

Лабораторна робота № 8

КОДОГЕНЕРАЦІЯ

Мета: набуття практичних навичок щодо кодогенерації визначень додатку в середовище розробки.

Хід роботи

«Онлайн-сервіс для підбору автозапчастин»

Завдання.

Виконати кодогенерацію елементів проекту:

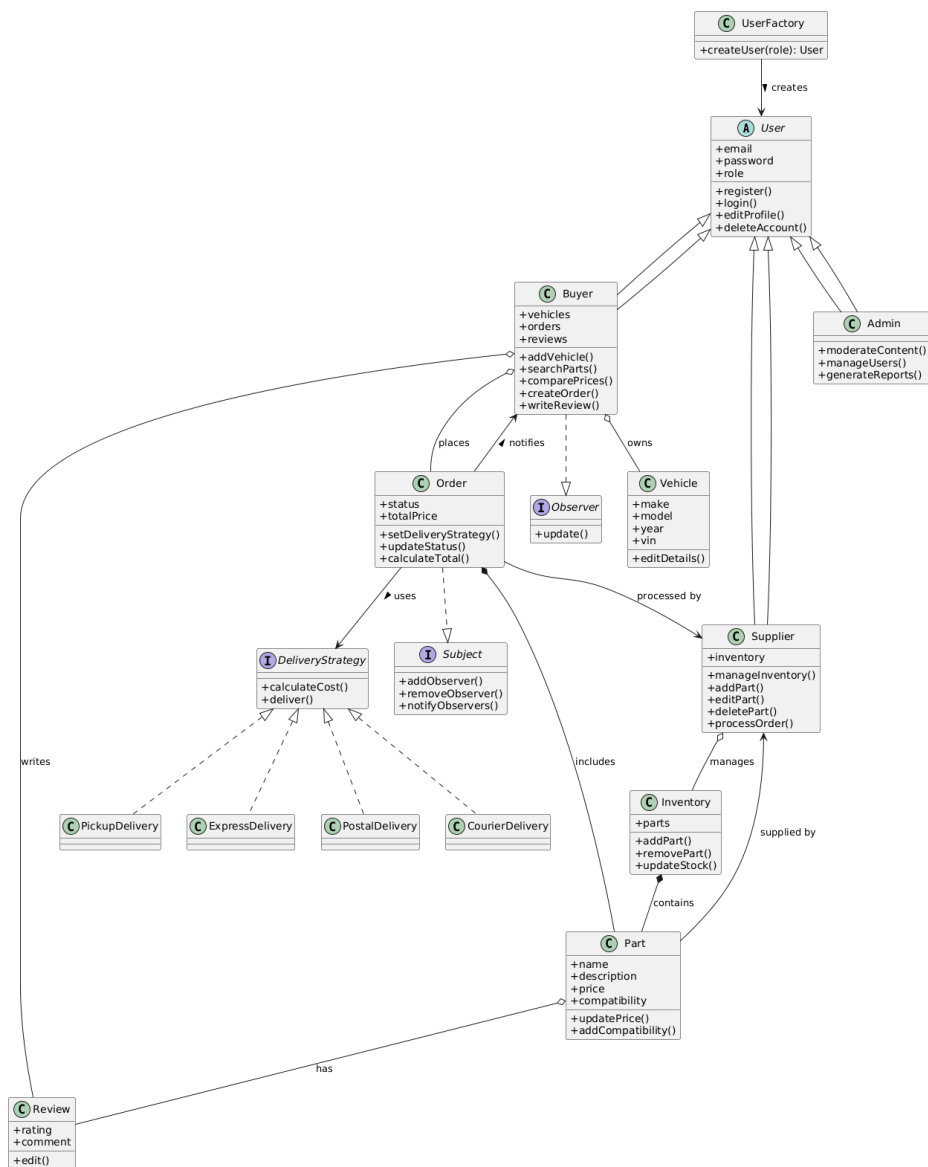


Рис. 1.1. Діаграма класів проекту

					ЖИТОМИРСЬКА ПОЛІТЕХНІКА.25.121.29.000 – Лр.8		
Змн.	Арк.	№ докум.	Підпис	Дата			
Розроб.		Трофімчук М.О.			Звіт з лабораторної роботи №8		
Перевір.		Левківський В.Л.					
Реценз.							
Н. Контр.							
Зав.каф.							
					Лім.	Арк.	Аркуші
						1	4
					ФІКТ, ар. ІПЗ-22-2		

Структура проекту. Для реалізації кодогенерації обрано мову програмування C#. Проект розбито на логічні шари: Інтерфейси, Моделі даних, Логіка та Користувачі:

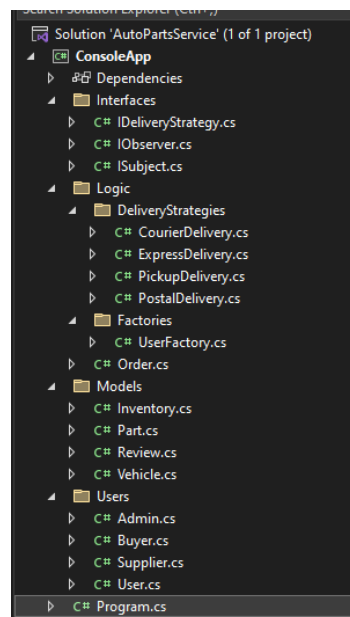


Рис. 1.2. Структура рішення проекту

Реалізація (Лістинг коду). Розглянемо лістинг основних класів проекту.

Лістинг Interfaces/IDeliveryStrategy.cs:

```
namespace AutoPartsService.Interfaces
{
    public interface IDeliveryStrategy
    {
        double CalculateCost(double orderTotal);
        void Deliver();
    }
}
```

Лістинг Interfaces/IObservable.cs:

```
namespace AutoPartsService.Interfaces
{
    public interface IObservable
    {
        void Update(string message);
    }
}
```

Лістинг Interfaces/ISubject.cs:

```
namespace AutoPartsService.Interfaces
{
    public interface ISubject
    {
        void AddObserver(IObservable observer);
        void RemoveObserver(IObservable observer);
        void NotifyObservers();
    }
}
```

Лістинг Logic/Order.cs:

```
namespace AutoPartsService.Logic
{
    public class Order : ISubject
    {
        public string Status { get; private set; }
        public double TotalPrice { get; private set; }

        private IDeliveryStrategy _deliveryStrategy;
        private List<IObserver> _observers = new List<IObserver>();

        public void SetDeliveryStrategy(IDeliveryStrategy strategy)
        {
            _deliveryStrategy = strategy;
            Console.WriteLine($"[Order] Обрано стратегію доставки:
{strategy.GetType().Name}");
        }

        public void CalculateTotal(double partsPrice)
        {
            if (_deliveryStrategy == null)
            {
                Console.WriteLine("[Order] Помилка: Спосіб доставки не обрано!");
                return;
            }
            double deliveryCost = _deliveryStrategy.CalculateCost(partsPrice);
            TotalPrice = partsPrice + deliveryCost;
            Console.WriteLine($"[Order] Розрахунок: Товар ({partsPrice}) + Доставка
({deliveryCost}) = {TotalPrice}");
        }

        public void UpdateStatus(string newStatus)
        {
            Status = newStatus;
            Console.WriteLine($"[Order] Статус змінено на: {Status}");
            NotifyObservers(); // Сповіщаємо покупця
        }

        // --- ISubject Implementation ---
        public void AddObserver(IObserver observer) => _observers.Add(observer);
        public void RemoveObserver(IObserver observer) => _observers.Remove(observer);

        public void NotifyObservers()
        {
            foreach (var observer in _observers)
            {
                observer.Update($"Ваше замовлення перейшло у статус: {Status}");
            }
        }
    }
}
```

Лістинг Logic/Factories/UserFactory.cs:

```
namespace AutoPartsService.Logic.Factories
{
    public class UserFactory
    {
        public static User CreateUser(string role)
        {
            switch (role.ToLower())
            {
                case "buyer": return new Buyer();
                case "admin": return new Admin();
                case "supplier": return new Supplier();
                default: throw new ArgumentException("Невідома роль користувача");
            }
        }
    }
}
```

					ЖИТОМИРСЬКА ПОЛІТЕХНІКА.25.121.29.000 – Лр.8	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		3

Лістинг Users/User.cs:

```
using System;

namespace AutoPartsService.Users
{
    public abstract class User
    {
        public string Email { get; set; }
        public string Password { get; set; }
        public string Role { get; set; }

        public virtual void Register() => Console.WriteLine($"[User] {Role} зареєстрований: {Email}");
        public virtual void Login() => Console.WriteLine($"[User] {Role} ({Email}) увійшов у систему.");
        public virtual void EditProfile() => Console.WriteLine("[User] Профіль оновлено.");
        public virtual void DeleteAccount() => Console.WriteLine("[User] Акаунт видалено.");
    }
}
```

Лістинг Users/Buyer.cs:

```
namespace AutoPartsService.Users
{
    public class Buyer : User, IObservable
    {
        public List<Vehicle> Vehicles { get; set; } = new List<Vehicle>();
        public List<Order> Orders { get; set; } = new List<Order>();
        public List<Review> Reviews { get; set; } = new List<Review>();

        public Buyer() { Role = "Buyer"; }

        public void AddVehicle(Vehicle vehicle)
        {
            Vehicles.Add(vehicle);
            Console.WriteLine($"[Buyer] Авто {vehicle.Model} додано до профілю.");
        }

        public void SearchParts() => Console.WriteLine("[Buyer] Пошук запчастин...");

        public Order CreateOrder(Part part)
        {
            var order = new Order();
            // Підписуємося на оновлення статусу
            order.AddObserver(this);
            Orders.Add(order);
            Console.WriteLine("[Buyer] Створено нове замовлення.");
            return order;
        }

        // Реалізація інтерфейсу IObservable
        public void Update(string message)
        {
            Console.WriteLine($" >>> [Сповіщення для {Email}]: {message}");
        }
    }
}
```

Опис реалізованих компонентів проєкту:

1. Interfaces/

Простір абстракцій. Це дозволяє зменшити зв'язність коду (loose coupling).

- IDeliveryStrategy.cs: Інтерфейс для розрахунку вартості та доставки.
- IObservable.cs: Інтерфейс для отримання сповіщень (для Buyer).

					ЖИТОМИРСЬКА ПОЛІТЕХНІКА.25.121.29.000 – Лр.8	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		4

- ISubject.cs: Інтерфейс для управління підписниками (для Order).

2. Models/

Простір допоміжних класів, які здебільшого зберігають дані.

- Part.cs: Інформація про запчастину.
- Vehicle.cs: Інформація про авто.
- Review.cs: Дані про відгук.
- Inventory.cs: Склад, що містить список Part.

3. Users/

Містить всю ієрархію наслідування від User.

- User.cs: Абстрактний клас із загальними полями (Email, Password).
- Buyer.cs: Реалізує IObserver, має методи пошуку та створення замовлення.
- Supplier.cs: Має інвентар та обробляє замовлення.
- Admin.cs: Адміністративні функції.

4. Logic/

Основне ядро взаємодії класів.

- Factories/UserFactory.cs: Статичний клас або метод, який породжує (інстанціює) потрібного користувача залежно від рядка ("admin", "buyer").
- DeliveryStrategies/: Окремі класи для кожного типу доставки (Express, Pickup...), які реалізують IDeliveryStrategy.
- Order.cs: "Диригент" процесу. Він використовує стратегію доставки та сповіщає користувача (Observer) про зміни.

Результати виконання програми:

```

Microsoft Visual Studio Debug Console
=== СИСТЕМА ПІДБОРУ АВТОЗАПЧАСТИН ===

[User] Buyer зареєстрований: driver@ua.net
[User] Supplier зареєстрований: parts@shop.com

--- Постачальник наповнює склад ---
[Inventory] Запчастину Масляний фільтр додано до інвентарю.

--- Покупець робить замовлення ---
[Buyer] Створено нове замовлення.
[Order] Обрано стратегію доставки: ExpressDelivery
[Order] Розрахунок: Товар (450) + Доставка (150) = 600

--- Обробка замовлення ---
[Supplier] Обробка нового замовлення...
[Order] Статус змінено на: Оброблено (Ready to ship)
>>> [Сповіщення для driver@ua.net]: Ваше замовлення перейшло у статус: Оброблено (Ready to ship)

[Order] Статус змінено на: Відправлено
>>> [Сповіщення для driver@ua.net]: Ваше замовлення перейшло у статус: Відправлено

Натисніть будь-яку клавішу для виходу...

```

Рис. 1.3. Демонстрація роботи програми.

					ЖИТОМИРСЬКА ПОЛІТЕХНІКА.25.121.29.000 – Лр.8	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		5

Висновок: У ході виконання лабораторної роботи було створено консольний додаток на мові С#. Реалізовано ієрархію класів згідно з UML-діаграмою. Успішно імплементовано патерни Strategy, Observer та Factory. Програма коректно виконує сценарій замовлення автозапчастин, розрахунку вартості доставки та сповіщення клієнта. Роботу виконано у повному обсязі.

					ЖИТОМИРСЬКА ПОЛІТЕХНІКА.25.121.29.000 – Лр.8	Арк.
						6
Змн.	Арк.	№ докум.	Підпис	Дата		