Міністерство освіти і науки України

Центральноукраїнський національний технічний університет

Механіко-технологічний факультет

Кафедра кібербезпеки та програмного забезпечення

Дисципліна: ОБ’ЄКТНО-ОРІЄНТОВАНЕ ПРОГРАМУВАННЯ

**Лабораторна робота №7**

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

|  |
| --- |
| Виконав: ст. гр. КI-23 |
| Карпець О. Р. |
| Перевірив викладач:  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++.

**Варіант 8**

**Завдання**

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 main.cpp:**

#include "mainwindow.h"

#include <QApplication>

int main(int argc, char \*argv[])

{

QApplication a(argc, argv);

MainWindow w;

w.show();

return a.exec();

}

**Лістинг mainwindow.h:**

#ifndef MAINWINDOW\_H

#define MAINWINDOW\_H

#include <QMainWindow>

#include <vector>

#include "shipment.h"

QT\_BEGIN\_NAMESPACE

namespace Ui { class MainWindow; }

QT\_END\_NAMESPACE

using namespace std;

class MainWindow : public QMainWindow

{

Q\_OBJECT

public:

MainWindow(QWidget \*parent = nullptr);

~MainWindow();

void showMessage(QString message);

void showError(QString message);

private slots:

void addShipment();

void deleteShipment();

void updateShipment();

void showShipments();

private:

Ui::MainWindow \*ui;

vector<Shipment> shipments;

int currentId = 1;

};

#endif // MAINWINDOW\_H

**Лістинг mainwindow.cpp:**

#include <QMessageBox>

#include "mainwindow.h"

#include "ui\_mainwindow.h"

MainWindow::MainWindow(QWidget \*parent)

: QMainWindow(parent), ui(new Ui::MainWindow), currentId(1)

{

ui->setupUi(this);

this->setWindowTitle("Деталі відправлення");

connect(ui->addButton, &QPushButton::clicked, this, &MainWindow::addShipment);

connect(ui->deleteButton, &QPushButton::clicked, this, &MainWindow::deleteShipment);

connect(ui->updateButton, &QPushButton::clicked, this, &MainWindow::updateShipment);

connect(ui->showButton, &QPushButton::clicked, this, &MainWindow::showShipments);

}

MainWindow::~MainWindow()

{

delete ui;

}

void MainWindow::addShipment()

{

QString sender = ui->senderEdit->text().trimmed();

QString receiver = ui->receiverEdit->text().trimmed();

double weight = ui->weightEdit->text().toDouble();

QString type = ui->typeEdit->text().trimmed();

if (sender.isEmpty() || receiver.isEmpty() || type.isEmpty() || weight <= 0) {

showError("Заповніть усі поля");

return;

}

shipments.emplace\_back(currentId++, sender, receiver, weight, type);

showMessage("Відправлення успішно додано!");

showShipments();

}

void MainWindow::deleteShipment()

{

int id = ui->idEdit->text().toInt();

auto it = find\_if(shipments.begin(), shipments.end(), [id](Shipment s)

{

return s.getId() == id;

});

if (it != shipments.end()) {

shipments.erase(it);

showMessage("Відправлення успішно видалено!");

} else {

showError("Відправлення з таким ідентифікатором не знайдено!");

}

showShipments();

}

void MainWindow::updateShipment()

{

int id = ui->idEdit->text().toInt();

QString newSender = ui->senderEdit->text().trimmed();

QString newReceiver = ui->receiverEdit->text().trimmed();

double newWeight = ui->weightEdit->text().toDouble();

QString newType = ui->typeEdit->text().trimmed();

auto it = find\_if(shipments.begin(), shipments.end(), [id](Shipment s)

{

return s.getId() == id;

});

if (it != shipments.end()) {

it->setSender(newSender);

it->setReceiver(newReceiver);

it->setWeight(newWeight);

it->setType(newType);

showMessage("Відправлення успішно оновлено!");

} else {

showError("Відправлення з таким ідентифікатором не знайдено!");

}

showShipments();

}

void MainWindow::showShipments()

{

ui->listWidget->clear();

for (const auto &shipment : shipments) {

ui->listWidget->addItem(

QString("ID: %1 | Відправник: %2 | Отримувач: %3 | Вага: %4 | Тип: %5")

.arg(shipment.getId())

.arg(shipment.getSender())

.arg(shipment.getReceiver())

.arg(shipment.getWeight())

.arg(shipment.getType())

);

}

}

void MainWindow::showMessage(QString message)

{

QMessageBox::information(this, "Успіх", message);

}

void MainWindow::showError(QString message)

{

QMessageBox::warning(this, "Помилка", message);

}

**Лістинг shipment.h:**

#ifndef SHIPMENT\_H

#define SHIPMENT\_H

#include <QString>

class Shipment {

public:

Shipment(int id, QString sender, QString receiver, double weight, QString type);

int getId() const;

QString getSender() const;

QString getReceiver() const;

double getWeight() const;

QString getType() const;

void setSender(QString newSender);

void setReceiver(QString newReceiver);

void setWeight(double newWeight);

void setType(QString newType);

private:

int id;

QString sender;

QString receiver;

double weight;

QString type;

};

#endif // SHIPMENT\_H

**Лістинг shipment.cpp:**

#include "shipment.h"

Shipment::Shipment(int id, QString sender, QString receiver, double weight, QString type)

: id(id), sender(sender), receiver(receiver), weight(weight), type(type) {}

int Shipment::getId() const {

return id;

}

QString Shipment::getSender() const {

return sender;

}

QString Shipment::getReceiver() const {

return receiver;

}

double Shipment::getWeight() const {

return weight;

}

QString Shipment::getType() const {

return type;

}

void Shipment::setSender(QString newSender)

{

sender = newSender;

}

void Shipment::setReceiver(QString newReceiver)

{

sender = newReceiver;

}

void Shipment::setWeight(double newWeight) {

weight = newWeight;

}

void Shipment::setType(QString newType) {

type = newType;

}

**Висновок**

У ході роботи я створив заголовковий файл класу MainWindow для Qt-застосунку. Реалізував базову структуру інтерфейсу з використанням QMainWindow та контейнера простіру імен std (vector) для зберігання об'єктів типу Shipment. Додав методи для керування даними: додавання, видалення, оновлення та відображення списку.

Отримав навички роботи з класами Qt, контейнерами C++ та організацією структури програми з використанням механізму слотів.