МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет інформаційних технологій

Кафедра комп’ютерної інженерії та інформаційних систем

Лабораторна робота №3

# 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.**

Стратегія (Strategy).

(code1.ts)

В рамках виконання задачі реалізації стратегій визначення оплати абонементів для користувачів спортзалу застосуєм патерн Стратегія. Має бути створений інтерфейс (клас MembershipPriceContext), який на основі типу (enum MembershipPeriodType) буде підбирати відповідну стратегію обрахунку та повертати ціну (метод getPrice()).

Тож для цього потрібно реалізувати сімейство схожих алгоритмів на основі інтерфейсу (MembershipStrategy) який забовязує реалізувати метод (calculatePrice(): number;).

Спостерігач (Observer).

(code2.ts)

Для виконання задачі, в ході якої потрібно реалізувати надсилання сповіщень користувачам про найближчі тренування, потрібно застосувати патерн Спостерігач. Для цього реалізуєм інтерфейс спостерігача (Announcement), який вимагатиме реалізацію надсилання сповіщень (метод update(training: string): void). Клас користувача (клас User) повинен імплементувати інтерфейс спостерігача та реалізовувати його методи. Далі в реалізованому роскладі тренувань (клас TrainingSchedule) організувати автоматичне оповіщення користувачів при створені нового тренування (метод setTraining()) за допомогою методу notifyUsers().

Команда (Command).

(code3.ts)

Для виконання задачі на реалізацію функціонала взаємодії з вправами, а саме для реалізації відміток про виконання вправ (check/uncheck) використати патерн Команда. На основі інтерфейсу Command, який забовязує реалізацію методу виконання команди (execute(): void;), створити дві команди (клас MarkExerciseCommand) та (клас UndoMarkExerciseCommand) які реалізовують потрібну логіку. А також реалізувати клас інвокер (клас CommandInvoker), для керування командами, який повинен вміти вибирати потрібну команду (метод setCommand()) та виконувати її (метод executeCommand())