Міністерство освіти і науки України

Центральноукраїнський національний технічний університет

Механіко-технологічний факультет

Кафедра кібербезпеки та програмного забезпечення

**Лабораторна робота №1**

**Тема:** «Контейнерні класи. Стандартна бібліотека шаблонів (STL) в С++.»

|  |
| --- |
| Виконав: ст. гр. КН-23 |
| Кобзар М. В. |
| Перевірив: Козірова Н. Л. |
|  |

Кропивницький

2024

**Тема: Контейнерні класи. Стандартна бібліотека шаблонів (STL) в С++.**

**Мета: ознайомитись Контейнерні класи та навчитись їх програмно**

**реалізовувати мовою С++.**

**Завдання:**

**Варіант 8**

Розробіть систему керування поштовими відправленнями. У цій системі

потрібно зберігати та керувати інформацією про відправлення, включаючи

даних про відправника, одержувача, вагу та тип поштового відправлення.

Використовуйте контейнерні класи зі стандартної бібліотеки шаблонів (STL)

для збереження та операцій над даними.

 Додайте декілька відправлень до списку, включаючи дані про

відправника, одержувача, вагу та тип відправлення.

 Виведіть список всіх відправлень.

 Видаліть певне відправлення за номером.

 Оновіть вагу та тип для певного відправлення.

 Виведіть оновлений список всіх відправлень.

Ви маєте розширити програму, додати додаткові методи та функціональні

можливості, які вам здаються відповідними.

**Лістинг :**

#include <iostream>

#include <vector>

#include <string>

#include <algorithm>

#include <limits>

using namespace std;

struct Delivery {

int id;

string sender;

string receiver;

double weight;

string type;

};

class DeliveryManager {

private:

vector<Delivery> deliveries;

int nextId = 1;

public:

void addDelivery(const string& sender, const string& receiver, double weight, const string& type) {

deliveries.push\_back({nextId++, sender, receiver, weight, type});

}

void displayDeliveries() const {

for (const auto& delivery : deliveries) {

cout << "ID: " << delivery.id << ", Sender: " << delivery.sender << ", Receiver: " << delivery.receiver

<< ", Weight: " << delivery.weight << ", Type: " << delivery.type << endl;

}

}

void removeDelivery(int id) {

deliveries.erase(remove\_if(deliveries.begin(), deliveries.end(), [id](const Delivery& d) {

return d.id == id;

}), deliveries.end());

}

void updateDelivery(int id, double weight, const string& type) {

for (auto& delivery : deliveries) {

if (delivery.id == id) {

delivery.weight = weight;

delivery.type = type;

break;

}

}

}

};

void displayMenu() {

cout << "1. Add Delivery" << endl;

cout << "2. Display Deliveries" << endl;

cout << "3. Remove Delivery" << endl;

cout << "4. Update Delivery" << endl;

cout << "5. Exit" << endl;

}

int main() {

DeliveryManager manager;

int choice;

do {

displayMenu();

cout << "Enter your choice: ";

cin >> choice;

if (cin.fail()) {

cin.clear(); // Очищаємо стан потоку вводу

cin.ignore(numeric\_limits<streamsize>::max(), '\n'); // Видаляємо залишок буфера

cout << "Invalid choice. Please try again." << endl;

continue;

}

switch (choice) {

case 1: {

string sender, receiver, type;

double weight;

cout << "Enter sender: ";

cin.ignore();

getline(cin, sender);

cout << "Enter receiver: ";

getline(cin, receiver);

cout << "Enter weight: ";

cin >> weight;

cout << "Enter type: ";

cin.ignore();

getline(cin, type);

manager.addDelivery(sender, receiver, weight, type);

break;

}

case 2:

manager.displayDeliveries();

break;

case 3: {

int id;

cout << "Enter ID to remove: ";

cin >> id;

manager.removeDelivery(id);

break;

}

case 4: {

int id;

double weight;

string type;

cout << "Enter ID to update: ";

cin >> id;

cout << "Enter new weight: ";

cin >> weight;

cout << "Enter new type: ";

cin.ignore();

getline(cin, type);

manager.updateDelivery(id, weight, type);

break;

}

case 5:

cout << "Exiting..." << endl;

break;

default:

cout << "Invalid choice. Please try again." << endl;

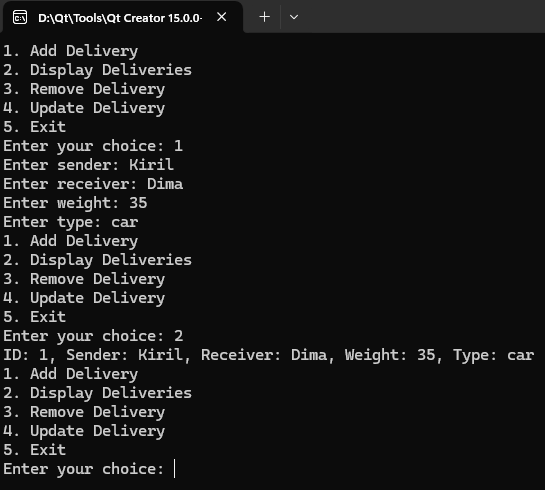
}

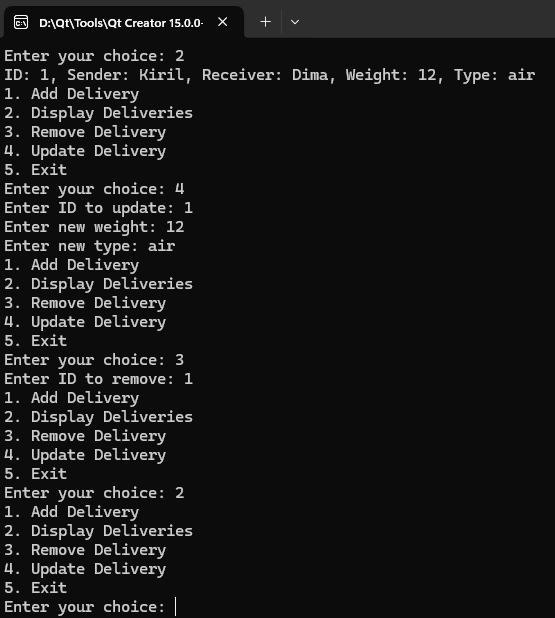
} while (choice != 5);

return 0;

}

**Результат:**



****

**Висновок :**

У рамках цього завдання було розроблено консольну програму для керування поштовими відправленнями .Програма дозволяє додавати нові відправлення, відображати список усіх відправлень, видаляти та оновлювати інформацію про відправлення за допомогою меню у консольному вікні. Ця реалізація забезпечує базові функціональні можливості для управління поштовими відправленнями.