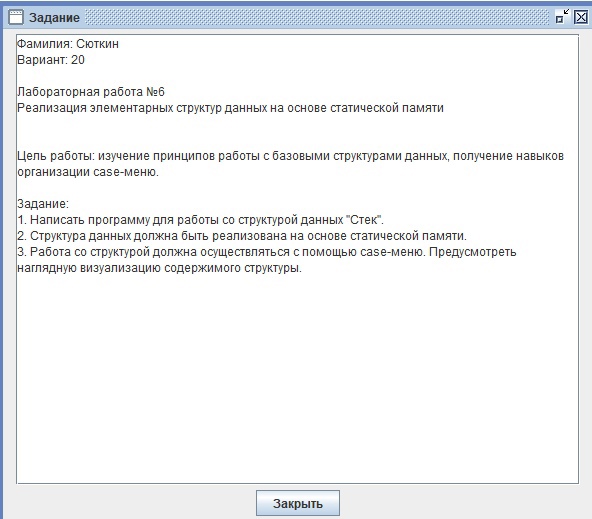


Рисунок 1. Формулировка задания

1. **Описание алгоритма**

Алгоритм работы кода заключается в следующем:

Инициализировать переменную "top" равным 0, чтобы представить пустой стек.

Повторять следующие действия до тех пор, пока пользователь не выберет вариант 6 (Выход):

1) Предоставить пользователю меню возможных действий.

2) Прочитать выбор пользователя.

3) На основе выбора пользователя выполнить одно из следующих действий:

4) Если пользователь выбирает вариант 1 (Добавить элемент), вызвать процедуру "addToStack", которая считает значение нового элемента от пользователя, а затем вызовет процедуру "Push", чтобы добавить элемент в стек.

5) Если пользователь выбирает вариант 2 (Вывод стека), вызвать процедуру "Print", чтобы распечатать содержимое стека.

6) Если пользователь выбирает вариант 3 (Проверка стека на пустоту), вывести, пуст стек или нет.

7) Если пользователь выбирает вариант 4 (Очистить стек), вызвать процедуру "clearStack", чтобы установить "top" равным 0 и очистить стек.

8) Если пользователь выбирает вариант 5 (Взять элемент из стека), вызвать функцию "Pop", чтобы удалить верхний элемент из стека и распечатать его.

После того, как пользователь выберет вариант 6 (Выход), вызвать процедуру "clearStack", чтобы очистить стек и выйти из программы.

1. **Код программы**

**uses** Crt;

**const**

MAX = 100;

**type**

tInf = Integer;

tStack = **array** [1..MAX] **of** tInf;

**var**

top: Integer;

Stack: tStack;

**procedure** Push(**var** Stack : tStack; data : tInf);

**begin**

**if** top = MAX **then**

Writeln('Ошибка')

**else begin**

inc(top);

Stack[top] := data;

**end**;

**end**;

**function** Pop(**var** Stack : tStack) : tInf;

**var**

Res: tInf;

**begin**

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

Writeln('Ошибка')

**else begin**

Res := Stack[top];

dec(top);

Clrscr;

**end**;

Pop := Res;

Writeln('Элемент: ', Res);

**end**;

**procedure** Print(Stack : tStack);

**var**

i: Integer;

**begin**

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

WriteLn('Стек пуст.')

**else begin**

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

Write(Stack[i], ' ');

WriteLn;

**end**;

WriteLn;

**end**;

**procedure** addToStack(**var** Stack : tStack);

**var**

data : tInf;

**begin**

Clrscr;

textcolor(LightGreen);

Write('Введите значение добавляемого элемента: '); ReadLn(data);

Push(Stack, data);

WriteLn;

**end**;

**procedure** clearStack(**var** Stack : tStack);

**begin**

top := 0;

Clrscr;

textcolor(LightMagenta);

WriteLn('Стек очищен');

**end**;

**var**

ch : char;

**begin**

top := 0;

**repeat**

textcolor(green);

Writeln('Программа для работы со стеком ');

Writeln('Выберите желаемое действие:');

textcolor(red);

Writeln('1) Добавить элемент.');

textcolor(yellow);

Writeln('2) Вывод стека.');

textcolor(blue);

Writeln('3) Проверка стека на пустоту');

textcolor(brown);

Writeln('4) Очистить стек');

textcolor(cyan);

Writeln('5) Взять элемент из Стека.');

textcolor(red);

Writeln('6) Выход.');

writeln;

readln(ch);

**case** ch **of**

'1' : addToStack(Stack);

'2' : **begin**

Clrscr;

textcolor(LightRed);

Write('Cодержимое стека = ');

Print(Stack);

**end**;

'3' : **begin**

Clrscr;

textcolor(White);

WriteLn('Стек пуст = ', top = 0);

**end**;

'4' : clearStack(Stack);

'5' : Pop(Stack);

**end**;

**until** ch='6';

clearStack(Stack);

**end**.

1. **Результат выполнения программы**

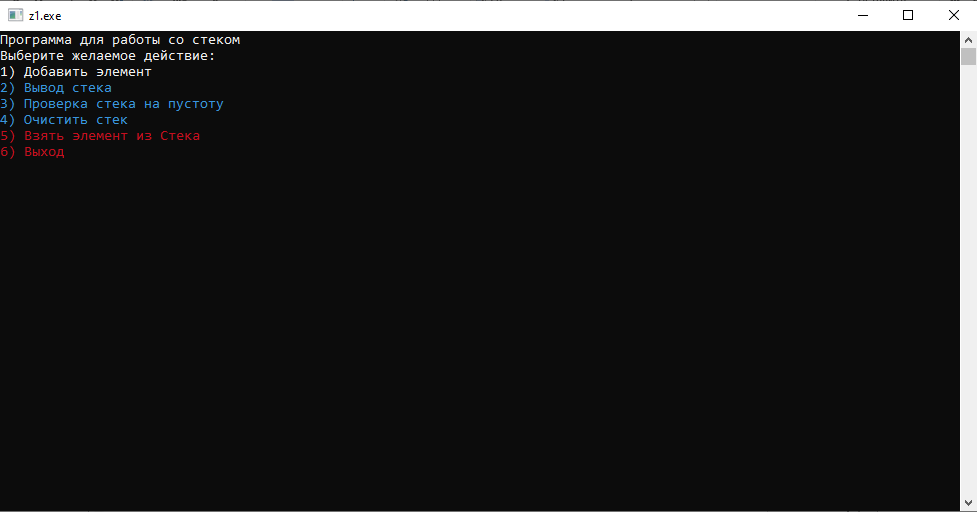


Рисунок 2 – Меню программы

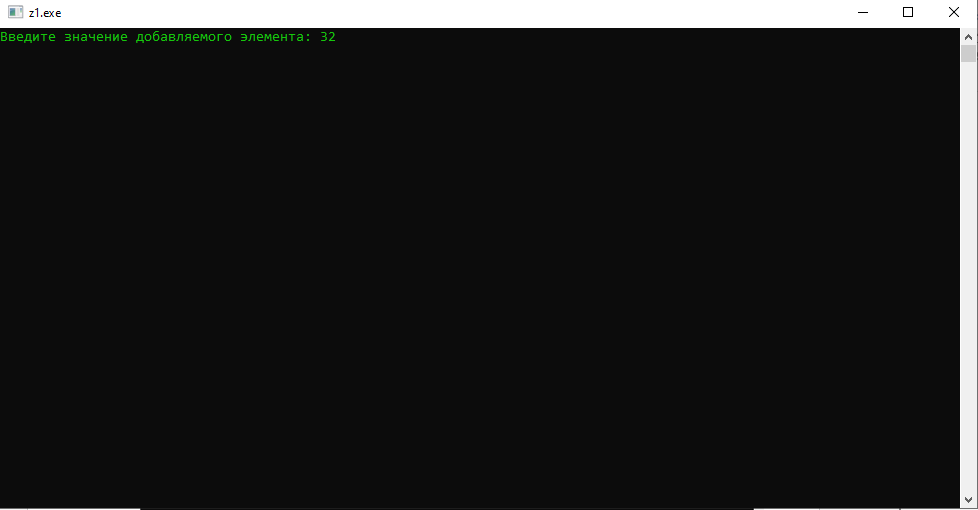


Рисунок 3 – Добавление элемента

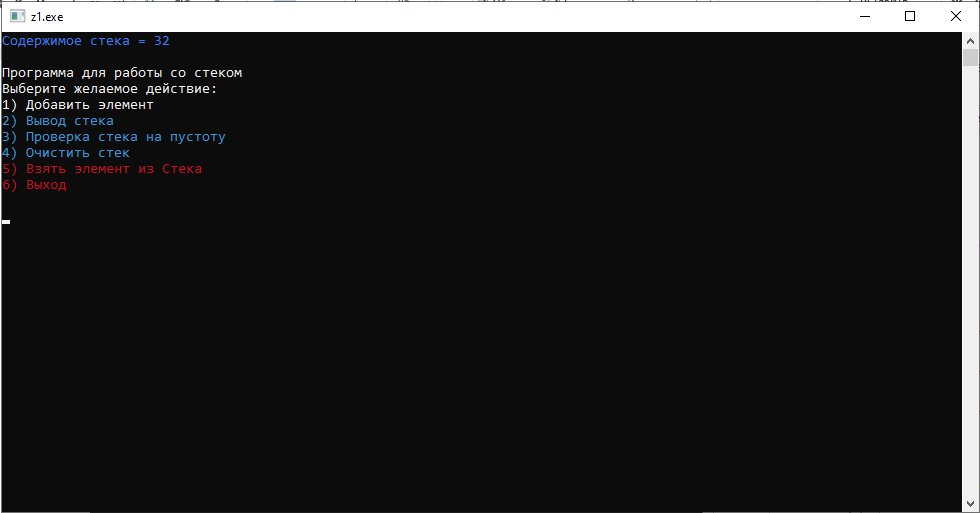


Рисунок 4 – Вывод содержимого стека

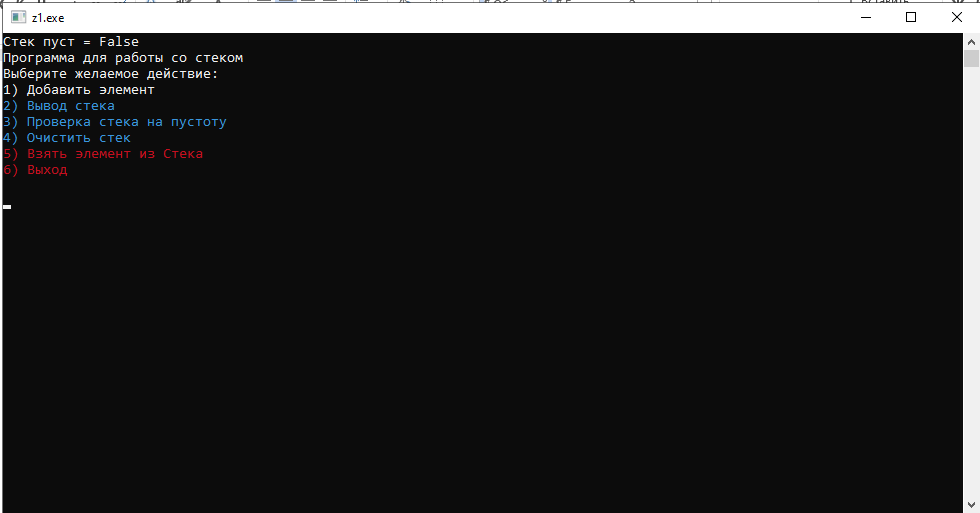


Рисунок 5 – Проверка стека на пустоту

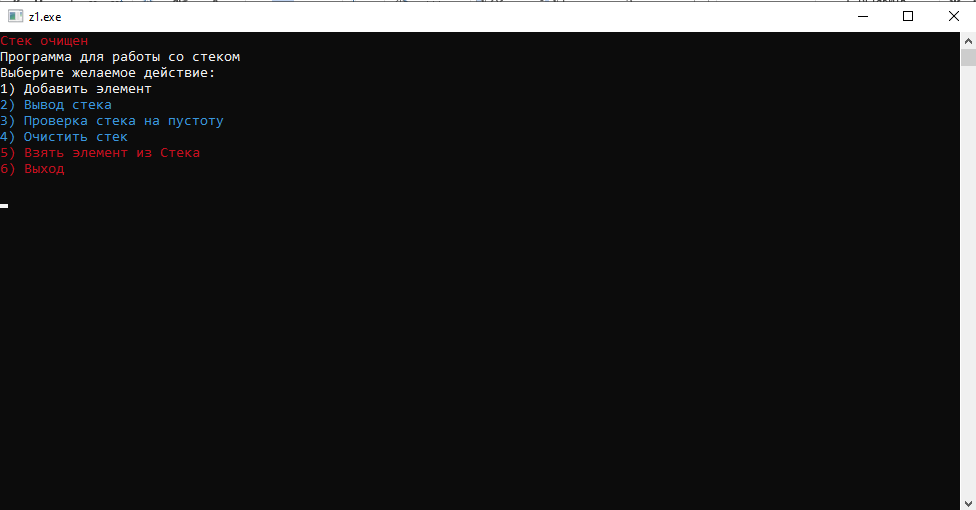


Рисунок 6 – Очистка стека

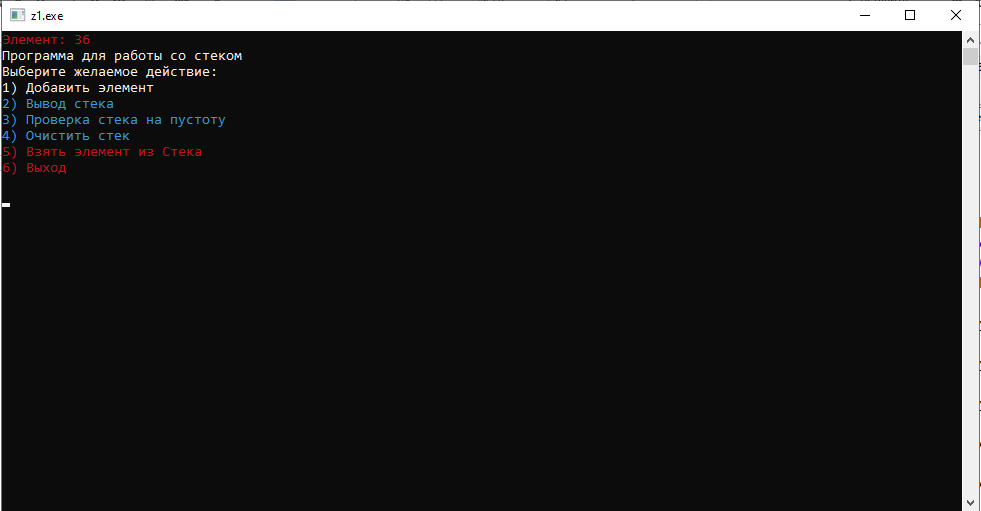


Рисунок 7 – Вывод элемента из стека

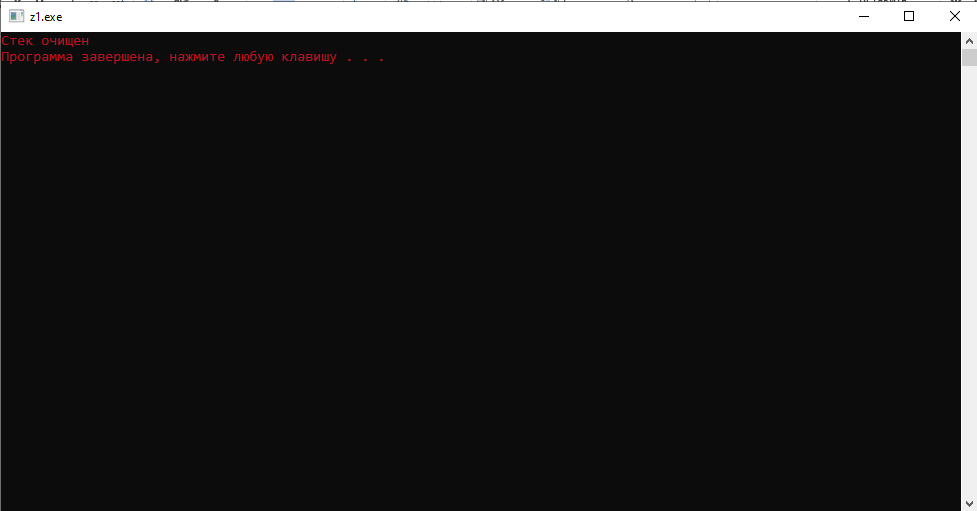


Рисунок 8 – Выход из программы

1. **Вывод**

Поставленная цель работы: Изучение принципов работы с базовыми структурами данных, получение навыков организации case-меню была выполнена. Работа проходила в несколько этапов: цель, формулировка задания, изучение новой информации, написание кода программы, результат выполнения программы, вывод. Так как программа была написана для работы со Стеком, было необходимо узнать, что это такое. Стек – абстрактный тип данных, представляющий собой список элементов, организованных по принципу LIFO (англ. last in – first out, «последним пришёл – первым вышел»). Итак, делая вывод, хочу сказать, что в ходе работы все цели были достигнуты, а полученные в ходе выполнения работы знания будут полезны и применимы в дальнейшем.