**Міністерство освіти і науки України**

**Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського"**

**Факультет інформатики та обчислювальної техніки**

**Кафедра інформатики та програмної інженерії**

**Звіт**

з комп’ютерного практикуму №6 з дисципліни

«Основи розробки програмного забезпечення на платформі Microsoft.NET»

**«Шаблони проектування. Поведінкові шаблони»**

**Виконав**

(шифр, прізвище, ім'я, по батькові)

*Голяка Д.В.*

**Перевірила**

(прізвище, ім'я, по батькові)

*Ліщук К.І.*

**Варіант 16**

**Умова**

Реалізувати завдання прокату обладнання. При цьому необхідно розраховувати вартість обслуговування в залежності від способів і умов прокату. Способи та умови можуть бути стандартними, пільговими і штрафними. Можлива поява нових умов. Необхідно реалізувати конкретні алгоритми розрахунку для кожного поєднання пропонованої послуги і способу її придбання

**UML-діаграма та код**

Робота містить багато коду, тому прикріплюю посилання на GitHub, де можна зручно переглянути весь код і UML-діаграму: [GitHub](https://github.com/DmytroHoliaka/labs-dotnet/tree/feature/lab6/Resources/BehavioralPatterns)

**Опис**  
У даній роботі було використано два поведінкові та два породжувальні патерни:

- Strategy для надання можливості динамічно змінювати алгоритм обчислення вартості обладнання;

- Command для перетворення запиту на об'єкт, який зберігається у списку для можливості відновлення попередніх станів;

- Singleton для гарантії існування лише одного екземпляра контексту;

- Factory Method для створення конкретних стратегій через перевірку вхідного параметра.

**Скріншоти результатів виконання**

A screen shot of a computer program

Description automatically generated

Рисунок 1 – Демонстрація роботи системи із використанням історії

**Висновок**

Упродовж виконання лабораторної роботи було вивчено поведінкові патерни проєктування. Після теоретичних відомостей, було створено програмне забезпечення, де я на практиці закріпив вивчені патерни. У роботі я використав Strategy для того, щоб колистувач під час виконання програми міг динамічно змінювати алгоритми обчислення, Command для перетворення запиту у об’єкт та збереження його у списку для імітації історії дій. Крім цього, я використав поведінкові патерни Singleton для гарантії існування лише одного екземпляру контексту даних та Factory Method для створення конктерних стратегій, що визначаються через вхідний параметр статичного методу.