

Лабораторна №3

РОЗРОБКА ДІАГРАМ КЛАСІВ СИСТЕМИ

Мета: набуття практичних навичок щодо побудови та використання діаграм класів системи.

Хід роботи

«Онлайн-сервіс для підбору автозапчастин»

Завдання.

Побудувати діаграму класів системи. Для цього:

1. Визначити основні класи
2. Ідентифікувати атрибути класів
3. Визначити методи класів
4. Реалізувати зв'язки між класами.

1) Основні класи та їхня відповідальність

Система складається з 8 основних класів. Кожен клас відповідає за певну частину функціональності:

- **User (абстрактний клас):** Базовий клас для всіх користувачів системи. Відповідає за загальні операції, такі як реєстрація, авторизація та керування профілем. Він є шаблоном для похідних класів і не створюється безпосередньо.

Атрибути	Тип	Опис
id	int	Унікальний ідентифікатор користувача.
email	string	Електронна пошта для авторизації.
password	string	Пароль (зберігається хешованим для безпеки).
role	string	Роль користувача (наприклад, "buyer", "supplier", "admin").

					ЖИТОМИРСЬКА ПОЛІТЕХНІКА.25.121.29.000 – Лр.3						
Змн.	Арк.	№ докум.	Підпис	Дата	Звіт з лабораторної роботи №3			Літ.	Арк.	Аркушів	
Розроб.		Трофімчук М.О.								1	
Перевір.		Левківський В.Л.						ФІКТ, ар. ІПЗ-22-2			
Реценз.											
Н. Контр.											
Зав.каф.											

Методи	Повертає	Опис
register()	void	Реєстрація нового користувача в системі.
login()	bool	Авторизація; повертає true, якщо успішно.
editProfile()	void	Редагування профілю (email, password тощо).
deleteAccount()	void	Видалення облікового запису.

- **Buyer (похідний від User):** Представляє покупця (кінцевого користувача).
Відповідає за взаємодію з транспортними засобами, пошуком запчастин, замовленнями та відгуками.

Атрибути	Тип	Опис
vehicles	List<Vehicle>	Список транспортних засобів, пов'язаних з покупцем.
orders	List<Order>	Список замовлень покупця.
reviews	List<Review>	Список відгуків, написаних покупцем.
Методи	Повертає	Опис
addVehicle(vehicle: Vehicle)	void	Додавання транспортного засобу до профілю.
searchParts(query: string)	List<Part>	Пошук запчастин за запитом (VIN, номер, опис).
comparePrices(parts: List<Part>)	void	Порівняння цін від різних постачальників.
createOrder(parts: List<Part>, delivery: string)	Order	Створення замовлення з опцією доставки.
addToWishlist(part: Part)	void	Додавання запчастини до списку бажаних.
writeReview(part: Part, rating: int, comment: string)	Review	Написання відгуку про запчастину.

- **Supplier (похідний від User):** Представляє постачальника запчастин. Відповідає за керування інвентарем, обробку замовлень та комунікацію з покупцями.

Атрибути	Тип	Опис
inventory	Inventory	Інвентар постачальника.
Методи	Повертає	Опис
manageInventory()	void	Керування інвентарем (загальний метод).
addPart(part: Part)	void	Додавання нової запчастини.
editPart(part: Part)	void	Редагування запчастини.
deletePart(part: Part)	void	Видалення запчастини.
processOrder(order: Order)	void	Обробка замовлення (зміна статусу).
contactBuyer(buyer: Buyer, message: string)	void	Зв'язок з покупцем (чат/email).
viewOrders()	List<Order>	Перегляд списку замовлень.

- **Admin (похідний від User):** Представляє адміністратора. Відповідає за модерацію, керування користувачами та звіти.

Атрибути	Тип	Опис
(немає додаткових)	-	Наслідує від User.
Методи	Повертає	Опис
moderateContent()	void	Модерація контенту (відгуки, запчастини).
manageUsers()	void	Керування користувачами.
generateReports()	void	Генерація звітів.
blockUser(user: User)	void	Блокування користувача.

- **Vehicle:** Представляє транспортний засіб. Відповідає за зберігання даних про авто для підбору сумісних запчастин.

Атрибути	Тип	Опис
id	int	Унікальний ідентифікатор.
make	string	Марка (наприклад, "Toyota").
model	string	Модель (наприклад, "Camry").
year	int	Рік випуску.
vin	string	VIN-код для точного підбору.
Методи	Повертає	Опис
editDetails()	void	Редагування деталей.
delete()	void	Видалення транспортного засобу.

- **Part:** Представляє запчастину. Відповідає за опис, ціну та сумісність.

Атрибути	Тип	Опис
id	int	Унікальний ідентифікатор.
name	string	Назва запчастини.
description	string	Опис.
price	float	Ціна.
compatibility	List<string>	Список сумісних моделей.
supplier	Supplier	Постачальник.
reviews	List<Review>	Список відгуків.
Методи	Повертає	Опис
updatePrice(newPrice: float)	void	Оновлення ціни.
addCompatibility(model: string)	void	Додавання сумісності.

- **Order:** Представляє замовлення. Відповідає за процес замовлення, статус та розрахунок.

Атрибути	Тип	Опис
id	int	Унікальний ідентифікатор.
buyer	Buyer	Покупець.
supplier	Supplier	Постачальник.
parts	List<Part>	Список запчастин.
status	string	Статус (наприклад, "pending", "shipped").
deliveryOption	string	Опція доставки.
totalPrice	float	Загальна ціна.
Методи	Повертає	Опис
updateStatus(newStatus: string)	void	Оновлення статусу.
calculateTotal()	float	Розрахунок загальної ціни.
cancel()	void	Скасування замовлення.

- **Review:** Представляє відгук. Відповідає за оцінки та коментарі про запчастини.

Атрибути	Тип	Опис
id	int	Унікальний ідентифікатор.
buyer	Buyer	Покупець.
part	Part	Запчастина.
rating	int	Оцінка (1-5).
comment	string	Коментар.
Методи	Повертає	Опис
edit()	void	Редагування відгуку.
delete()	void	Видалення відгуку.

- **Inventory:** Представляє інвентар постачальника. Відповідає за керування запасами запчастин.

Атрибути	Тип	Опис
supplier	Supplier	Постачальник.
parts	List<Part>	Список запчастин.
Методи	Повертає	Опис
addPart(part: Part)	void	Додавання запчастини.
removePart(part: Part)	void	Видалення запчастини.
updateStock(part: Part, quantity: int)	void	Оновлення запасів.
viewStock()	List<Part>	Перегляд запасів.

2. Зв'язки між класами

Зв'язки показують, як класи взаємодіють. Вони включають наслідування (стрілка з порожнім трикутником), агрегації та асоціації.

- **Наслідування**

Buyer, **Supplier** та **Admin** успадковують від базового класу **User**. Це означає, що всі ці класи повторюють базові можливості користувача (реєстрація, логін, редагування профілю), але розширюють їх власним функціоналом.

- **Агрегація / Композиція**

Buyer → Vehicle

Клас *Buyer* містить список транспортних засобів, які належать користувачу. Існування *Vehicle* пов'язане з покупцем, але кожен автомобіль є окремим об'єктом системи.

Buyer → Order

Покупець створює замовлення. *Order* пов'язаний із покупцем та залежить від нього логічно, але залишається самостійним об'єктом системи.

Buyer → Review

Покупець залишає відгуки на деталі. Відгуки логічно належать покупцеві.

Supplier → Inventory

Постачальник має інвентар (склад). Це композиція: *Inventory* існує лише разом із постачальником.

Inventory → Part

Інвентар містить перелік деталей. Це агрегація: *Part* може існувати окремо або бути доданою кількома постачальниками.

Order → Part

Кожне замовлення містить одну або кілька деталей. Це асоціативний зв'язок «замовлення включає деталі».

○ Асоціації

Part → Supplier

Кожна деталь має постачальника, який її додає, редагує й продає.

Part → Review

Деталь може мати кілька відгуків від покупців.

Review → Buyer

У кожного відгуку є автор — покупець.

Review → Part

Відгук стосується конкретної деталі.

Order → Buyer / Supplier

Замовлення пов'язує покупця та постачальника в рамках торгової операції.

○ Типи зв'язків та їх призначення

Наслідування

(generalization):

Створює ієрархію користувачів із різними ролями.

Композиція:

Використовується там, де життєвий цикл одного об'єкта залежить від іншого (*Supplier → Inventory*).

					ЖИТОМИРСЬКА ПОЛІТЕХНІКА.25.121.29.000 – Лр.3	Арк.
						7
Змн.	Арк.	№ докум.	Підпис	Дата		

Агрегація:

Показує логічну приналежність об'єктів, але без строгої залежності (*Buyer* → *Vehicle*, *Inventory* → *Part*).

Асоціація:

Відображає взаємодію між сутностями (*Order* ↔ *Buyer/Supplier*, *Part* ↔ *Review*).

Таким чином ці зв'язки забезпечують цілісність даних: наприклад, видалення постачальника вплине на інвентар (композиція), а замовлення пов'язує покупця та постачальника.

Описані класи та зв'язки між ними продемонстровані на діаграмі:

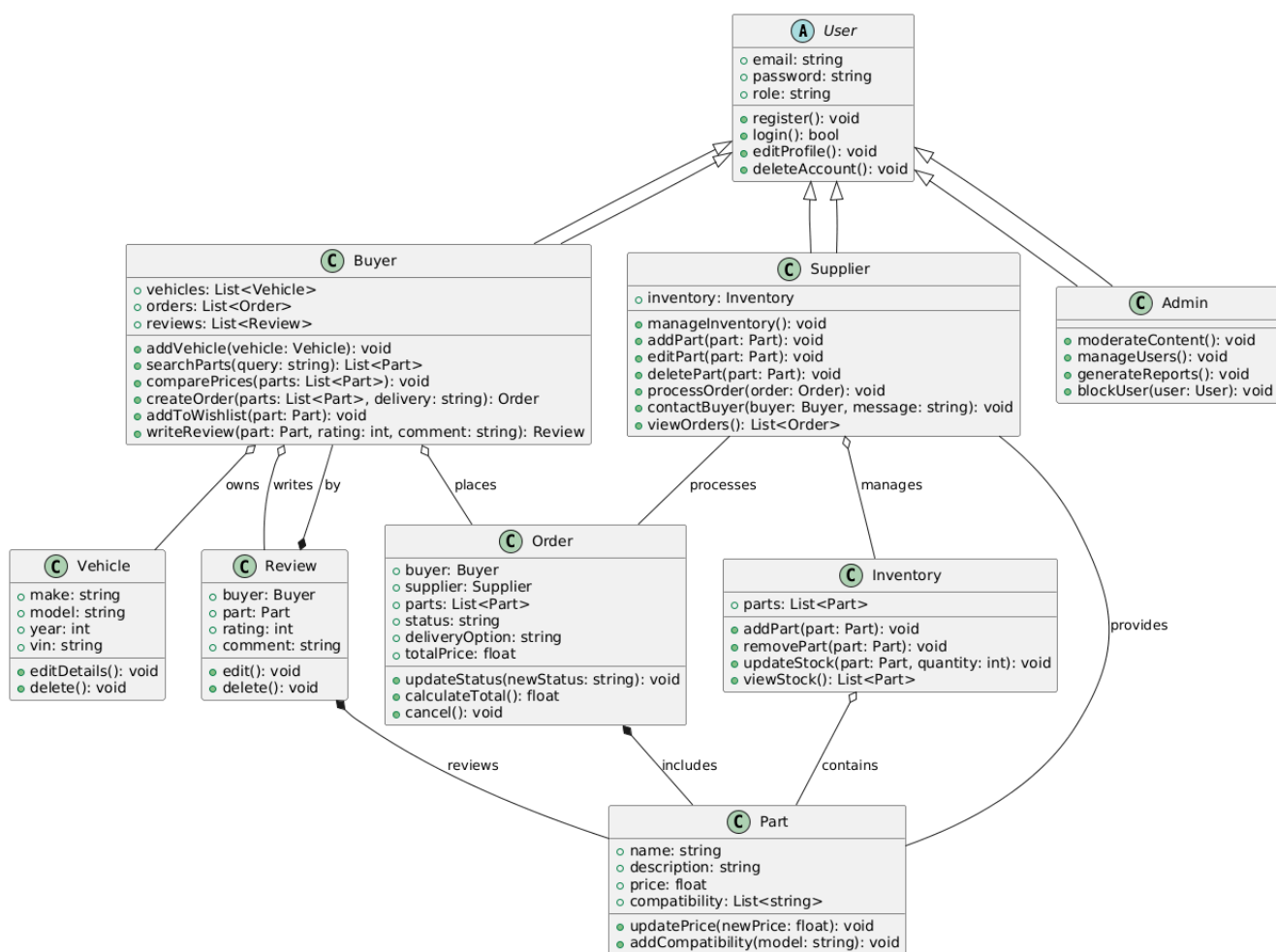


Рис. 3.1. Діаграма класів системи

Висновок: в ході виконання лабораторної роботи було набуто практичних навичок щодо побудови та використання діаграм класів системи. Побудовано та проаналізовано діаграму для власної системи. Роботу виконано у повному обсязі.