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»**

**«МДК 05.02 РАЗРАБОТКА КОДА ИНФОРМАИОННЫХ СИСТЕМ»**

Выполнил: студент учебной группы

ИСПк-101-51-00

Лысков Лев Алексеевич

Преподаватель:

Сергеева Елизавета Григорьевна

Киров

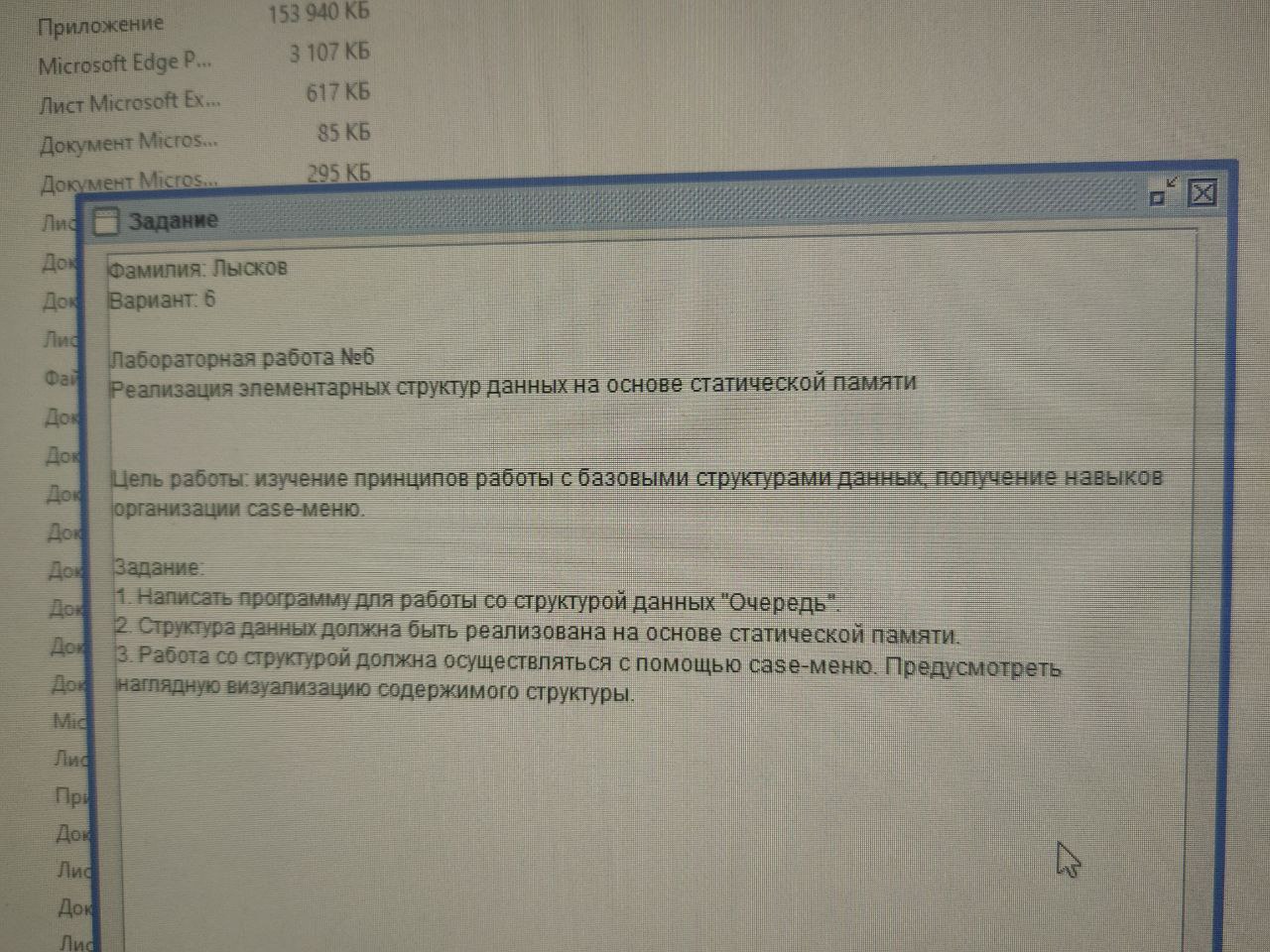
2024

В отчете должны отображаться:

1. Цель работы
2. Формулировка задания (с вариантом)
3. Описание алгоритма
4. Код программы
5. Результат выполнения программы
6. Вывод
7. **Цель работы**

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

1. **Формулировка задания**

****

**3. Описание алгоритма**

Очередь — это структура данных, которая хранит элементы в порядке их добавления, при этом позволяя доступ только к первому добавленному элементу. Очередь работает по принципу "первый вошел, первый вышел".

1. Инициализация очереди:

* Создается структура Queue, которая содержит массив items для хранения элементов очереди и указатели front и rear для отслеживания начала и конца очереди.
* Процедура initializeQueue устанавливает начальные значения front и rear на 0.

2. Проверка на заполненность и пустоту очереди:

* Функции isFull и isEmpty проверяют, заполнена ли очередь и пуста ли она соответственно.

3. Добавление элемента в очередь:

* Процедура enqueue добавляет новый элемент в очередь, увеличивая rear и помещая элемент в соответствующую позицию в массиве items.

4. Удаление элемента из очереди:

* Процедура dequeue удаляет элемент из начала очереди, увеличивая front и тем самым "исключая" элемент из очереди.

5. Вывод содержимого очереди:

* Процедура displayQueue выводит все элементы, находящиеся в очереди, начиная с элемента, следующего за front и заканчивая элементом rear.

6. Основной цикл программы:

* Пользователю предлагается выбрать действие: добавить элемент в очередь, удалить элемент из очереди, вывести содержимое очереди или выйти из программы.
* С помощью конструкции case выбирается соответствующая операция в зависимости от ввода пользователя.
* Цикл продолжается до тех пор, пока пользователь не выберет выход из программы.

**5. Код программы**

**program** QueueExample;

**const**

MAX\_SIZE = 5; // Максимальный размер очереди

**type**

Queue = **record**

items: **array**[1..MAX\_SIZE] **of** Integer;

front, rear: Integer;

**end**;

**procedure** initializeQueue(**var** q: Queue);

**begin**

q.front := 0;

q.rear := 0;

**end**;

**function** isFull(q: Queue): Boolean;

**begin**

isFull := (q.rear = MAX\_SIZE);

**end**;

**function** isEmpty(q: Queue): Boolean;

**begin**

isEmpty := (q.front = q.rear);

**end**;

**procedure** enqueue(**var** q: Queue; data: Integer);

**begin**

**if** isFull(q) **then**

writeln('Очередь переполнена!')

**else**

**begin**

q.rear := q.rear + 1;

q.items[q.rear] := data;

**end**;

**end**;

**procedure** dequeue(**var** q: Queue);

**begin**

**if** isEmpty(q) **then**

writeln('Очередь пуста!')

**else**

**begin**

q.front := q.front + 1;

**end**;

**end**;

**procedure** displayQueue(q: Queue);

**var**

i: Integer;

**begin**

writeln('Содержимое очереди:');

**for** i := q.front + 1 **to** q.rear **do**

write(q.items[i], ' ');

writeln;

**end**;

**var**

myQueue: Queue;

choice, data: Integer;

**begin**

initializeQueue(myQueue);

**repeat**

writeln('1. Добавить элемент в очередь');

writeln('2. Удалить элемент из очереди');

writeln('3. Вывести содержимое очереди');

writeln('0. Выйти');

write('Выберите действие: ');

readln(choice);

**case** choice **of**

1: **begin**

write('Введите элемент для добавления: ');

readln(data);

enqueue(myQueue, data);

**end**;

2: dequeue(myQueue);

3: displayQueue(myQueue);

0: writeln('Выход из программы');

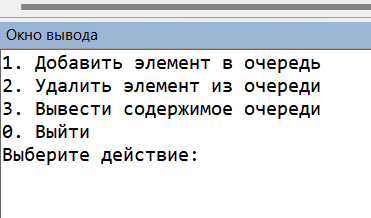
**else** writeln('Некорректный выбор');

**end**;

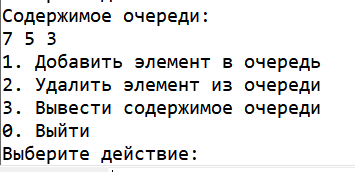
**until** choice = 0;

**end**.

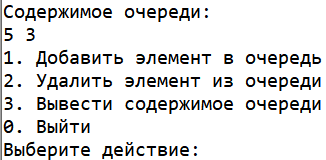
**6. Результаты выполнения программы**



Скриншот 5 - результат выполнения программы.



Скриншот 6 - результат выполнения программы.



Скриншот 7 – результат выполнения программы.

**7. Вывод**

В ходе данной работы была реализована программа для работы со структурой данных «Очередь». В программе можно добавлять и удалять элементы очереди, также показывать её содержимое, всё это осуществляется с помощью case-меню.