

Тест до практичної роботи 3 ІПЗ

kradozhon.maxim@lil.kpi.ua [Switch account](#)



* Indicates required question

Email *

Your email

ПІБ *

Your answer

Група *

Choose

Дати визначення поняттю **шаблон проектування**

2 points

- ☐ Шаблон проектування - це мова програмування для вираження архітектурних концепцій та взаємодії між об'єктами в програмі. Використання шаблонів дозволяє розробникам говорити на спільній мові та швидше розуміти структуру системи.
- ☐ Шаблон проектування - це своєрідний паттерн, який надає готовий скрипт для вирішення певної задачі. Використання шаблонів дозволяє розробникам швидко впроваджувати рішення, що вже були успішно випробувані та оптимізовані
- ☐ Шаблон проектування - це абстрактний шлях до вирішення типових завдань у програмуванні. Використання шаблонів допомагає уникнути зайвого коду та сприяє створенню розширюваних та зручних у використанні програм.
- ☐ Шаблон проектування - це архітектурний підхід, що визначає структуру та взаємодію об'єктів для ефективного вирішення проблеми. Це як готовий конструктор, який можна використовувати для створення різних видів програм.
- ☐ Шаблон проектування - це загальний та перевірений спосіб вирішення типових проблем в програмному проектуванні. Він включає в себе структуру та взаємодію об'єктів, класів та компонентів, що спрощує розробку, розширення та обслуговування програмного забезпечення.

Встановити відповідність між назвою шаблону та його типом

3 points

	Породжувальні	Поведінкові	Структурні
Декоратор	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ітератор	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Адаптер	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Проксі	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Медіатор	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Одинак	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Міст	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Фабрика	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Композит	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Прототип	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Знімок	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Спостерігач	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

дозволяє клієнтам обробляти окремі об'єкти та їх композиції однаковим чином. Він створює ієрархію з компонентів, де як окремий об'єкт і його складові можуть бути оброблені інколи навіть ідентично.

використовується тоді, коли вам потрібно вбудувати об'єкти в структуру дерева для представлення частини-цілого в єдинообразний спосіб.

надає механізм динамічного додавання нових функціональностей об'єктам без зміни їхньої структури.

базується на ідеї композиції класів для створення нових об'єктів додатковими можливостями

Декоратор	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Проксі	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Композит	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Що є основними елементами паттерну *Композит*?

1 point

- ☐ Реальний об'єкт
- ☐ Папка
- ☐ Контейнер
- ☐ Файл
- ☐ Дерево
- ☐ Листок
- ☐ Компонент

Які елементи входять до паттерну *Proxy*?

1 point

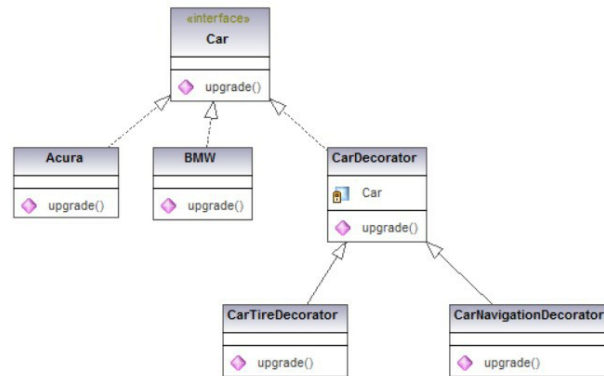
- ☐ Real Subject
- ☐ Container
- ☐ Component
- ☐ Subject
- ☐ Proxy
- ☐ Leaf

В яких випадках слід застосовувати паттерн декоратор?

2 points

- ☐ Коли необхідно контролювати або обмежувати доступ до методів об'єкта, додаючи додаткову логіку.
- ☐ Якщо потрібно реалізувати лінійне завантаження об'єктів та ініціалізувати їх тільки в момент фактичного використання.
- ☐ Необхідно зберігати результати попередніх викликів для покращення продуктивності.
- ☐ Якщо необхідно додавати функції до об'єктів на льоту та забезпечувати гнучку альтернативу успадкуванню.
- ☐ Коли вам потрібно динамічно додавати нову функціональність об'єктам без зміни їхньої структури
- ☐ Якщо вам потрібно комбінувати різні функціональні елементи об'єктів, але створення великої кількості підкласів не є практичним.

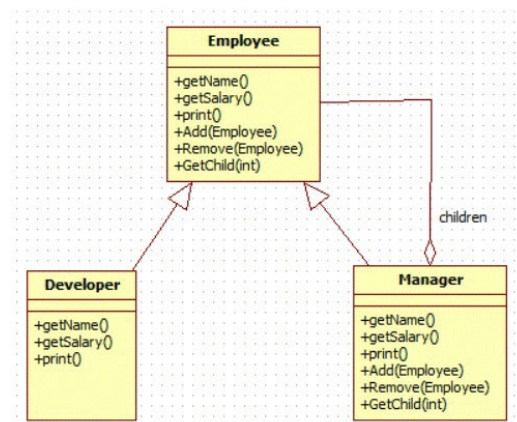
Встановити відповідність між класом або інтерфейсом на діаграмі нижче та компонентом (учасником) шаблону. 2 points



Component Decorator ConcreteDecorator ConcreteCompor

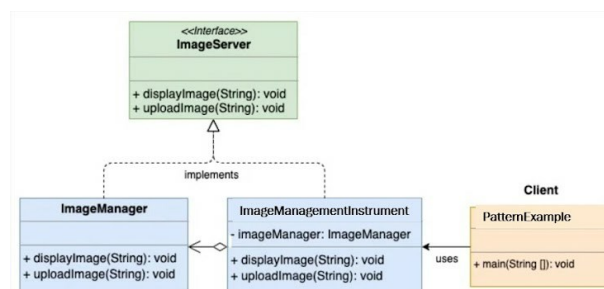
Car	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Acura	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
CarDecorator	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
CarNavigationDecorator	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
BMW	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
CareTireDecorator	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Якому шаблону проектування відповідає дана схема? Встановити відповідність між класом або інтерфейсом на діаграмі нижче та компонентом (учасником) шаблону. 3 points



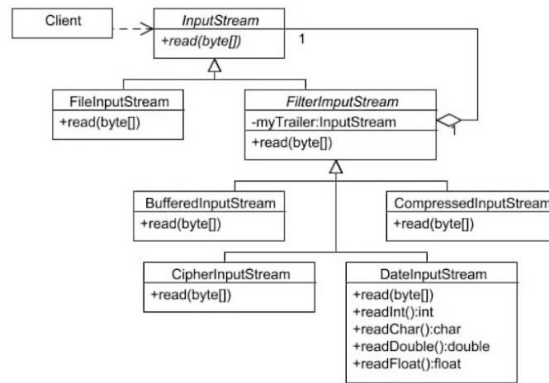
Your answer

Якому шаблону проектування відповідає дана схема? Встановити відповідність між класом або інтерфейсом на діаграмі нижче та компонентом (учасником) шаблону. 3 points



Your answer

Написати на мові Java як буде виглядати на клієнті створення екземпляра 2 points
класу **FileInputStream** (конструктор має приймати параметр *шлях до файлу*). Обгорнути даний об'єкт в декоратор *BufferedInputStream* та викликати метод read.



Your answer

Quilgo Test ID *
This question is filled automatically DO NOT EDIT OR REMOVE

Submit

Clear form

Never submit passwords through Google Forms.

This form was created inside of National Aviation University. [Report Abuse](#)

Google Forms

