

Технічна документація: Naval Vessels System

Студент: Луханін Богдан Сергійович , 244(Б)

Дисципліна: Об'єктно-орієнтоване програмування (C#)

Технології: .NET 9.0, C# 13, Console Application

1. Вступ

Naval Vessels System є програмним рішенням для моделювання та управління військово-морським флотом. Система дозволяє симулювати життєвий цикл всіх бойових одиниць: від виробництва та найму екіпажу до проведення бойових маневрів

Головна мета проекту: Демонстрація принципів ООП (включно з SOLID) та побудова гнучкої архітектури для управління сутностями

2. Архітектура Системи

Проект побудований за принципом **розділення відповідальності**. Система складається з чотирьох основних шарів:

2.1. Data Models (Сутності)

Ядро системи, що описує об'єкти реального світу.

- **Vessel (Abstract Class):** Базова абстракція корабля. Містить спільні характеристики (`Name` , `ArmorThickness` , `MainWeaponCaliber` , `Speed`) та визначає контракт для всіх спадкоємців.
- **Battleship** : Спеціалізація класу `Vessel` . Особливість: має режим `сонара` (`SonarMode`), який модифікує бойові характеристики.
- **Submarine** : спеціалізація класу `Vessel` . Особливість: має режим занурення (`SubmergeMode`), що дозволяє атакувати приховано.
- **Captain** : відповідає за накопичення бойового досвіду (`CombatExperience`). Один капітан може керувати кораблем.

2.2. Logic / Controller Layer

Відповідає за бізнес-логіку та обробку команд користувача.

- **IController (Interface)**: Визначає "контракт" системи - набір публічних методів, доступних ззовні. Це дозволяє приховати реалізацію від користувача.

```
using System.Collections.Generic;

namespace NavalVessels
{
    public interface IController
    {
        string HireCaptain(string fullName);

        string ProduceVessel(string name, string vesselType, double mainWeaponCaliber, double speed);

        string AssignCaptain(string selectedCaptainName, string selectedVesselName);

        string CaptainReport(string captainFullName);

        string VesselReport(string vesselName);

        string ToggleSpecialMode(string vesselName);

        string ServiceVessel(string vesselName);

        string AttackVessels(string attackingVesselName, string defendingVesselName);
    }
}
```

- **Controller (Implementation)**: реалізує інтерфейс **IController**. Тут відбувається умовна валідація даних, створення об'єктів та виклики методів сутностей

2.3. Data Storage (Repository)

- **VesselRepository**: відповідає за зберігання колекції кораблів у пам'яті. Реалізує методи додавання (**Add**), пошуку (**FindByName**) та видалення (**Remove**).

2.4. Presentation Layer

- **Startup**: Точка входу в програму. Відповідає за зчитування команд з консолі (інпут/аутпут) та передачу їх у контролер.

3. Реалізація принципів ООП

◆ Абстракція

Ми не можемо створити екземпляр класу `Vessel`, оскільки в реальності не існує "просто корабля". Корабель завжди належить до певного типу. Тому клас `Vessel` оголошено як `abstract`.

```
public abstract class Vessel { ... }
```

◆ Інкапсуляція

Всі поля даних є приватними. Доступ до них здійснюється через публічні властивості з валідацією. Це гарантує цілісність даних (до прикладу! неможливо створити корабель без імені або з від'ємною швидкістю).

```
public string Name
{
    get ⇒ name;
    set
    {
        if (string.IsNullOrEmpty(value)) throw new ArgumentNullException
        (...);
        name = value;
    }
}
```

◆ Поліморфізм

Система обробляє різні типи кораблів уніфіковано через базовий клас.

1. **Колекція:** Ми зберігаємо всі кораблі в одному списку `List<Vessel>`, незалежно від того, чи це лінкор, чи субмарина.
2. **Override:** Метод `ToString()` перевизначено у кожному класі для виведення специфічної інформації (наприклад, статус сонара для лінкора).
3. **runtime поведінка:** метод `RepairVessel()` викликається то однаково, але для лінкора відновлює здоров'я до 300, а от для субмарини - до 200.

4. Ключові алгоритми та логіка

4.1. Механіка Атаки (**AttackVessels**)

Алгоритм бою реалізовано наступним чином:

1. Перевірка: атакуючий та жертва не повинні бути знищеними (Armor > 0)
2. агресор додає жертву у свій список цілей (**Targets**).
3. Жертва отримує пошкодження, що дорівнюють калібру головної гармати Агресора (**MainWeaponCaliber**)
4. Якщо броня падає нижче нуля, вона встановлюється в 0
5. Капітан агресора отримує +10 до свого досвіду (**CombatExperience**).

```
public string AttackVessels(string attackingVesselName, string defendingVesselName)
{
    Vessel attacker = this.vessels.FindByName(attackingVesselName);
    Vessel defender = this.vessels.FindByName(defendingVesselName);

    if (attacker == null)
    {
        return $"Vessel {attackingVesselName} could not be found.";
    }

    if (defender == null)
    {
        return $"Vessel {defendingVesselName} could not be found.";
    }

    if (attacker.ArmorThickness == 0)
    {
        return $"Unarmored vessel {attackingVesselName} cannot attack or be attacked.";
    }

    if (defender.ArmorThickness == 0)
    {
        return $"Unarmored vessel {defendingVesselName} cannot attack or be attacked.";
    }

    attacker.Attack(defender);
    attacker.Captain.IncreaseCombatExperience();
    defender.Captain.IncreaseCombatExperience();

    return $"Vessel {defendingVesselName} was attacked by vessel {attackingVesselName} - current armor thickness: {defender.ArmorThickness}.";
}
```

4.2. Перемикання режимів (**ToggleSpecialMode**)

Використано механізм зміни стану (State Toggle).

- **Для Battleship (Sonar Mode):**
 - on: Калібр зброї +40, Швидкість -5.
 - off: Калібр зброї -40, Швидкість +5.
- **Для Submarine (Submerge Mode):**

- on: Калібр зброї +40, Швидкість -4.
- off: Калібр зброї -40, Швидкість +4.

```
public string ToggleSpecialMode(string vesselName)
{
    Vessel vessel = this.vessels.FindByName(vesselName);

    if (vessel == null)
    {
        return $"Vessel {vesselName} could not be found.";
    }

    if (vessel is Battleship battleship)
    {
        battleship.ToggleSonarMode();
        return $"Battleship {vesselName} toggled sonar mode.";
    }
    else if (vessel is Submarine submarine)
    {
        submarine.ToggleSubmergeMode();
        return $"Submarine {vesselName} toggled submerge mode.";
    }

    return string.Empty;
}
```

5. Діаграма класів (UML)

Візуалізація зв'язків:

- **Battleship** is a **Vessel**
- **Submarine** is a **Vessel**
- **Vessel** has a **Captain**
- **Controller** uses **VesselRepository**

6. Висновок

В ході виконання поставленого завдання було розроблено надійну систему з використанням найкращих практик C#. Архітектура дозволяє легко розширювати функціонал (наприклад, додати новий тип корабля **Destroyer** або будь-якого іншого), не змінюючи основну логіку контролера, що відповідає принципу **Open/Closed Principle**.

7. Очікуваний вхід / вихід

HireCaptain Chester_Nimitz
Captain Chester_Nimitz is hired.
HireCaptain Karl_Donitz
Captain Karl_Donitz is hired.
ProduceVessel USS_Colorado Battleship 16 21
Battleship USS_Colorado is manufactured with the main weapon caliber of 16 inches and a maximum speed of 21 knots.
AssignCaptain Chester_Nimitz USS_Colorado
Captain Chester_Nimitz command vessel USS_Colorado.
ToggleSpecialMode USS_Colorado
Battleship USS_Colorado toggled sonar mode.
VesselReport USS_Colorado
USS_Colorado
*Type: Battleship
*Armor thickness: 300
*Main weapon caliber: 56
*Speed: 16 knots
*Targets: None
*Sonar mode: ON
Quit

HireCaptain Chester_Nimitz
Captain Chester_Nimitz is hired.
HireCaptain Harald_Lange
Captain Harald_Lange is hired.
ProduceVessel USS_Colorado Battleship 16 21
Battleship USS_Colorado is manufactured with the main weapon caliber of 16 inches and a maximum speed of 21 knots.
ProduceVessel U-505 Submarine 21.1 18.2
Submarine U-505 is manufactured with the main weapon caliber of 21.1 inches and a maximum speed of 18.2 knots.
AssignCaptain Chester_Nimitz USS_Colorado
Captain Chester_Nimitz command vessel USS_Colorado.
AssignCaptain Harald_Lange U-505

Captain Harald_Lange command vessel U-505.

AttackVessels USS_Colorado U-505

Vessel U-505 was attacked by vessel USS_Colorado - current armor thickness: 184.

VesselReport USS_Colorado

- USS_Colorado

*Type: Battleship

*Armor thickness: 300

*Main weapon caliber: 16

*Speed: 21 knots

*Targets: U-505

*Sonar mode: OFF

VesselReport U-505

- U-505

*Type: Submarine

*Armor thickness: 184

*Main weapon caliber: 21.1

*Speed: 18.2 knots

*Targets: None

*Submerge mode: OFF

CaptainReport Chester_Nimitz

Chester_Nimitz has 10 combat experience and commands 1 vessels.

- USS_Colorado

*Type: Battleship

*Armor thickness: 300

*Main weapon caliber: 16

*Speed: 21 knots

*Targets: U-505

*Sonar mode: OFF

Quit