Міністерство освіти і науки України

Харківський національний університет радіоелектроніки

Кафедра Програмної інженерії

КУРСОВА РОБОТА

ПОЯСНЮВАЛЬНА ЗАПИСКА

з дисципліни «Об’єктно-орієнтоване програмування»

АВТОСАЛОН

Керівник Професор Бондарєв В. М.

Студент гр. ПЗПІ-19-4 Мартиненко А.О.

Комісія:

Старший викладач \_\_\_\_\_\_ Черепанова Ю.Ю.

Професор \_\_\_\_\_\_\_\_Бондарєв В.М.

Доцент \_\_\_\_\_\_\_ Побіженко І.О.

Харків 2020

**ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ РАДІОЕЛЕКТРОНІКИ**

Кафедра: ***Програмної інженерії***

Дисципліна: ***Об’єктно-орієнтоване програмування***

Спеціальність: ***121***  ***Інженерія програмного забезпечення***

Освітня програма: ***Програмна інженерія***

Курс 1 .Група *ПЗПІ-19 -4* . Семестр *2* .

***ЗАВДАННЯ***

***на курсовий проект студента***

***Мартиненко Андрій Олексійович***

(Прізвище, Ім'я, По батькові)

1 Тема проекту: *Автосалон*

2 Термін здачі студентом закінченого проекту: ***“19” - червня - 2020 р***.

3 Вихідні дані до проекту:

*Специфікація програми, методичні вказівки до виконання курсової роботи*

4 Зміст розрахунково-пояснювальної записки:

*Вступ,* *специфікація програми, проектна специфікація, інструкція користувача, висновки\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

5 Перелік графічного матеріалу:

*Діаграми класів та прецедентів,* *схема об’єктної моделі, алгоритми, приклади екранних форм\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**КАЛЕНДАРНИЙ ПЛАН**

|  |  |  |
| --- | --- | --- |
| *№* | *Назва етапу* | *Термін виконання* |
| 1 | Видача теми, узгодження і затвердження теми | 28-02-2020 р. |
| 2 | Формулювання вимог до програми | 28-02-2019 – 01-04-2020 р. |
| 3 | Розробка моделей | 01-04-2020 – 05-04-2020р. |
| 4 | Розробка проектної специфікації | 05-04-2020 – 12-04-2020 р. |
| 5 | Розроблення двох типів користувачів: адміністратор та клієнт | 12-04-2020 – 17-04-2020 р. |
| 6 | Розробка функцій зберігання та завантаження даних | 17-04-2020 – 20-04-2020 р. |
| 7 | Тестування і доопрацювання розробленої програмної системи. | 20-04-2020 – 20-05-2020 р. |
| 8 | Оформлення пояснювальної записки, додатків, графічного матеріалу | 20-05-2020 – 31-05-2020 р. |
| 9 | Захист | 01-06-2020 – 10-06-2020 р. |

Студент Мартиненко А.О.

Керівник Бондарєв В.М.

« 31 »\_травня\_\_\_\_\_\_\_\_\_\_ 2020 р.

**РЕФЕРАТ**

Пояснювальна записка до курсової роботи: 35 с., 25 рис., 7 додатків, 4 джерела.

БАЗА АВТОМОБІЛІВ, БАЗА КЛІЄНТІВ, АДМІНІСТРАТОР, ЗАМОВЛЕННЯ, АВТОМОБІЛЬ, КЛАС, МОВА ПРОГРАМУВАННЯ C#, ОБ’ЕКТНО-ОРІЄНТОВАНЕ ПРОГРАМУВАННЯ, ПЛАТФОРМА .NET, ПРОГРАМА.

Метою роботи є розробка програми «Автосалон» на засадах об'єктно-орієнтованого програмування для замовлення автомобілів в автосалоні.

Методи розробки базуються на використанні середи розробки Microsoft Visual Studio 2019, Windows Forms, платформи .NET Framework 4.7.2, мови програмування C#.

У результаті отримана програма під назвою «Автосалон», яка дозволяє звичайному користувачу переглядати список автомобілів для замовлення, підібрати авто за власними побажаннями, замовити автомобіль та відмовитися від замовлення. Авторизований користувач (клієнт) може переглядати список своїх замовлень, скористатися автоматизованим підбором авто.

**ЗМІСТ**

[ВСТУП 7](#_Toc11182230)

[1 СПЕЦИФІКАЦІЯ ПРОГРАМИ 8](#_Toc11182231)

[1.1 Діаграма прецедентів 8](#_Toc11182232)

[1.2 Бізнес-вимоги 8](#_Toc11182233)

[1.3 Вимоги користувачів 9](#_Toc11182234)

[1.4 Функціональні вимоги 9](#_Toc11182235)

[1.5 Нефункціональні вимоги 10](#_Toc11182236)

[1.6 Опис інтерфейсу користувача 11](#_Toc11182237)

[2 ПРОЕКТНА СПЕЦИФІКАЦІЯ 19](#_Toc11182238)

[2.1 Діаграма класів 19](#_Toc11182239)

[2.2 Об’єктна структура програми 21](#_Toc11182240)

[2.2.1 Абстрактний клас User 20](#_Toc11182241)

[2.2.2 Клас Admin 20](#_Toc11182242)

[2.2.3 Клас Client 21](#_Toc11182243)

[2.2.4 Клас Order 21](#_Toc11182244)

[2.2.5 Клас Car 21](#_Toc11182245)

[2.2.6 Клас DAO 21](#_Toc11182246)

[2.2.7 Основний клас Showroom 21](#_Toc11182247)

[3 ІНСТРУКЦІЯ КОРИСТУВАЧА 22](#_Toc11182249)

[3.1 Встановлення, запуск та вхід до програми 22](#_Toc11182250)

[3.2 Рєестрація у програмі 24](#_Toc11182251)

[3.3 Користування адмінпанеллю 25](#_Toc11182252)

[3.4 Користування програмою в ролі клієнта 28](#_Toc11182253)

[ДОДАТОК А Абстрактний клас User 34](#_Toc11182254)

[ДОДАТОК Б Клас Admin 34](#_Toc11182255)

[ДОДАТОК В Клас Сlient 34](#_Toc11182256)

[ДОДАТОК Г Клас Order 35](#_Toc11182257)

[ДОДАТОК Д Клас Car 35](#_Toc11182258)

[ДОДАТОК E Клас DAO 36](#_Toc11182258)

[ДОДАТОК Ж Клас Showroom 38](#_Toc11182258)

ВСТУП

Придбання авто це відповідальний вчинок, тому автосалон надає користувачу можливість за власними побажаннями підібрати авто до смаку. Клієнт обирає авто за маркою, технічним станом, технічними характеристиками та ціною. Ввівши основні характеристики та скориставшись авто-підбором клієнт отримає найбіль вигідний та пасуючий автомобіль, щоб дізнатися про авто більше також є параметр «опис особливостей виконання».

Розроблена програма дозволить користувачу виконувати усі згадані вище дії. Користувач реєструється в базі, формується його список замовлень у адміністратора австосалону. Клієнт має можливість корегувати список своїх замовлень.

Окрім розробки вище описаної програмної системи метою даної курсової роботи є освоїти мову програмування С#, навчитися об’єктно-орієнтованому програмуванню, розвити навички роботи в Windows Forms.

Задачі роботи:

* Розробка системи зчитування та запису даних;
* Створення логіки роботи системи;
* Робота з користувачем;
* Розробка логіки додавання, пошуку та видалення інформації;
* Тестування готового продукту;

1. СПЕЦИФІКАЦІЯ ПРОГРАМИ
   1. Діаграма прецедентів

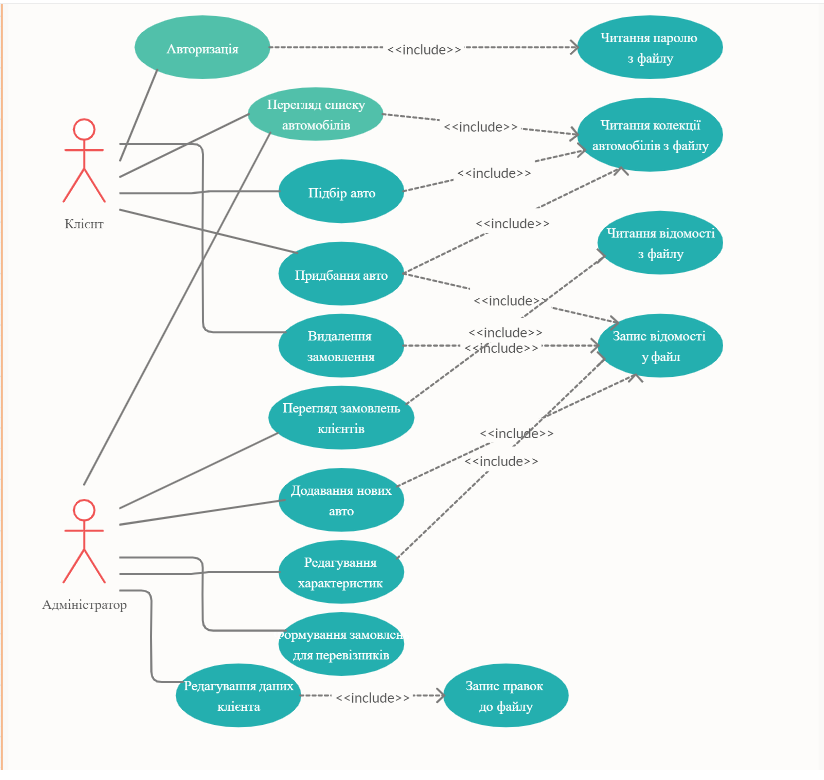


Рисунок 1.1 – Діаграма прецедентів

* 1. Бізнес-вимоги

Серед бізнес-вимог слід зазначити наступні:

* Продаж та відмова клієнта від замовлення;
* Формування списку замовлень для постачальників від кожного клієнта;
* Відображення списку автомобілів.
  1. Вимоги користувачів

Серед вимог користувачів слід зазначити наступні:

* Перегляд списку автомобілів;
* Підбір авто за вказаними параметрами;
* Замовлення автомобіля;
* Відмова від замовлення;
* Авторизація користувача (введення паролю);
* Перегляд замовлень клієнтів (адміністратор);
* Додавання нового авто (адміністратор);
* Редагування характеристик авто (адміністратор);
  1. Функціональні вимоги

Серед функціональних вимог слід зазначити наступні:

* Система має зчитувати з файла список автомобілів та зберігати зміни;
* Система має зчитувати з файла список авто для перевізників та записувати туди зміни за необхідності;
* Система має зчитувати з файла список замовлень кожного клієнта та записувати туди зміни у ньому за необхідності;
* Система має звіряти введений пароль з паролем, зчитаним та розшифрованим з файлу;
* Система має шифрувати та зберігати змінений пароль у файл;
* Система має порівнювати введені характеристики клієентом з існуючими в списку, наявними в автосалоні та знаходити усі авто з ним;
* Система має зберігати дані про автомобіль у файл;
* Система має считувати інформацію. Про клієнта та надавати список його замовлень.
  1. Нефункціональні вимоги

Серед нефункціональних вимог слід зазначити наступні:

* Легкість та простота використання;
* Легкість переміщення;
* Цілісність;
* Ефективність та стійкість до збоїв;
* Взаємодія системи і зовнішнього світу;
* Обмеження дизайну і реалізації.

Серед обмежень дизайну і реалізації слід зазначити наступні:

* Використання платформи .NET Framework 4.7.2;
* Використання мови програмування C#;
* Використання Windows Forms.
  1. Опис інтерфейсу користувача

Перед входом до програми користувач має увійти або зареєструватися «Авторизація», він побачить вікно введення паролю (рис. 1.2).

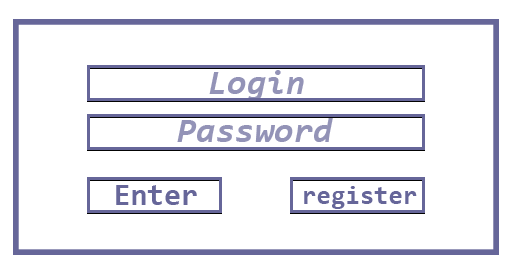


Рисунок 1.2 – Вікно авторизації

У цьому вікні є такі елементи:

* Поле введення паролю адміністратора та клієнта;
* Кнопка «Enter»;
* Кнопка «register»;

Якщо пароль, введений у поле, вірний, після натиснення кнопки «Enter» це вікно закривається, а головне вікно відкривається в залежності від користувача (адміністратор або клієнт) (рис. 1.4)

Якщо користувача немає в системі, йому потрібно зареєструватися, відкривається форма на (рис. 1.3)

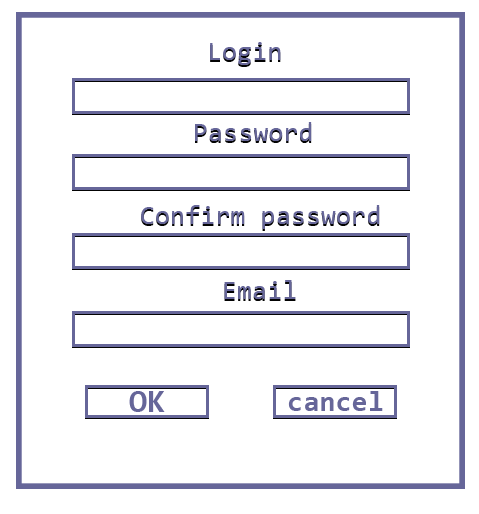


Рисунок 1.3 – Вікно реєстрації

У цьому вікні є такі елементи:

* Поле введення паролю адміністратора та клієнта;
* Поля для введення «Login», «Password», «Confirm password», «Email»;
* Кнопка «OK»;
* Кнопка «Cancel»;

Після запуску програми відкривається її головне вікно (рис. 1.4), на якому відображається список автомобілів доступних для замовлення та список замовлених авто на другій сторінці, на головній сторінці є можливість підбору авто за вказаними характеристиками, є можливість відмовитися від замовлення.

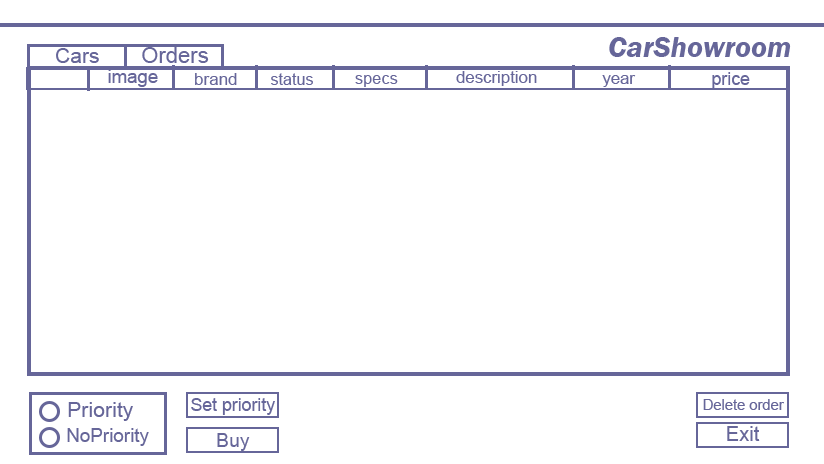


Рисунок 1.4 – Головне вікно (клієнт)

У цьому вікні є такі елементи:

* Таблиця автомобілів;
* Вибір приорітету автопидбору;
* Кнопка «Set priority»;
* Кнопка «Buy»;
* Кнопка «Delete order»;
* Кнопка «Exit».

Кнопка «Exit» закриває програму.



Рисунок 1.5 – Головне вікно (адміністратор)

На вікні містяться елементи:

* Таблиці «Vehicles» - список авто та «Clients» - список формувать заявок для перевізників;
* «File» – «Load», «Save», «Exit»;
* «Car» – «New», «Edit», «Delete»;
* «Client» – «New», «Edit», «Delete»;
* «Help» – «Info»;
* «Exit»;

Після тире наведено пункти підменю.

Слід зазначити, що авторизований користувач (адмінітратор) може редагувати таблицю «Vehicles».

Після натиснення у меню «Clients» відкривається вікно з інформацією для перевізників(рис. 1.6).

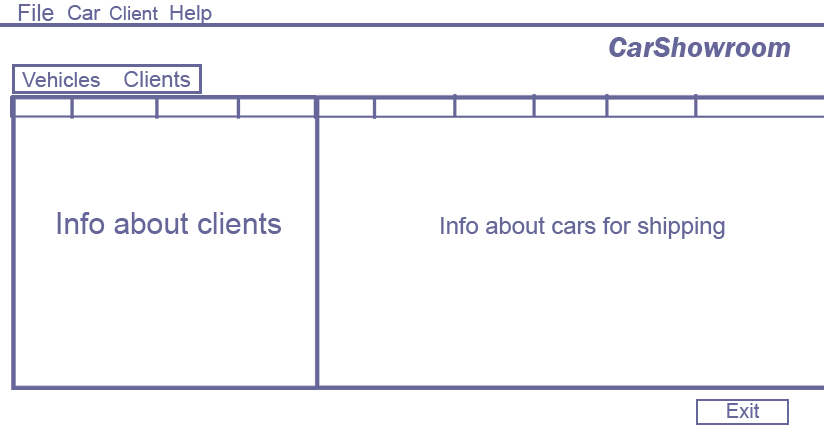


Рисунок 1.6 – Вікно формувань замовлень для постачальників

Після натиснення у меню «File» відкривається меню «File» «Load», «Save», «Exit»;

(рис. 1.7).

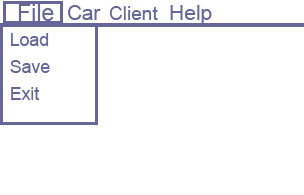


Рисунок 1.7 – меню «File»

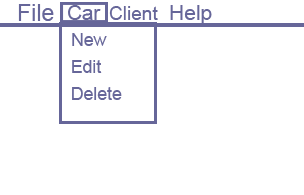


Рисунок 1.8 – Меню «Car»

У цьому вікні є такі елементи:

* «New» - додати новий автомобіль до списку;
* «Edit» - зміна автомобіля у списку;
* «Delete» - додати новий автомобіль до списку;

Після натиснення у меню «New» відкривається вікно для заповнення характеристик авто(рис. 1.9).

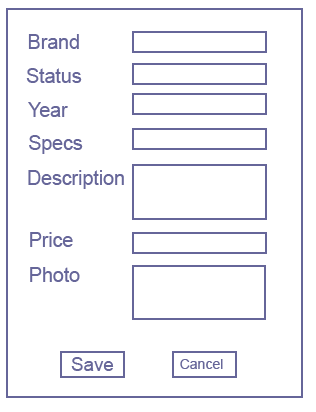


Рисунок 1.9 – Вікно додавання автомобіля

На ньому є такі елементи:

* Марка;
* Технічний стан;
* Рік випуску;
* Технічні характеристики;
* Запитувана ціна;
* Зображення;

Кнопка «Cancel» закриває це вікно.

При натисканні «New» відкривається ця форма для редагування.

При натисканні «Delete» авто видаляється зі списку.

Адміністратор має можливість зареєструвати клієнта в системі. Для цього потрібно обрати пункт «Client» - «New», відкриється форма реєстрації (рис.1.3).

Для автоматизації підбору авто, клієнту потрібно натиснути кнопку «Set priority» (рис.1.4). Відкриється вікно підбору(рис.1.10)



Рисунок 1.10 – Вікно автопідбору

На ньому є такі елементи:

* Поле введення марки;
* Поле введення технічного стану;
* Поле введення характеристик;
* Поле введення ціни;
* Кнопка «Accept»;
* Кнопка «Cancel»;

Кнопка «Назад» закриває це вікно та відкриває згорнуте раніше вікно зі списком автомобілів (рис. 1.4).

1. ПРОЕКТНА СПЕЦИФІКАЦІЯ
   1. Діаграма класів

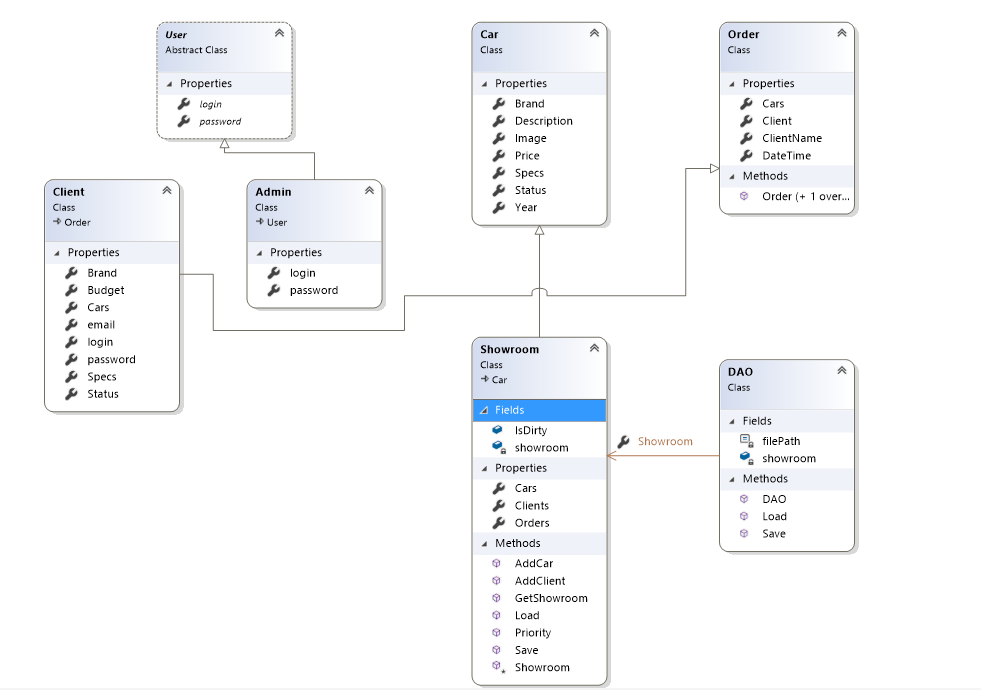
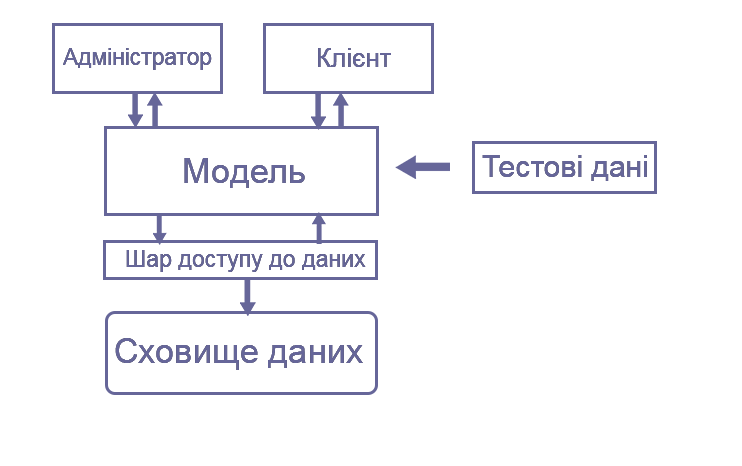


Рисунок 2.1 – Діаграма класів

* 1. Об’єктна структура програми

Програма складається з 7 класів і 6 форм. Усі форми були розглянуті у попередньому розділі. На рисунку 2.2 наведено архітектуру проекту.



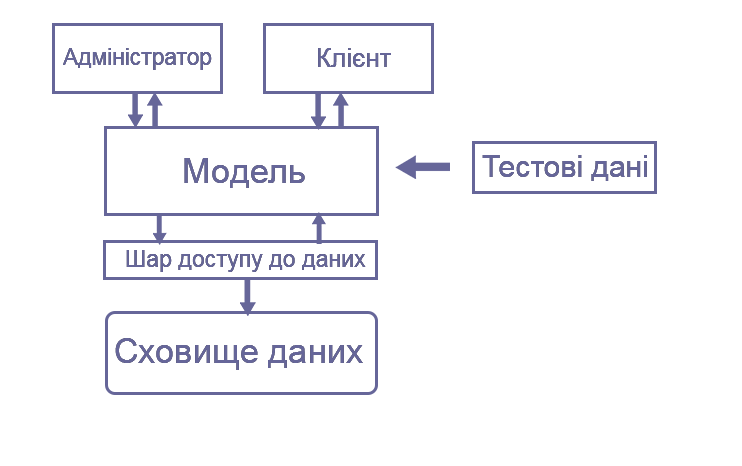


Рисунок 2.2 – Архітектура проекту

2.2.1 Абстрактний клас User

Абстрактний лас User описує логін та пароль користувача. Він має дві відкриті властивості: «login» та «password».

2.2.2 Клас Admin

Клас Admin успадковується від класу User, тому він має перегружені властивості «login» та «password», які в адміністратора записані admin та admin:

2.2.3 Клас Сlient

Клас Client успадковується від абстрактного класу User, має дві перегружені властивості: «login» та «password», «email». Властивості: «Brand», «Specs», «Status», «Budget» і колекцію «Car» автомобілів, що описує вимоги клієнта до марки, технічного стану, технічних характеристик та запитуваної ціни.

2.2.4 Клас Order

Клас Order описує замовлення, у нього з’являються такі властивості: колекція авто та колекція покупців.

2.2.5 Клас Car

Клас Car описує автомобіль за допомогою наступних властивостей: технічний стан авто, марка авто, рік випуску авто, технічні характеристики авто, запитувана ціна авто та зображення авто.

2.2.6 Клас DAO

Допоміжний клас, який допомагає зберігати та завантажувати дані програми.

Цей клас має декілька методів: «Save», «Load», «Copy».

2.2.7 Основний клас Showroom

Клас Showroom успадковується від класу Car.

Клас має такі методи:

AddCar – додовання нового авто;

AddClient – реєстрація нового клієнта;

Priority – вибір приорітету авто підбору;

Булевий метод IsDirty – перевірка на змінення даних;

1. ІНСТРУКЦІЯ КОРИСТУВАЧА
   1. Встановлення, запуск, вхід до програми та реєстрація

Для користування програмою її необхідно завантажити на ваш комп’ютер. Після цього необхідного відкрити файл за шляхом \CarShowroom\CarShowroom\bin\Debug.

Після запуску програми відкривається вікно входу (рис. 1.2).

Потрібно увійти до програми як адміністратор, або клієнт, щоб увійти в ролі адміністратора потрібно в поле «login» ввести admin та в поле «password» ввести admin. Щоб увійти в ролі клієнта потрібно зареєструватися, наступний пункт.

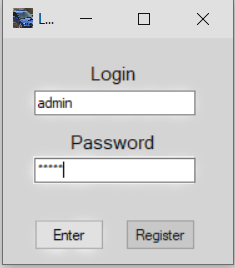


Рисунок 1.2 – Авторизація в програму

* 1. Реєстрація у програмі:

Для реєстрації потрібно натиснути кнопку Register на формі входу(рис.1.2)

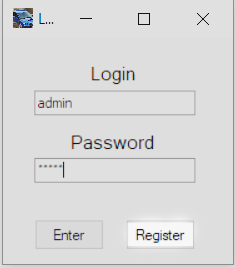


Рисунок 1.2 – Вхід до програми

Відкриється нова форма, потрібно заповнити поля «login», «password», «confirm password», «email» та натиснути «Accept» (рис.1.3)

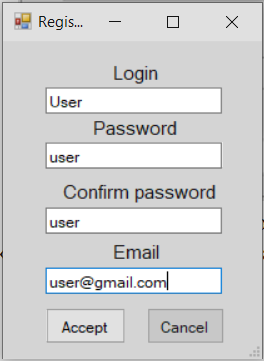


Рисунок 1.3 – Реєстрація у програмі

Далі відкриється форма з (рис.1.2) для виконання входу.

* 1. Користування адмінпанеллю. Вхід у адмінпанель

Після виконання входу в ролі адміністратора одразу відкриється форма адмінпанелі (рис.1.4).

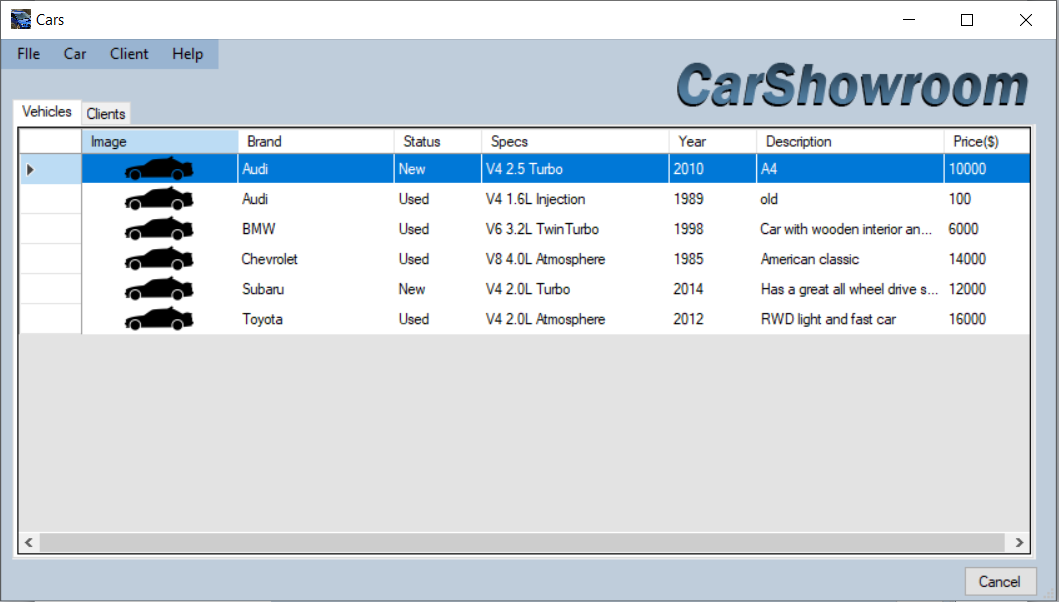


Рисунок 1.4 - Адмінпанель

1) Додавання нового авто:

Щоб додати нове авто потрібно обрати меню «Car» - «New», відкриється форма внесення даних(рис.1.5).

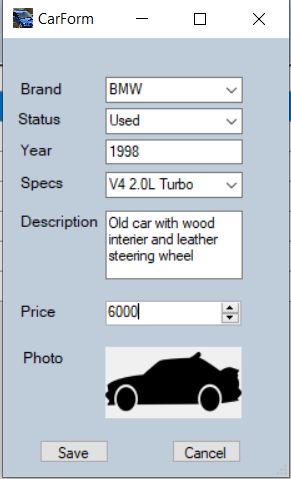


Рисунок 1.5 – Додавання нового авто

Усі поля потребують вводу даних типу «String» окрім поля «Price».

2) Редагування авто:

Щоб відредагувати авто потрібно натиснути меню «Car» - «Edit»

та відкриється форма як на рис.1.5 для зміни даних.

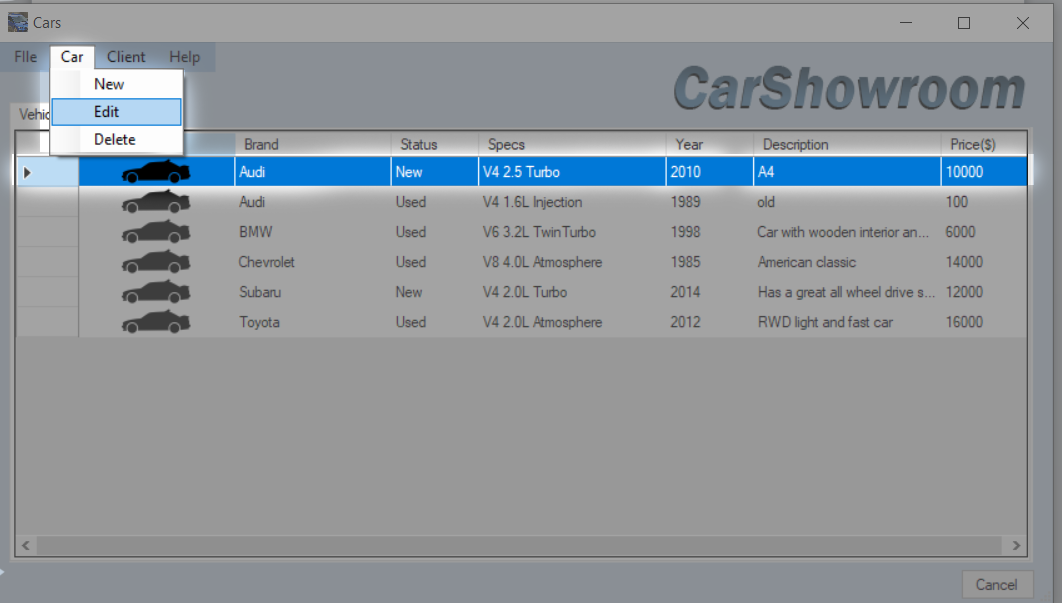


Рисунок 1.6 – Редагування авто

3) Видалення авто:

Щоб видалити автомобіль зі списку потрібно натиснути меню «Car» - «Delete». Після натискання авто більше не буде перебувати у списку.

4) Керування клієнтською базою

На формі адмінпанелі є друга вкладка «Clients»:

Адміністратор може реєструвати клієнтів у офлайн режимі, також іх видаляти за допомогою «Clients»-«New» та «Clients»-«Delete» (рис.1.7).

Також у вкладці «Clients» відображається база покупців та їх список замовлень для перевізників.

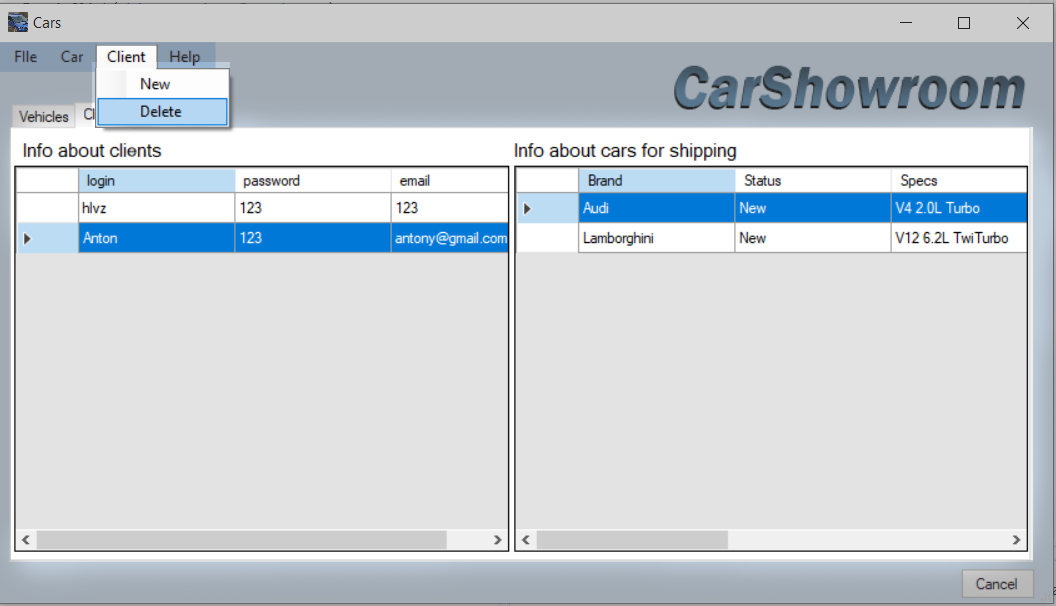


Рисунок 1.7 – Керування клієнтською базою

* 1. Користування програмою в ролі клієнта:

Після введеня своїх зареєстрованих даних в форму входу (рис.1.2), користувачу відкривається форма списку авто та параметрів підбору.



Рисунок 1.8 – Панель користувача

1) Користування авто підбором:

Для того щоб підібрати за вподобанними характеристиками з переліку авто, потрібно натискнути кнопку «Set priority». Відкриється форма для заповнення параметрів підбору авто(рис.1.9).

2) Форма авто підбору

Форма має п’ять основних параметрів пошуку «Brand»-марка, «Status»-технічний стан, «Specs»-технічні характеристики та «Budget»(ввести потрібно число більше або рівне тому, скільки кштує авто). Потрібно обрати зі списку параметри які вас цікавлять та написати фінансові можливості, натиснути «Accept».

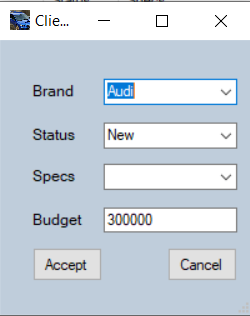


Рисунок 1.10 – форма підбору

Далі знову стане доступна форма (рис.1.8), потрібно натиснути RadioButton «Priority», якщо в списку є автомобіль котрий задовольняє параметрам, він залишиться у списку, а інші зникнуть(рис 1.8).

3) Купівля авто:

Виконавши попередню дію, ми підібрали один автомобіль. Щоб його придбати потрібно натиснути «Buy»

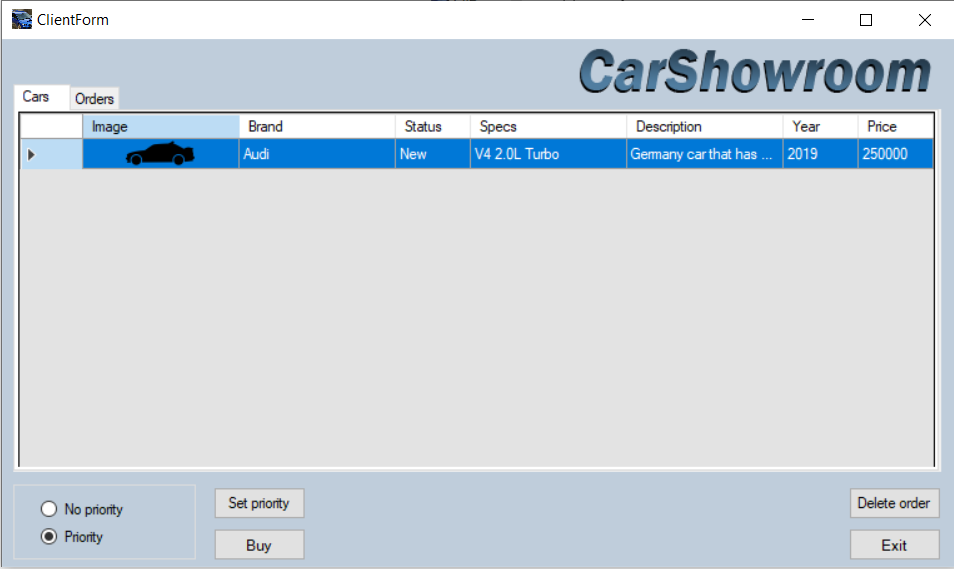


Рисунок 1.8 – Результат підбору

Натиснувши «Buy», ми побачимо повідомлення підтвердження купівлі(рис.1.9).

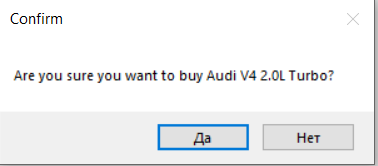


Рисунок 1.9 – підтвердження купівлі

4) Керування списком замовлень:

Після підтвердження, переходимо до вкладки «Orders». Та бачимо що ми дійсно зробили замовлення(рис.1.10)



Рисунок 1.10 – Замовлення

5) Відміна свого замовлення:

Щоб відмінити замовлення достатньо натиснути кнопку «Delete order». Нам запропонують погодитись(рис 1.11).

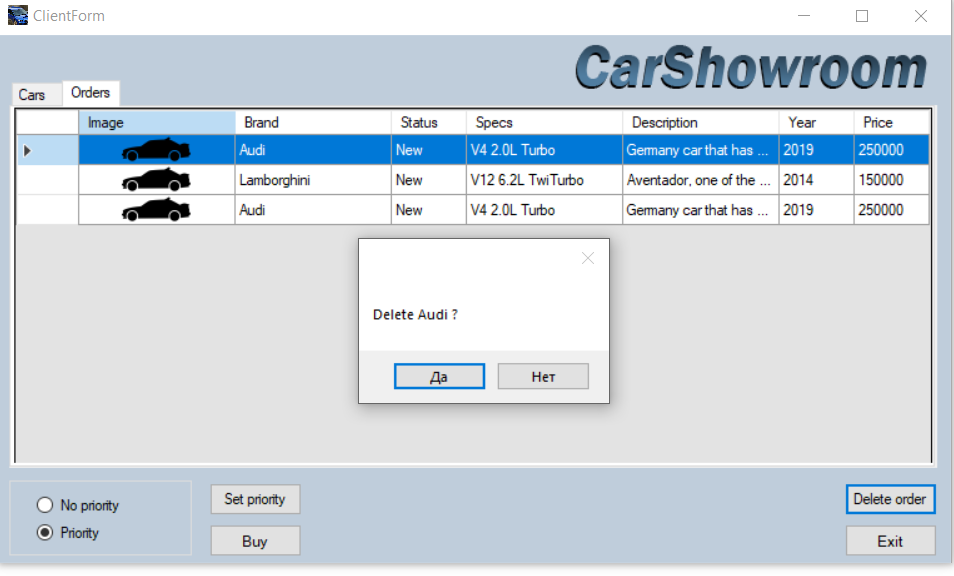


Рисунок 1.11 – Відміна замовлення

**ВИСНОВКИ**

У ході виконання курсової роботи були освоєні методи збереження даних до файлу, робота з елементом dataGridView та заповненням даних, було детально розглянуто створення програми на Windows Forms Application.

У результаті отримана програма під назвою «Автосалон», яка дозволяє звичайному користувачу передивлятися список автомобілів та за допомогою функції підбору, знаходити потрібну для того щоб придбати, також можлива відміна замовлення. Декілька інакший функціонал має адміністратор автосалону, він в змозі оновлювати ассортимент автосалону, редагувати будь-які моделі, видаляти їх та переглядати список клієнтів, які формують замовляння для перевізників.

Програма написана на мові програмування C# на платформі .NET Framework 4.7.2 з використанням технологій Windows Forms у середі розробки Microsoft Visual Studio 2019.

Усі дані програми зберігаються локально у файлах за допомогою серіалізації.

Галузь застосування – офлайн автосалон, ведення бази автомобілів та клієнтів автосалону.

**ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ**

1. Microsoft Developer Network. Библиотека MSDN / MSDN – сеть разработчиков Microsoft [Електронний ресурс] – Режим доступу до ресурсу: https://msdn.microsoft.com/ru-ru/.
2. Бондарев В. М. Объектно-ориентированное программирование на С# [Текст] : учеб. Пособие / В. М. Бондарев. – Харьков : СМИТ, 2009. –224 с. – Библиогр. : ISBN 978-966-2028-34-8.
3. Троелсен, Э. Язык программирования C# 5.0 и платформа .NET 4.5, 6-е изд. [Текст]: пер. с англ./ Э. Троелсен.- М.: Вильямс, 2013 - 1312 с. – ISBN 978-5-8459-1814-7, 978-1-43-024233-8.
4. Metanit.com. [Електронний ресурс] – Режим доступу до ресурсу: <https://metanit.com/sharp/>.

ДОДАТОК А  
Абстрактний клас User

public abstract class User

{

public abstract string login { set; get; }

public abstract string password { set; get; }

}

ДОДАТОК Б  
Клас Admin

class Admin : User

{

//логін

public override string login { get; set; } ="admin";

//пароль

public override string password { get; set; } = "admin";

}

ДОДАТОК В  
Клас Сlient

public class Client : User

{

//Вимоги до марки

public string Brand { set; get; }

//Вимоги до тех. характеристик

public string Specs { set; get; }

//Вимоги до тех. стану

public string Status { set; get; }

//Вимоги до запитуваної ціни

public int Budget { set; get; }

public override string login { get; set; }

public override string password { get; set; }

public string email { set; get; }

//Колекція запитуваних авто

public List<Car> Cars { private set; get; } = new List<Car>();

}

ДОДАТОК Г  
Клас Order

public class Order

{

//Колекція придбаних авто

public List<Car> Cars { private set; get; }

//Колекція покупців

public Client Client { private set; get; }

//Дані

public string ClientName { get => Client.login;

}

ДОДАТОК Д  
Клас Car

public class Car

{

//Опис особливостей виконання

public string Description { set; get; }

//Технічний стан авто

public string Status { set; get; }

//Марка авто

public string Brand { set; get; }

//Рік випуску авто

public string Year { set; get; }

//Технічні характеристики авто

public string Specs { get; set; }

//Запитувана ціна авто

public decimal Price { get; set; }

//Зображення авто

public Image Image { get; set; }

}

ДОДАТОК Е  
Клас DAO

public class DAO

{

Showroom showroom;

const string filePath = "showroom.bin";

public DAO(Showroom showroom)

{

this.showroom = showroom;

}

public Showroom Showroom

{

get => default;

set

{

}

}

//Збереження даних

public void Save()

{

using(Stream stream = File.Create(filePath))

{

var serializer = new BinaryFormatter();

serializer.Serialize(stream,showroom);

}

}

//Завантаження даних

public void Load()

{

using (Stream stream = File.OpenRead(filePath))

{

var serializer = new BinaryFormatter();

Showroom st = (Showroom)serializer.Deserialize(stream);

Copy(st.Clients, showroom.Clients);

Copy(st.Cars, showroom.Cars);

Copy(st.Orders, showroom.Orders);

}

//Перенесення списку

void Copy<T>(List<T> from, List<T> to)

{

to.Clear();

to.AddRange(from);

}

}

}

ДОДАТОК Ж  
Клас Showroom

public class Showroom : Car

{

public List<Client> Clients { private set; get; }

public List<Car> Cars { private set; get; }

public List<Order> Orders { private set; get; }

private static Showroom showroom;

// Перевірка зміни даних.

public bool IsDirty;

static public Showroom GetShowroom()

{

if (showroom == null)

{

showroom = new Showroom();

}

return showroom;

}

protected Showroom()

{

Clients = new List<Client>();

Cars = new List<Car>();

Orders = new List<Order>();

}

//Вибір приорітету (автопідбору)

public List<Car> Priority(Client cl)

{

List<Car> res = new List<Car>();

if (string.IsNullOrWhiteSpace(cl.Specs))

{

res = Cars.Where(o => o.Brand == cl.Brand && o.Price <= cl.Budget

&& o.Status == cl.Status).ToList();

}

else

{

res = Cars.Where(o => o.Brand == cl.Brand && o.Price <= cl.Budget

&& o.Specs == cl.Specs && o.Status == cl.Status).ToList();

}

return res;

}

//Додавання нового авто

public void AddCar(Car car)

{

Cars.Add(car);

}

//Реєстрація нового клієнта

public void AddClient(Client client)

{

Clients.Add(client);

}

//Збереження даних

public void Save()

{

new DAO(this).Save();

}

//Завантаження данних з файлу

public void Load()

{

new DAO(this).Load();

}

}

Репозиторій: <https://github.com/awdrallyracer/Kursovaia>