24F-0040

Laiba

Lab07

Class Task

Task 02:

**(task01 is included in weekly task)**

#include<iostream>

#include<string>

using namespace std;

class Queue {

public:

string data;

Queue\* face;

Queue\* rear;

Queue\* next;

string id;

string name;

Queue() {

face = nullptr;

rear = nullptr;

next = nullptr;

}

Queue(string d) {

data = d;

}

void enqueue();

void dequeue();

int if\_Empty();

void peek();

void display();

~Queue() {

delete face;

delete rear;

delete next;

}

};

int Queue::if\_Empty() {

if (face == nullptr && rear == nullptr) {

return true;

}

return false;

}

void Queue::enqueue() {

string i, n;

cout << "Enter id of customer: ";

cin >> i;

cin.ignore();

cout << "Enter name of customer: ";

getline(cin, n);

Queue\* new\_node = new Queue();

new\_node->name = n;

new\_node->id = i;

if (if\_Empty()) {

face = rear = new\_node;

}

else {

rear->next = new\_node;

rear = new\_node;

}

}

void Queue::dequeue() {

if (face == nullptr) {

cout << "Queue is empty!\n";

}

else if (face == rear) {

cout << "Name is: " << face->name << endl;

cout << "ID is: " << face->id;

face = rear = nullptr;

}

else {

Queue\* new\_node = face;

cout << "Name is: " << face->name << endl;

cout << "ID is: " << face->id;

face = face->next;

delete new\_node;

}

}

void Queue::peek() {

if (face == nullptr) {

cout << "Queue is empty\n";

}

else {

cout << "The next customer to dequeue is: ";

cout << "Name is: " << face->name << endl;

cout << "ID is: " << face->id;

}

}

void Queue::display() {

Queue\* new\_node;

new\_node = face;

int c = 0;

cout << "All customers data in Queue is : \n";

if (face == nullptr) {

cout << "Queue is Empty\n";

return;

}

else if (face == rear) {

cout << "Customer " << ++c << " data: \n";

cout << "Name is: " << new\_node->name << endl;

cout << "ID is: " << new\_node->id;

}

else {

while (new\_node != nullptr) {

cout << "Customer " << c++ << "data: \n";

cout << "Name is: " << new\_node->name << endl;

cout << "ID is: " << new\_node->id;

new\_node = new\_node->next;

}

}

}

int main() {

Queue q;

int choice = 1;

while (choice != 0) {

cout << "\nEnter what do you wanna do: ";

cout << "\n1)Add Customer.\n2)Serve Customer.\n3)Next Customer.\n4)Check if Empty.\n5)Display Queue.\n0)End programme.\n";

cin >> choice;

cout << "\n\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n";

switch (choice) {

case 1: {

q.enqueue();

break;

}

case 2: {

q.dequeue();

break;

}case 3: {

q.peek();

break;

}case 4: {

if (q.if\_Empty()) {

cout << "Queue is empty.\n";

}

else {

cout << "Queue is not empty.\n";

}

break;

}case 5: {

q.display();

break;

}case 0: {

return 0;

break;

}default: {

cout << "Invalid input choice\n";

}

}

cout << "\n\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n";

}

system("Pause");

return 0;

}









