МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ

“БРЕСТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ”

**КАФЕДРА ИИТ**

ОТЧЁТ

по лабораторной работе №3

**«Стэк и очередь»**

Выполнил:

Студент 1 курса

группы ПО-9

Харитонович Захар Сергеевич

Проверила:

Войцехович О. Ю.

Брест 2022

**Цель работы:** Изучение принципов организации и работы с абстрактной структурой данных СТЭК в форме односвязного линейного списка и с абстрактной структурой данных ОЧЕРЕДЬ в форме односвязного линейного списка.

**Порядок выполнения работы**

*Задание 1*

Разработать консольное приложение, которое с помощью абстрактной структуры данных СТЭК проверяет соответствие открывающих и закрывающих HTML-тэгов во фрагменте HTML кода, введённого с клавиатуры

**Реализация**

#include <iostream>

#include <stack>

using namespace std;

int main() {

string input;

getline(cin, input);

stack<string> tags;

bool isRight = true;

for (int i = 0; i < input.size(); i++) {

if (input[i] == '<') {

string current;

i++;

for (; i < input.length() && input[i] != '>'; i++) {

if (input[i] == ' ') {

while (i < input.length() && input[i] != '>') i++;

break;

}

current += input[i];

}

if (current[0] == '/') {

string temp = current;

current.erase(0, 1);

if (tags.empty()) {

cout << "No pair for <" << temp << ">" << endl;

isRight = false;

continue;

}

if (current == tags.top()) {

tags.pop();

} else {

cout << "No pair for <" << temp << "> or wrong tag closing." << endl;

isRight = false;

continue;

}

} else {

tags.push(current);

}

}

}

while (!tags.empty()) {

cout << "No pair for <" << tags.top() << ">" << endl;

isRight = false;

tags.pop();

}

if (isRight) cout << "Everything is right." << endl;

return 0;

}

*Задание 2*

Даны две непустые очереди, которые содержат одинаковое количество элементов. Объединить очереди в одну, в которой элементы исходных очередей чередуются.

Даны две непустые очереди. Элементы каждой из очередей упорядочены по возрастанию. Объединить очереди в одну с сохранением упорядоченности элементов.

**Реализация**

#include <iostream>

#include <queue>

using namespace std;

queue<int> to\_one\_queue(queue<int>, queue<int>);

queue<int> to\_one\_queue\_sorted(queue<int>, queue<int>);

int main() {

queue<int> A, B;

cout << "A: ";

for (int i = 3; i <= 11; i += 2) {

cout << i << " ";

A.push(i);

}

cout << endl << "B: ";

for (int i = 1; i <= 13; i += 3) {

cout << i << " ";

B.push(i);

}

cout << endl;

queue<int> result;

result = to\_one\_queue(A, B);

cout << "To one queue: ";

while (!result.empty()) {

cout << result.front() << " ";

result.pop();

}

result = to\_one\_queue\_sorted(A, B);

cout << endl << "To one queue sorted: ";

while (!result.empty()) {

cout << result.front() << " ";

result.pop();

}

cout << endl << endl;

return 0;

}

queue<int> to\_one\_queue(queue<int> A, queue<int> B) {

queue<int> total;

while (!A.empty() && !B.empty()) {

total.push(A.front());

A.pop();

total.push(B.front());

B.pop();

}

return total;

}

queue<int> to\_one\_queue\_sorted(queue<int> A, queue<int> B) {

queue<int> total;

while (!A.empty() || !B.empty()) {

if (A.empty()) {

while (!B.empty()) {

total.push(B.front());

B.pop();

}

break;

}

if (B.empty()) {

while (!A.empty()) {

total.push(A.front());

A.pop();

}

break;

}

if (A.front() < B.front()) {

total.push(A.front());

A.pop();

} else {

total.push(B.front());

B.pop();

}

}

return total;

}

**Вывод:** изучены принципы работы с абстрактными структурами данных стэк и очередь путём решения поставленных задач, подразумевающих использование изучаемых структур.