Міністерство освіти і науки України

Національний лісотехнічний університет України

Кафедра інформаційних технологій

**Модульна контрольна робота №3**

з навчальної дисципліни

**ООАП**

**Варіант - №8**

**Виконав:**

Студент КН - 41/2

Юрас Назар

**Перевірив:**

Яркун В.І

Львів – 2020

Навести опис поведінкового шаблону Стратегія: суть патерна, структура,

застосування, переваги та недоліки.

**Стратегія** — це поведінковий патерн проектування, який визначає сімейство схожих алгоритмів і розміщує кожен з них у власному класі. Після цього алгоритми можна заміняти один на інший прямо під час виконання програми.

**Коли застосовувати:**

* Якщо вам потрібно використовувати різні варіації якого-небудь алгоритму всередині одного об’єкта.
* Якщо у вас є безліч схожих класів, які відрізняються лише деякою поведінкою.
* Якщо ви не хочете оголювати деталі реалізації алгоритмів для інших класів.
* Якщо різні варіації алгоритмів реалізовано у вигляді розлогого умовного оператора. Кожна гілка такого оператора є варіацією алгоритму.

**Структура:**

* **Контекст** зберігає посилання на об’єкт конкретної стратегії, працюючи з ним через загальний інтерфейс стратегій.
* **Стратегія** визначає інтерфейс, спільний для всіх варіацій алгоритму. Контекст використовує цей інтерфейс для виклику алгоритму.
* Для контексту неважливо, яка саме варіація алгоритму буде обрана, оскільки всі вони мають однаковий інтерфейс.
* **Конкретні стратегії** реалізують різні варіації алгоритму.
* Під час виконання програми контекст отримує виклики від клієнта й делегує їх об’єкту конкретної стратегії.
* Клієнт повинен створити об’єкт конкретної стратегії та передати його до конструктора контексту. Крім того, клієнт повинен мати можливість замінити стратегію на льоту, використовуючи сетер поля стратегії. Завдяки цьому, контекст не знатиме про те, яку саме стратегію зараз

**Переваги:**

* Гаряча заміна алгоритмів на льоту.
* Ізолює код і дані алгоритмів від інших класів.
* Заміна спадкування делегуванням.
* Реалізує *принцип відкритості/закритості*.

**Недоліки:**

* Ускладнює програму внаслідок додаткових класів.
* Клієнт повинен знати, в чому полягає різниця між стратегіями, щоб вибрати потрібну.