Лабораторна робота №8-9
Вступ до паттернів
(максимум 10 балів)

Мета: Набуття практичних навичок роботи з деякими складнішими у реалізації паттернами в C# на прикладі модифікації вже розробленого консольного додатку, який імітує роботу того чи іншого підприємства або установи.

Завдання

1. Для лабораторної роботи 6-7 вдало реалізувати:
   - (5 бали) породжуючий паттерн проектування “Фабричний метод”;
   - (3 бали) поведінковий паттерн проектування “Спостерігач”.
2. (2 бали) Побудувати відповідні UML-діаграми.

Зрозуміло, що студент повинен повністю розуміти код. Щодо UML-діаграм, то оцінка виставлятиметься, в першу чергу, за старання, а не за правильність побудови.

Коротка характеристика паттернів

Породжуючі (якщо ціллю є створення об’єкта)

Одинак – якщо щось має стати єдиним у програмі: адміністратор, база даних тощо.
Прототип – якщо потрібно створити точну копію деякого об’єкта (не виключено, що, при цьому, вноситиметься певна невелика зміна значень параметрів методу).
Будівельник – якщо необхідно створити шаблон із послідовним виконанням низки методів.
Фабричний метод – якщо потрібно, щоб повернувся той чи інший клас, залежно від тих чи інших умов.
Абстрактна фабрика – якщо в програмі є кілька фабричних методів, то варто об'єднати їх в єдину фабрику.

Структурні (якщо ціллю є підвищення зручності)

Фасад – якщо має виконатись певний перелік методів залежно від тієї чи іншої ситуації.
Адаптер – якщо потрібно адаптувати деякий існуючий функціонал до нового.
Проксі (заступник) – якщо потрібно створити клас, який би керував іншим класом. *Досить зрозумілий приклад в книзі А. Будая.*

Декоратор – якщо існуючий функціонал певного класу потрібно доповнити новим. *Досить зрозумілий приклад в книзі А. Будая.*
Міст – якщо деякий функціонал потрібно розділити на 2 класи з метою уникнення дубляжу коду.
Легковаговик – якщо у величезному масиві об'єктів часто зустрічаються поля з однаковими ресурсоємним значеннями. *Досить зрозумілий приклад на сайті refactoring.guru.*
Компонувальник – щось там з бінарними деревами та рекурсією.

Поведінкові (якщо ціллю є організація зв'язного "спілкування" між об'єктами)

Ланцюг обов'язків (Chain of Responsibility) – якщо необхідно "методом спроб і помилок" знайти той метод, який "візьметься" обробляти запит.
Команда (Command) – якщо необхідно створити об'єкт–передатчик від об'єкта А до об'єкта Б. Це дозволяє організовувати відкладення операцій, їх відміну, запам'ятовування тощо.
Ітератор (Iterator) – якщо потрібно пройтись по багатьом об'єктам програми.
Посередник (Mediator) – якщо доцільним є створення метода–передатчика, який приймає запити від інших методів і за допомогою switch передає керування іншому методу. Цей паттерн мало ефективний при роботі з консоллю.
Хранитель (Memento) – якщо потрібно зберігати/відновлювати попередній(–ні) стан(–ни) об'єкта(–ів).
Наглядач (Observer) – якщо необхідно реалізувати механізм підписки.
Стан (State) – якщо потрібно, щоб об'єкт набував ті чи інші аспекти поведінки залежно від значення якогось із глобальних полів.
Стратегія (Strategy) – якщо потрібно, щоб об'єкт набував ті чи інші аспекти поведінки залежно від вибраного методу–обробника.
Шаблонний метод (Template Method) – якщо можемо виділити якусь стандартизовану послідовність викликів методів. При цьому, частину з кроків можна перевизначати (override).
Відвідувач (Visitor) – якщо потрібно заглянути у деякі класи для того, щоб дістати й використати їх вміст.

Рекомендована література

1. Сайт https://metanit.com/sharp/tutorial/.
2. Сайт https://refactoring.guru/uk/design-patterns:
    - https://refactoring.guru/uk/design-patterns/catalog
    - https://refactoring.guru/uk/design-patterns/csharp

3. Книга https://learn.ztu.edu.ua/pluginfile.php/632/mod_resource/content/1/DesignPatterns_AndriyBuday.pdf.
4. Текстовий лекційний матеріал.

Додаткова література

1. https://www.youtube.com/watch?v=dhnsegiPXoo&list=PLLWMQd6PeGY3ob0Ga6vn1czFZfW6e-FLr
2. https://www.youtube.com/watch?v=VqgXn7wsPsc&list=PLuGqgO5WmeGOGGbaBwJvvQ7_Su6cIBc8C