

Wzorzec projektowy: polecenie

Wzorzec polecenie rozwiązuje problem oddzielenia obiektu wywołującego, od obiektu, który wykonuje operację. Przypadek ten jest w szczególności przydatny, w przypadku, gdy potrzebujemy:

- rejestrować historię operacji np. cofanie/ponawianie działań
- kolejkować żądania (np. zlecenia do wykonania w innym wątku lub systemie)
- dynamicznie konfigurować listę operacji do wykonania
- umożliwić makra, czyli zestawy powiązanych komend

Składa się z kilku głównych elementów:

1. **Command** – interfejs lub klasa abstrakcyjna definiująca metodę `execute()`.
2. **ConcreteCommand** – konkretny obiekt polecenia, który implementuje metodę `execute()` i deleguje zadanie do odpowiedniego odbiorcy (*Receiver*).
3. **Receiver** – obiekt wykonujący rzeczywistą operację.
4. **Invoker** – obiekt wywołujący polecenie (np. przycisk w GUI).
5. **Client** – tworzy polecenia i konfiguruje ich powiązania.

Implementacja podanego wzorca niesie za sobą konsekwencje, zarówno negatywne jak i również pozytywne, i są to:

Do głównych zalet, należy łatwość implementacji funkcji cofania i powtarzania, wspieranie kolejkowanie i rejestrowanie poleceń, zwiększenie elastyczności i spójności kodu, oraz łatwa możliwość konfigurowania działań.

Implementacja tego wzorca niesie również za sobą negatywne obciążenie, takie jak możliwe zwiększenie liczby klas (każde polecenie jako osobna klasa), oraz potencjalne rozdrobnienie kodu – każda prosta operacja może wymagać dodatkowego obiektu

Problemy jakie dany wzorzec rozwiązuje:

Rozwiązuje problem ciasnego powiązania obiektów wywołujących akcję, a obiektem wykonującym akcję. Polecenie zamyka żądanie wykonania akcji w jednym obiekcie, przez co obiekt wywołujący nie musi znać szczegółów implementacji, można tworzyć dynamiczne i łatwo modyfikowalne polecenia.

Przykład:

Interfejs Command definiuje metodę execute().

OnCommand to ConcreteCommand, który implementuje metodę execute().

Receiver to interfejs, a implementatorzy muszą zapewnić implementację dla metod.

TV i DVDPlayer to dwa typy odbiorników, które są przekazywane do ConcreteCommand, podobnie jak OnCommand.

Invoker zawiera polecenie. Jest to klucz do oddzielenia nadawcy od odbiorcy.

Invoker otrzymuje OnCommand ->, który wywołuje odbiornik (TV) w celu wykonania tego polecenia.