Министерство образования Республики Беларусь

Учреждение образования

«Белорусский государственный университет

информатики и радиоэлектроники»

Кафедра инженерной психологии и эргономики

­­­­­

Основы алгоритмизации и программирования

Отчет по лабораторной работе №12

«Очереди»

Выполнил: Усов А.М.

Студент группы 310901

Преподаватель: Василькова А. Н.

Минск 2023

Цель: сформировать знания и умения по работе с подпрограммами, приобрести навыки написания программ с использованием очередей.

# Задание 5. Дан набор из 10 чисел. Создать две очереди: первая должна содержать числа из исходного набора с нечетными номерами (1, 3, ..., 9), а вторая — с четными (2, 4, ..., 10); порядок чисел в каждой очереди должен совпадать с порядком чисел в исходном наборе. Вывести указатели на начало и конец каждой из полученных очередей.

#include <iostream>

using namespace std;

template <typename T>

class Queue {

class Node {

T data;

Node\* next;

public:

Node(T data, Node\* next = nullptr) : data(data), next(next) {}

T getData() { return data; }

Node\* getNext() { return next; }

void setNext(Node\* next) { this->next = next; }

void setData(T data) { this->data = data; }

};

Node\* head;

Node\* tail;

int size;

public:

Queue() : head(nullptr), tail(nullptr), size(0) {}

~Queue() {

while (head) {

Node\* temp = head;

head = head->getNext();

delete temp;

}

}

void push(T data) {

if (head == nullptr) {

head = tail = new Node(data);

}

else {

tail->setNext(new Node(data));

tail = tail->getNext();

}

size++;

}

T pop() {

if (head == nullptr) {

throw "Queue is empty";

}

T data = head->getData();

Node\* temp = head;

head = head->getNext();

delete temp;

size--;

return data;

}

T front() {

if (head == nullptr) {

throw "Queue is empty";

}

return head->getData();

}

T shift() {

if (tail == nullptr) {

throw "Queue is empty";

}

return tail->getData();

}

int getSize() {

return size;

}

bool isEmpty() {

return size == 0;

}

};

int main() {

int\* arr = new int[10];

for (int i = 1; i < 11; i++) {

arr[i-1] = i;

}

Queue<int > odd;

Queue<int > even;

for (int i = 0; i < 10; i++) {

if ((i + 1)% 2 == 0) {

even.push(arr[i]);

}

else {

odd.push(arr[i]);

}

}

cout << "Odd queue: " << endl;

cout << "Head: " << odd.front() << endl;

cout << "Tail: " << odd.shift() << endl;

size\_t size1 = odd .getSize();

for (int i = 0; i < size1; i++) {

cout << odd.pop() << " ";

}

cout << endl;

cout << "Even queue: " << endl;

cout << "Head: " << even.front() << endl;

cout << "Tail: " << even.shift() << endl;

size\_t size2 = even.getSize();

for (int i = 0; i < size2; i++) {

cout << even.pop() << " ";

}

cout << endl;

return 0;

}

# Результат работы программы представлен на рисунке 1.

# 

Рисунок 1 – Результат выполнения программы