# Adapter

## Zalety:

+ Możliwość ponownego użycia

+ Większa elastyczność

+ Mniejsze skomplikowanie podstawowych klas, przez możliwość użycia polimorfizmu do przełączania między różnymi implementacjami adapterów

## Wady:

- Wszystkie żądania są przekazywane dalej, co wpływa na obniżenie efektywności.

- Czasami konieczne jest wielokrotne dostosowywanie, aby osiągnąć oczekiwany rezultat.

# Dekorator

## Zalety:

+ Zwiększona adaptacyjność

+ Rozszerzanie funkcji bez konieczności dziedziczenia

+ Estetyczny kod

+ Bardziej zoptymalizowane funkcjonalności

## Wady:

- Wysoka liczba obiektów

- Problemy z debugowaniem

- Znaczna złożoność oprogramowania

Dziedziczenie(proste i złożone)

## Zalety:

+ Ponowne wykorzystanie kodu

+ Rozszerzanie i dostosowywanie potrzeb "niższych" klas

+ Umożliwia klasyfikację obiektów do różnych grup o wspólnych cechach

## Wady:

- Skomplikowanie

- Potencjalne wystąpienie niejednoznaczności