

Wzorzec Dekorator jest przydatny, gdy chcemy mieć elastyczność w dodawaniu funkcjonalności do obiektów bez konieczności tworzenia dużej liczby pochodnych klas lub modyfikowania istniejącego kodu. Dzięki temu wzorcowi można łatwo rozszerzać funkcjonalność obiektów bez konieczności zmiany ich struktury. Najczęstsze zastosowanie tego wzorca to:

1. Rozszerzanie funkcjonalności klas bazowych
Dekoratory pozwalają dynamicznie dodawać nowe funkcje do istniejących klas, bez konieczności modyfikacji tych klas.
2. Wzmacnianie obiektów:
Pozwala to na elastyczne dodawanie nowych zachowań do obiektów już istniejących w czasie działania programu.
3. Utrzymywanie pojedynczej odpowiedzialności
Wzorzec Dekorator pomaga utrzymać zasadę pojedynczej odpowiedzialności (Single Responsibility Principle), ponieważ każdy dekorator może być odpowiedzialny tylko za jedną konkretną funkcjonalność.
4. Łączenie funkcjonalności
Dekoratory mogą być łączone, co pozwala na tworzenie złożonych obiektów z wieloma różnymi funkcjonalnościami.