МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет інформаційних технологій

Кафедра комп’ютерної інженерії та інформаційних систем

Лабораторна робота №5

# з дисципліни «Обєктно-орієнтовані технології програмування»

на тему:

«Поведінкові патерни»

Виконав:

студент 1 курсу, групи КІ2м-23-3 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Гоцалюк М.М.

(Підпис)

Перевірив: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Лисенко С.М.

(Підпис)

Хмельницький - 2024

**Третій рівень.**

### Ітератор (Iterator).

(code1.ts)

NOTE: "цей патерн буде корисний якщо у вас є складна структура даних, і ви хочете приховати від клієнта деталі її реалізації, або коли потрібно мати кілька варіантів перебору колекцій"

В рамках виконання задачі реалізації колекцї вправ використати патерн Ітератор, для додавання інкапсульованого перебору колекції. Реалізувати інтерфейс ітератора (Iterator) з абстрактними методами next() та hasNext() та конкретний ітератор (клас ExerciseIterator) на його основі. Реалізувати колекцію вправ (клас ExerciseCollection) яка імплементує інтерфейс IterableCollection, що в свою чергу забезпечить обовязкову реалізацію методу createIterator(), також додати можливість додавання (метод addExercise()) та видалення (метод deleteExercise()) вправ з колекції.

### Стан (State).

(code2.ts)

Використати патерн Стан для реалізації стану клубного абонемента, створити інтерфейс State який має два абстракитних методи applyState() - метод який примінить логіку зміни стана та getName() для перевірки поточного стану. Стан абонемента моє три варіанта і всі вони повині імплементувати інтерфейс State і реалізовувати потрібну логіку у необхадних методах: активний (клас ActiveState), призупинений (клас SuspendedState) та неактивний (клас ExpiredState). Це має функціонувати в контексті вже реалізованого абонемента (клас Membership) за допомогою методів setState() та applyState().

### Ланцюжок Обов'язків (Chain of responsibility).

(code3.ts)

Для виконання задачі реалізації реєстраційної логіки застосувати застосувати шаблон Ланцюжок Обов'язків. Реалізувати наступну логіку: 1й крок реєстрація (клас RegistrationHandler), 2й авторизація (клас AuthorizationHandler), 3й (клас DisclaimerHandler) якщо користувач не підписав дисклеймер надати таку можливість інакше пропустити, 4й крок налаштування профілю (клас ProfileSetupHandler)якшо користувач робить авторизацію вперше інакше пропустити , 5й (клас StartUseHandler).

Створити базовий клас обробника подій (клас Handler) для наслідування конкретними реалізаціями подій, який забовяже реалізацію метода setSuccessor() - визначатиме наступну подію в ланцюжку виконання, та також методу handle() для її виконання.