Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

«**Пермский национальный исследовательский политехнический университет»**

Кафедра «Информационные технологии и автоматизированные системы»

**ОТЧЕТ**

Дисциплина: «Основы алгоритмизации и программирования»

Тема : "Очередь"

Выполнил работу:

Студент группы РИС-22-1Б

Батин В.В.

Проверил:

Доцент кафедры ИТАС

Полякова О.А.

Г. Пермь-2023

**Постановка задачи.**

Вариант 5:

Сформировать очередь, добавить в неё элементы, удалить из неё элементы.

**Анализ задачи**

Метод enqueue добавляет новый элемент в конец очереди.

Метод dequeue удаляет элемент из начала очереди и возвращает его значение.

Метод peek возвращает значение элемента в начале очереди, не удаляя его.

Метод isEmpty проверяет, пуста ли очередь.

**Код.**

#include <iostream>

template <typename T>

class Queue {

private:

struct Node {

T data;

Node\* next;

Node\* prev;

};

Node\* front;

Node\* rear;

public:

Queue() {

front = nullptr;

rear = nullptr;

}

~Queue() {

while (!isEmpty()) {

dequeue();

}

}

void enqueue(T item) {

Node\* newNode = new Node;

newNode->data = item;

newNode->next = nullptr;

newNode->prev = rear;

if (isEmpty()) {

front = newNode;

}

else {

rear->next = newNode;

}

rear = newNode;

}

T dequeue() {

if (isEmpty()) {

std::cerr << "Queue is empty" << std::endl;

exit(EXIT\_FAILURE);

}

Node\* temp = front;

T data = temp->data;

if (front == rear) {

front = nullptr;

rear = nullptr;

}

else {

front = front->next;

front->prev = nullptr;

}

delete temp;

return data;

}

T peek() {

if (isEmpty()) {

std::cerr << "Queue is empty" << std::endl;

exit(EXIT\_FAILURE);

}

return front->data;

}

bool isEmpty() {

return front == nullptr;

}

};

int main() {

Queue<int> queue;

queue.enqueue(1);

queue.enqueue(2);

queue.enqueue(3);

std::cout << queue.peek() << std::endl;

queue.dequeue();

std::cout << queue.peek() << std::endl;

queue.dequeue();

std::cout << queue.peek() << std::endl;

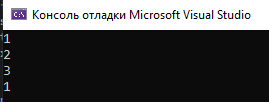
queue.dequeue();

std::cout << queue.isEmpty() << std::endl;

return 0;

}

**Результат выполнения**

****