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-101-51-00

Борисов Константин Александрович

Преподаватель:

Сергеева Елизавета Григорьевна

Киров

2023

**Цель работы:**

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

**Задания:**

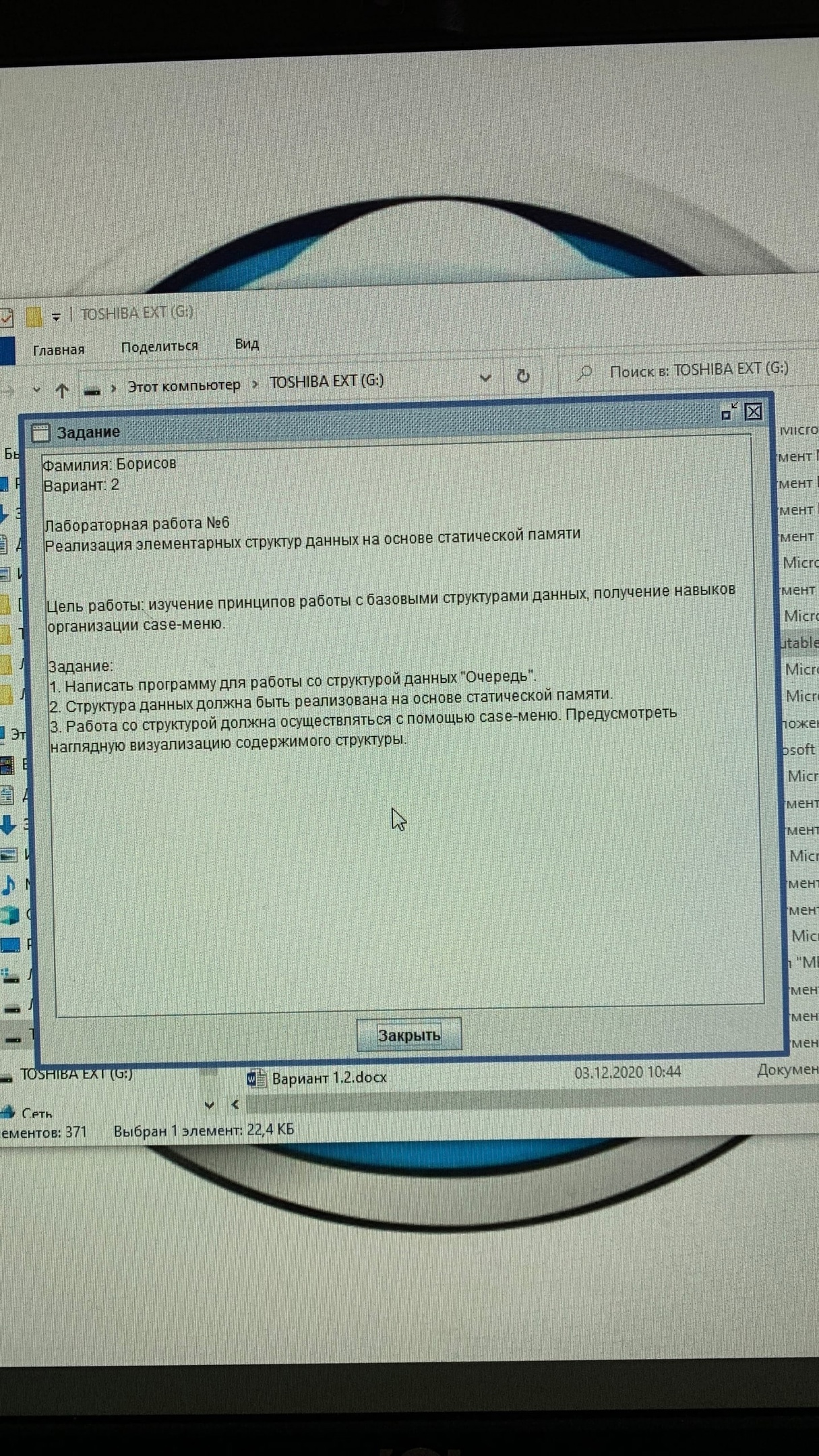


Рисунок 1 – Скриншот задания

**Описание алгоритма:**

Очередь – это структура данных, добавление и удаление элементов в которой происходит путём операций и соответственно. Притом первым из очереди удаляется элемент, который был помещен туда первым, то есть в очереди реализуется принцип “первым вошел — первым вышел” (англ. first-in, first-out — FIFO).

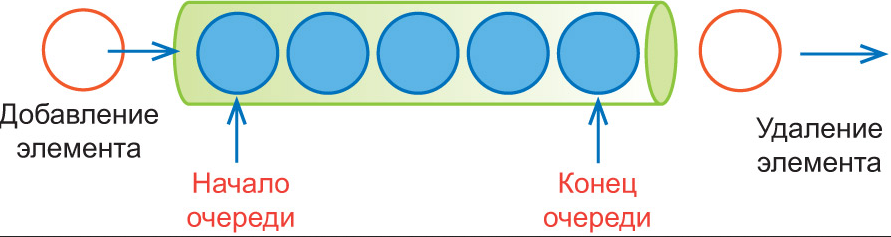
****

Рисунок 2 – Описание алгоритма

**Код программы:**

uses crt;

const

queue\_size = 9;

type

QueueElement = Integer;

QueueArray = array[1..queue\_size] of QueueElement;

QueuePointer = 1..queue\_size;

var

Que: QueueArray;

Front, Rear: QueuePointer;

Element: QueueElement;

a, begin\_queue, end\_queue: Integer;

procedure Queue;

begin

begin\_queue := 1;

end\_queue := 1;

end;

function IsEmpty: Boolean;

begin

IsEmpty := begin\_queue = end\_queue;

end;

function IsFull: Boolean;

begin

IsFull := end\_queue = queue\_size;

end;

procedure Inqueue(Element: QueueElement);

begin

ClrScr;

if IsFull then

PrintLn('Очердь заполнена!')

else

begin

Que[end\_queue] := Element;

end\_queue := end\_queue + 1;

end;

end;

procedure Delqueue;

var

i: QueuePointer;

begin

ClrScr;

if IsEmpty then

PrintLn('Очередь пустая!')

else

begin

Element := Que[begin\_queue];

for i := begin\_queue to end\_queue - 1 do

Que[i] := Que[i + 1];

end\_queue := end\_queue - 1;

end;

end;

procedure PrintQueue;

var

i: QueuePointer;

begin

ClrScr;

Print('Очередь: ');

for i := begin\_queue to end\_queue - 1 do

Write(Que[i], ' ');

WriteLn;

end;

begin

Queue;

repeat

PrintLn('1 - Добавить элемент');

PrintLn('2 - Удалить элемент');

PrintLn('3 - Показать элементы очереди');

PrintLn('0 - Выйти');

Print('Выберите действие: ');

ReadLn(a);

case a of

1: begin

ClrScr;

Print('Введите элемент для добавления в очередь: ');

ReadLn(Element);

Inqueue(Element);

end;

2: Delqueue;

3: PrintQueue;

0: exit;

end;

until a = 0;

end.

**Результат выполнения программы:**

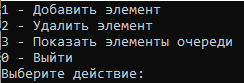
****

Рисунок 3 – Меню программы



Рисунок 4 – Внесение элемента в очередь



Рисунок – Удаление элемента



Рисунок 6 – Вывод всех элементов

**Вывод:**

В ходе выполнения данной домашней контрольной работы, были получены знания об структуре данных на основе статической памяти, добавление и удаление элементов из массива и укреплены навыки работы с case-меню.