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

Лаптев Дмитрий Андреевич

Преподаватель:

Сергеева Елизавета Григорьевна

Киров

2024

1. Цель работы

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

2. Формулировка задания.

Вариант 12.

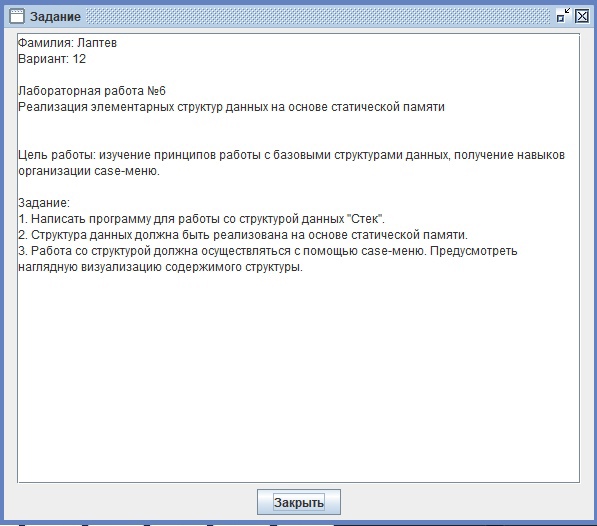


Рисунок 1 – Вариант работы.

3. Описание алгоритма.

Данный алгоритм реализует структуру стека с возможностью выбора между статической и динамической реализацией.

1. В начале программа запрашивает пользователя выбрать структуру стека (статическая или динамическая) и инициализирует соответствующие переменные и указатели.

2. Пользователь может добавлять элементы в стек, удалять элементы из стека, проверять, пуст ли стек, и просматривать содержимое стека.

3. В зависимости от выбранной структуры (статической или динамической), добавление элементов и удаление элементов реализовано через разные процедуры.

4. При добавлении элемента программа добавляет его в стек, увеличивая размер стека или изменяя указатели, в зависимости от типа стека.

5. При удалении элемента программа удаляет его из стека, уменьшая размер стека или освобождая память, если это динамический стек.

6. Проверка на пустоту стека осуществляется путем проверки размера стека или наличия элементов в динамическом стеке.

7. Просмотр содержимого стека выводит элементы стека на экран.

Этот алгоритм предоставляет базовый функционал работы со стеком и дает пользователю возможность выбрать удобный для него тип структуры стека.

4. Схема.

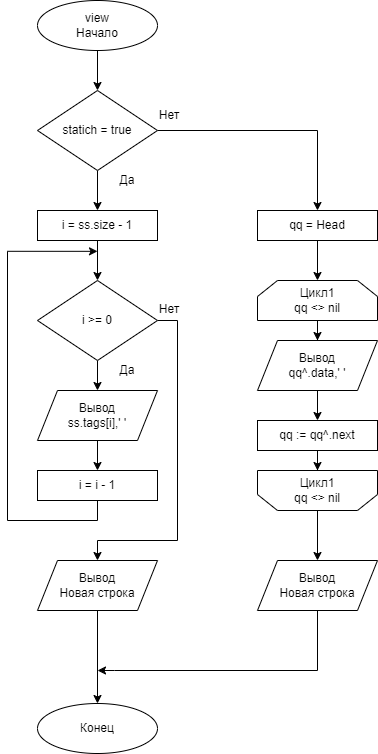


Рисунок 2 – процедура «View».

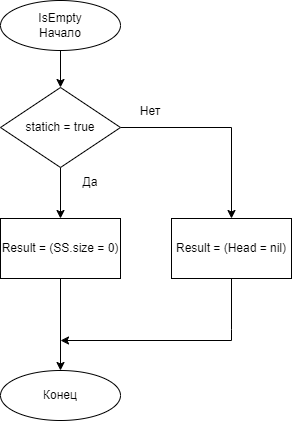


Рисунок 3 – функция «IsEmpty».

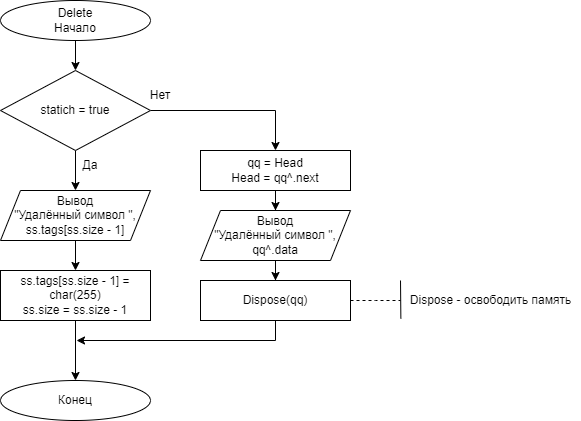


Рисунок 4 – процедура «Delete».

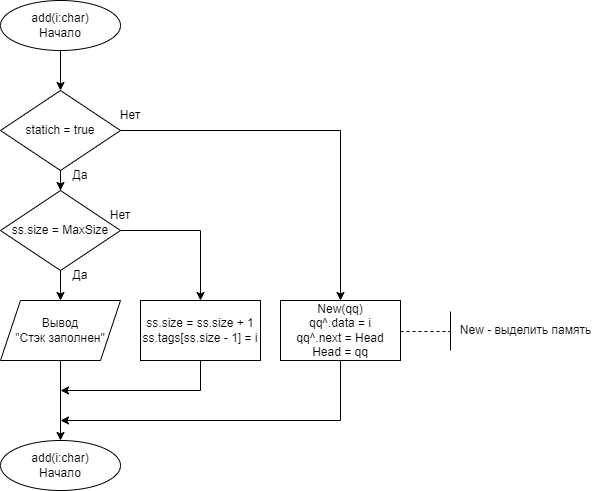


Рисунок 5 – процедура «add».

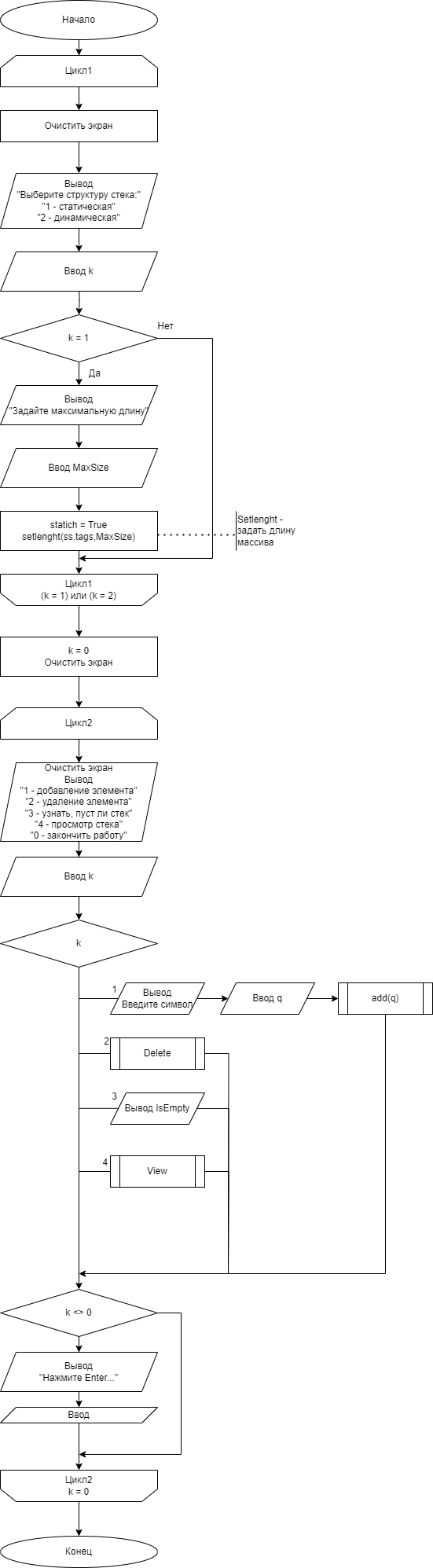


Рисунок 6 – основная программа.

5. Код программы.

uses crt;

type StatStack = record

tags: array of char;

size: integer;

end;

type DinStack = ^Node;

Node = record

data: char;

next: DinStack;

end;

var

SS:StatStack;

MaxSize,K:integer;

Statich: boolean;

q:char;

Head: DinStack;

procedure view();

var i:integer;

qq: DinStack;

begin

If Statich = True then

begin

for i:=ss.size-1 downto 0 do

write(ss.tags[i],' ');

writeln;

end

else

begin

qq:=Head;

while qq <> nil do begin

write(qq^.data,' ');

qq := qq^.next;

end;

writeln;

end;

end;

procedure Delete();

var qq: DinStack;

begin

if statich = True then

begin

writeln('Удалённый символ ', SS.tags[ss.size - 1]);

Ss.tags[ss.size - 1] := char(255);

ss.size := ss.size - 1;

end

else

begin

qq := Head;

Head := qq^.next;

writeln('Удалённый символ ', qq^.data);

Dispose(qq);

end;

end;

procedure add(i:char);

var qq: DinStack;

begin

if statich = True then begin

if SS.size = MaxSize then

begin

writeln('Стэк заполнен.');

exit;

end

else

begin

SS.size := SS.size + 1;

SS.tags[SS.size - 1] := i;

end

end

else

begin

New(qq);

qq^.data := i;

qq^.next := Head;

Head := qq;

end;

end;

function IsEmpty: Boolean;

begin

if statich = true then

Result := (SS.size = 0)

else

Result := (Head = nil)

end;

begin

repeat

writeln('Выбирите структуру стека:');

writeln('1 - статическая');

writeln('2 - динамическая');

readln(k);

if k = 1 then begin

statich := True;

writeln('Задайте максимальную длину');

readln(MaxSize);

setlength(ss.tags,MaxSize);

end;

until (k = 1) or (k =2);

k := 0;

clrscr;

repeat

clrscr;

writeln('1 - добавление элемента');

writeln('2 - удаление элемента');

writeln('3 - узнать, пуст ли стек');

writeln('4 - просмотр стека');

writeln('0 - закончить работу');

readln(k);

case k of

1: begin writeln('Введите символ'); readln(q); add(q); end;

2: Delete;

3: writeln(IsEmpty);

4: view;

end;

writeln('Нажмите enter...');

readln(q);

until k = 0;

end.

6. Результат выполнения программы.

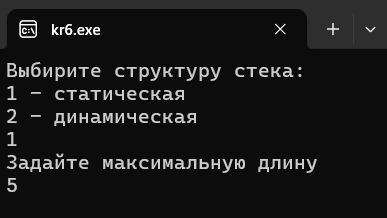


Рисунок 7.

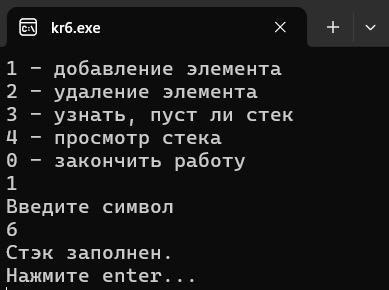


Рисунок 8.

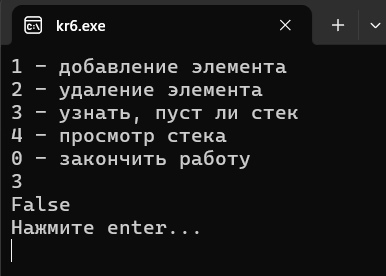


Рисунок 9.

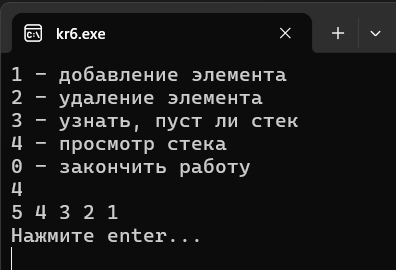


Рисунок 10.

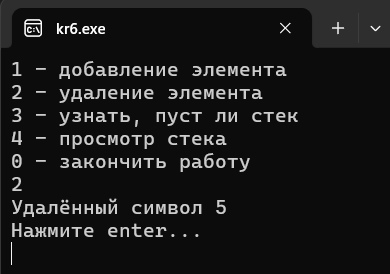


Рисунок 11.

7. Вывод.

Выполняя домашнюю контрольную работу, мы смогли достичь поставленной цели: мы изучили принципы работы с базовыми структурами данных, а также получили навыки организации case-меню.

В ходе работы над задание мы вспомнили и поработали с процедурами и функциями внутри Pascal, вспомнили как работает case-меню.