

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ  
імені ІГОРЯ СІКОРСЬКОГО»

**ДОМАШНЯ КОНТРОЛЬНА РОБОТА**

з дисципліни «Технології об'єктно-орієнтованого програмування для  
радіоінженерів»

Виконав:  
студент РТФ  
групи РС-11

Київ – 2023

**Зміст:** Розробка програмного забезпечення з використання об'єктно-орієнтованого програмування на мові C++. Закріпити отримані знання і здобути практичний досвід розробки програмного забезпечення з використанням технології об'єктно-орієнтованого програмування (ООП) шляхом написання програмного коду згідно поставленого завдання.

**Умова задачі:** Створити програмне забезпечення (ПЗ) з використанням ООП на мові C++ відповідно до варіанту задачі з можливістю логування важливих моментів роботи програми. Програма має містити мінімум два класи написаних власне студентом для реалізації поставленої задачі. Створити класи, специфікації яких наведені нижче. Визначити конструктори і методи set.. (), get.. (), toString (). Задати критерій вибору даних і вивести ці дані на консоль. У кожному класі, що володіє інформацією, має бути оголошено кілька конструкторів. Необхідно перевантажити оператор '«' для виводу даних на консоль. Має бути можливість записати дані всіх об'єктів у файл з шифруванням (студент самостійно обирає спосіб шифрування). Додатково має бути написаний клас для логування важливих моментів роботи програми з виводом у файл. Для виконання роботи був обраний варіант № 4 - Car

Car: id, Марка, Модель, Рік випуску, Колір, Ціна, Реєстраційний номер. Створити масив об'єктів. Вивести: а) список автомобілів заданої марки; б) список автомобілів заданої моделі, які експлуатуються більше n років; с) список автомобілів заданого року випуску, ціна яких більше зазначеної.

### **Опис розробленого програмного забезпечення:**

Розроблене програмне забезпечення є системою керування гаражем, реалізованою за допомогою об'єктно-орієнтованого підходу на мові програмування C++.

Основні складові програмного забезпечення:

Клас "Car": Представляє автомобіль з такими атрибутами як ідентифікатор, марка, модель, рік випуску, колір, ціна та реєстраційний номер. Клас має методи для отримання та встановлення значень атрибутів, а також метод toString() для перетворення об'єкту в рядок.

Клас "Garage": Представляє гараж, який містить колекцію автомобілів (об'єктів класу "Car"). Клас має методи для додавання автомобіля до гаражу, отримання списку автомобілів за певними критеріями (марка, модель, рік випуску та ціна), а також метод toString(), який перетворює гараж у рядок. Клас також має можливість зберігання всіх автомобілів у файлі з шифруванням XOR.

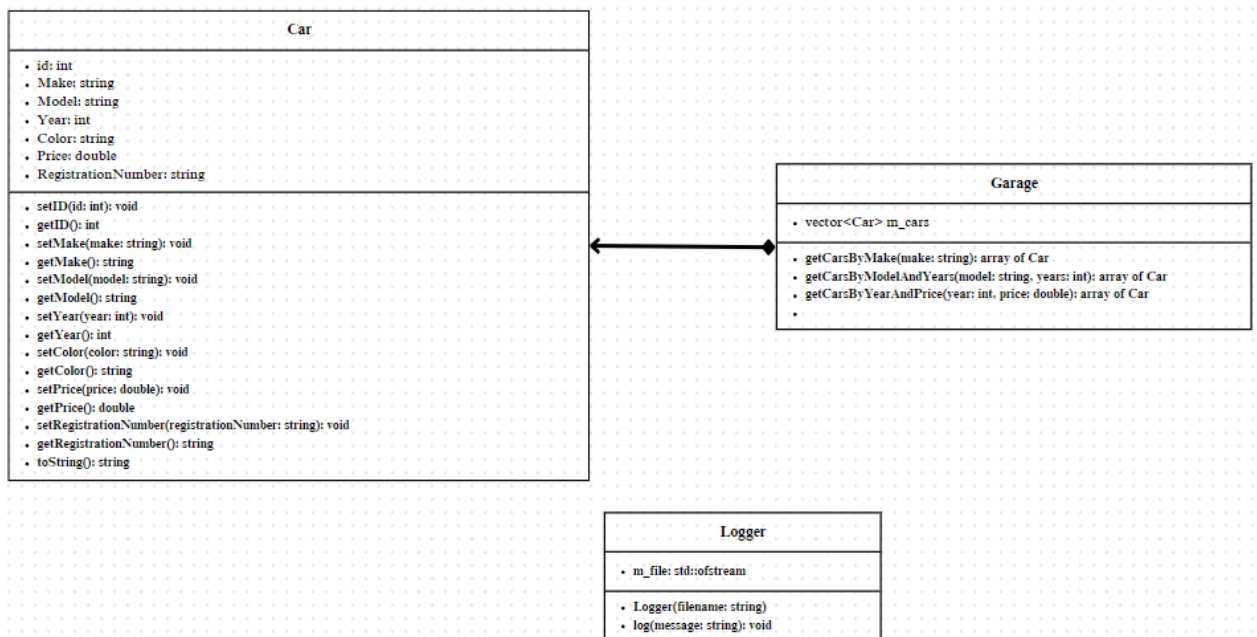
Клас "Logger": Представляє систему логування важливих моментів роботи програми. Клас приймає назву файлу, до якого будуть записуватися повідомлення, і має метод log(), який дозволяє логувати повідомлення у файл.

Програмне забезпечення дозволяє користувачу додавати автомобілі до гаражу, шукати автомобілі за різними критеріями (марка, модель, рік випуску, ціна) та зберігати список автомобілів у файл. Крім того, система логування дозволяє реєструвати важливі моменти роботи програми, які записуються у вказаний файл.

Всі класи мають визначені конструктори, методи set...() та get...() для доступу до атрибутів, а також метод toString(), який повертає рядкове представлення об'єкту. Клас "Garage" також перетворює свою колекцію автомобілів в рядок. Використовуючи оператор <<, дані можна виводити на консоль або записувати у файл.

Розроблене програмне забезпечення надає зручний інтерфейс для керування автомобілями в гаражі, забезпечуючи можливість пошуку за різними критеріями, збереження даних та логування важливих моментів роботи програми.

## UML diagram:



## Программный код:

```
#include "Garage.h"
#include "Logger.h"
#include <iostream>

int main() {
    Garage garage;

    // Creating cars
    Car car1(1, "Toyota", "Camry", 2019, "Black", 25000.0, "AB1234CD");
    Car car2(2, "Honda", "Civic", 2020, "Red", 22000.0, "EF5678GH");
    Car car3(3, "Toyota", "Corolla", 2015, "Silver", 18000.0, "IJ9101KL");

    // Adding cars to the garage
    garage.addCar(car1);
    garage.addCar(car2);
    garage.addCar(car3);

    // Logging important moments
    Logger logger("log.txt");
    logger.log("Cars added to the garage.");

    // Displaying all cars in the garage
    std::cout << "All cars in the garage:\n" << garage << std::endl;

    // Getting cars by make
    std::string make = "Toyota";
    std::cout << "Cars with make " << make << ":\n";
    std::vector<Car> carsByMake = garage.getCarsByMake(make);
    for (const auto& car : carsByMake) {
        std::cout << car << std::endl;
    }
}
```

```

}

// Getting cars by model and years
std::string model = "Camry";
int years = 3;
std::cout << "Cars with model " << model << " that have been in operation for more
than " << years << " years:\n";
std::vector<Car> carsByModelAndYears = garage.getCarsByModelAndYears(model,
years);
for (const auto& car : carsByModelAndYears) {
    std::cout << car << std::endl;
}

// Getting cars by year and price
int year = 2019;
double price = 20000.0;
std::cout << "Cars with year " << year << " and price higher than " << price <<
":\n";
std::vector<Car> carsByYearAndPrice = garage.getCarsByYearAndPrice(year, price);
for (const auto& car : carsByYearAndPrice) {
    std::cout << car << std::endl;
}

// Saving cars to a file
garage.saveToFile("cars.txt");
logger.log("Cars saved to file.");

return 0;
}

```

**Посилання на репозиторій GitHub:** \_\_\_\_\_

**Висновок:** під час виконання даного завдання було розроблено програмне забезпечення на мові програмування C++, яке використовує принципи об'єктно-орієнтованого програмування. Програма включає три класи.

Для виконання завдання було використано стандартну бібліотеку C++. Вона надає широкий набір засобів і функцій, що використовуються для розробки програмного забезпечення, включаючи роботу з файлами, вводом-виводом, рядками та іншими типами даних.

Однією з основних складнощів при виконанні цієї роботи було правильне розуміння та використання принципів об'єктно-орієнтованого програмування, включаючи визначення класів, створення об'єктів, використання методів та властивостей об'єктів.

Також виникли складнощі з шифруванням даних при записі в файл. Це вимагало додаткового вивчення принципів шифрування та використання відповідних засобів мови C++ для його реалізації.

Все взятє врахування, виконання цього завдання було важливим досвідом, який допоміг зрозуміти та практично застосувати принципи об'єктно-орієнтованого програмування, роботу з файлами, шифрування даних та інші важливі аспекти розробки програмного забезпечення на мові C++.

Також виникли складнощі з шифруванням даних при записі в файл. Це вимагало додаткового вивчення принципів шифрування та використання відповідних засобів мови C++ для його реалізації.

Все взятє врахування, виконання цього завдання було важливим досвідом, який допоміг зрозуміти та практично застосувати принципи об'єктно-орієнтованого програмування, роботу з файлами, шифрування даних та інші важливі аспекти розробки програмного забезпечення на мові C++.

### **Джерела**

1. Stroustrup, B. (2013). The C++ Programming Language (4th ed.). Addison-Wesley. - Основний підручник для вивчення мови C++, який був використаний для вивчення основних концепцій та синтаксису.
2. Cplusplus.com. (n.d.). C++ Reference. Retrieved from <http://www.cplusplus.com/reference/> - Основний онлайн ресурс для пошуку інформації про різні функції та класи стандартної бібліотеки C++.
3. Stack Overflow. (n.d.). Retrieved from <https://stackoverflow.com/> - Важливий ресурс для вирішення конкретних проблем під час розробки.