Міністерство освіти і науки України

Центральноукраїнський національний технічний університет

Механіко-технологічний факультет

Кафедра кібербезпеки та програмного забезпечення

ЗВІТ

ПРО ВИКОНАННЯ ЛАБОРАТОРНОЇ РОБОТИ №7

з дисципліни «Об’єктно-орієнтоване програмування»

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

|  |
| --- |
| Виконав: ст. гр. КН-23 |
| Гребенюк Д.О. |
| Перевірив: асистент  Козірова Н.Л. |
|  |

Кропивницький

2024

**Мета:** Ознайомитись Контейнерні класи та навчитись їх програмно реалізовувати мовою С++.

**Варіант 4**

**Завдання 1**

Розробіть систему управління складом магазину автозапчастин. У цій системі потрібно зберігати та керувати інформацією про наявні автозапчастини. Використовуйте контейнерні класи зі стандартної бібліотеки шаблонів (STL) для збереження та операцій над списком запчастин.

• Додайте декілька запчастин до складу, включаючи їх назву, виробника, ціну та кількість на складі.

• Виведіть список запчастин на складі.

• Видаліть одну або декілька запчастин зі складу за назвою.

• Оновіть ціну та кількість для певної запчастини.

• Виведіть оновлений список запчастин на складі. Ви маєте розширити програму, додати додаткові методи та функціональні можливості, які вам здаються відповідними.

**Завдання 1**

*Лістинг main.cpp*

#include <iostream>  
  
#include "warehouse.h"  
  
using namespace std;  
  
int main()  
{  
 system("chcp 65001 & cls");  
  
 Warehouse warehouse;  
  
 warehouse.addPart(Part("Гальмівні колодки", "Brembo", 1500.0, 20));  
 warehouse.addPart(Part("Масляний фільтр", "Bosch", 300.0, 50));  
 warehouse.addPart(Part("Свічки запалювання", "Denso", 200.0, 100));  
  
 std::cout << "Список запчастин на складі:" << std::endl;  
 warehouse.printParts();  
  
 warehouse.removePartByName("Масляний фільтр");  
  
 warehouse.updatePart("Свічки запалювання", 220.0, 90);  
  
 std::cout << "\nОновлений список запчастин на складі:" << std::endl;  
 warehouse.printParts();  
  
 std::cout << "\nПошук запчастини 'Гальмівні колодки':" << std::endl;  
 warehouse.findPartByName("Гальмівні колодки");  
  
 warehouse.sortPartsByPrice();  
 std::cout << "\nСписок запчастин після сортування за ціною (за зростанням):" << std::endl;  
 warehouse.printParts();  
  
 return 0;  
}

*Лістинг part.cpp*

#include "part.h"  
  
#include <iostream>  
  
std::string Part::getName() const {  
 return name;  
}  
  
double Part::getPrice() const {  
 return price;  
}  
  
void Part::setPrice(double newPrice) {  
 price = newPrice;  
}  
  
void Part::setQuantity(int newQuantity) {  
 quantity = newQuantity;  
}  
  
void Part::print() const {  
 std::cout << "Назва: " << name << ", Виробник: " << manufacturer  
 << ", Ціна: " << price << ", Кількість: " << quantity << std::endl;  
}

*Лістинг part.h*

#ifndef PART\_H  
#define PART\_H  
  
#include <string>  
  
class Part {  
private:  
 std::string name;  
 std::string manufacturer;  
 double price;  
 int quantity;  
  
public:  
 Part(const std::string& name, const std::string& manufacturer, double price, int quantity)  
 : name(name), manufacturer(manufacturer), price(price), quantity(quantity) {}  
  
 std::string getName() const;  
  
 double getPrice() const;  
  
 void setPrice(double newPrice);  
  
 void setQuantity(int newQuantity);  
  
 void print() const;  
  
};  
  
#endif // PART\_H

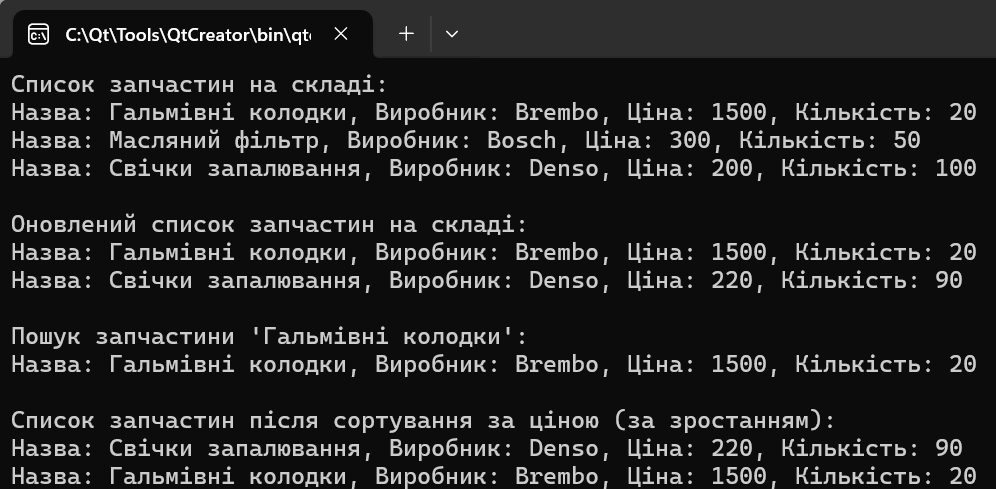
*Лістинг warehouse.cpp*

#include "warehouse.h"  
  
#include <algorithm>  
#include <iostream>  
  
void Warehouse::addPart(const Part &part) {  
 parts.push\_back(part);  
}  
  
void Warehouse::removePartByName(const std::string &name) {  
 parts.erase(std::remove\_if(parts.begin(), parts.end(), [&name](const Part& p) {  
 return p.getName() == name;  
 }), parts.end());  
}  
  
void Warehouse::updatePart(const std::string &name, double newPrice, int newQuantity) {  
 for (auto& part : parts) {  
 if (part.getName() == name) {  
 part.setPrice(newPrice);  
 part.setQuantity(newQuantity);  
 break;  
 }  
 }  
}  
  
void Warehouse::printParts() const {  
 for (const auto& part : parts) {  
 part.print();  
 }  
}  
  
void Warehouse::findPartByName(const std::string &name) const {  
 for (const auto& part : parts) {  
 if (part.getName() == name) {  
 part.print();  
 return;  
 }  
 }  
 std::cout << "Запчастину не знайдено." << std::endl;  
}  
  
void Warehouse::sortPartsByPrice() {  
 std::sort(parts.begin(), parts.end(), [](const Part& a, const Part& b) {  
 return a.getPrice() < b.getPrice();  
 });  
}

*Лістинг warehouse.h*

#ifndef WAREHOUSE\_H  
#define WAREHOUSE\_H  
  
#include <vector>  
  
#include "part.h"  
  
class Warehouse {  
private:  
 std::vector<Part> parts;  
  
public:  
 void addPart(const Part& part);  
  
 void removePartByName(const std::string& name);  
  
 void updatePart(const std::string& name, double newPrice, int newQuantity);  
  
 void printParts() const;  
  
 void findPartByName(const std::string& name) const;  
  
 void sortPartsByPrice();  
  
};  
  
#endif // WAREHOUSE\_H

*Результат виконання*



**Висновок:**

Виконавши цю лабораторну роботу, я здобув практичні навички роботи з контейнерними класами та стандартною бібліотекою шаблонів (STL) у C++. У рамках завдання я розробив програму для управління складом автозапчастин, яка дозволяє виконувати основні операції: додавання, видалення, оновлення даних про запчастини, сортування списку за ціною та пошук елементів.

Реалізація вимагала структурування коду, створення класів із відповідними методами та використання STL-контейнерів, таких як std::vector.

Результати виконання демонструють, що програма відповідає поставленим вимогам, працює коректно та забезпечує необхідний функціонал.