# Wzorce projektowe w programowaniu

Szymon Rychter

#### Spis treści

- 1. Czym są wzorce projektowe?
- 2. Zalety wzorców projektowych
- 3. Podział wzorców projektowych
- 4. Wzorce kreacyjne
  - Metoda wytwórcza
  - Budowniczy
  - Prototyp
- 5. Wzorce strukturalne
  - Adapter
  - Most
  - Kompozyt
- 6. Wzorce czynnościowe (behawioralne/operacyjne)
  - Strategia
  - Obserwator
  - Mediator
  - Iterator
  - Polecenie
- 7. Źródła

## Czym są wzorce projektowe?

**Wzorce projektowe** to sprawdzone rozwiązania często powtarzających się problemów projektowych w programowaniu obiektowym.



## Zalety wzorców projektowych

- 1. Uniwersalność wzorce projektowe są uniwersalne i mogą być stosowane w różnych projektach i językach programowania.
- 2. Sprawdzone rozwiązania wzorce projektowe to sprawdzone rozwiązania, które są wypracowane przez doświadczonych programistów i mają potwierdzoną skuteczność w praktyce.
- 3. Wspólny język wzorce projektowe stanowią wspólny język dla programistów i pozwalają na łatwiejsze zrozumienie i komunikację w zespole.
- 4. Zwiększenie efektywności stosowanie wzorców projektowych pozwala na szybsze i efektywniejsze tworzenie kodu źródłowego oraz na łatwiejsze wprowadzanie zmian.
- 5. Ułatwienie utrzymania kodu wzorce projektowe ułatwiają utrzymanie kodu źródłowego poprzez zwiększenie jego czytelności i struktury.

## Podział wzorców projektowych

Wzorce projektowe można podzielić na trzy rodziny według tego, co robią:

- Wzorce kreacyjne
- Wzorce strukturalne
- Wzorce czynnościowe (behawioralne lub operacyjne)



## Wzorce kreacyjne

**Wzorce kreacyjne** - opisują proces tworzenia nowych obiektów poprzez inicjalizację i konfigurację klas oraz innych typów danych.

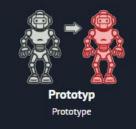






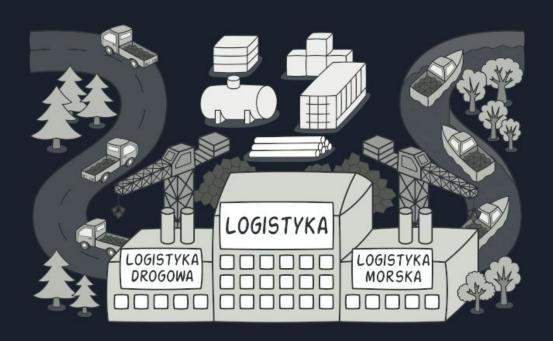






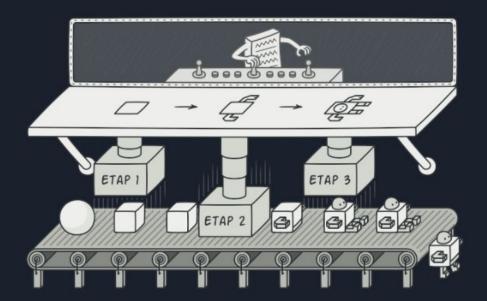
## Metoda wytwórcza

**Metoda wytwórcza** jest kreacyjnym wzorcem projektowym, który udostępnia interfejs do tworzenia obiektów w ramach klasy bazowej, ale pozwala podklasom zmieniać typ tworzonych obiektów.



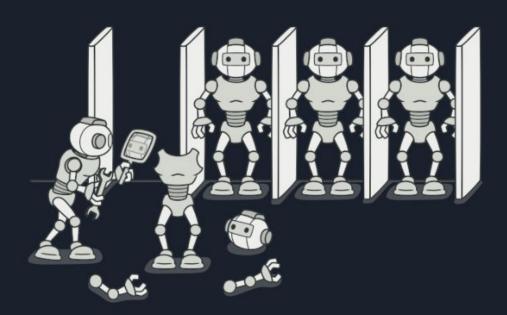
#### Budowniczy

**Budowniczy** jest kreacyjnym wzorcem projektowym, który daje możliwość tworzenia złożonych obiektów etapami, krok po kroku. Wzorzec ten pozwala produkować różne typy oraz reprezentacje obiektu używając tego samego kodu konstrukcyjnego.



## Prototyp

**Prototyp** to kreacyjny wzorzec projektowy, który umożliwia kopiowanie już istniejących obiektów bez tworzenia zależności pomiędzy twoim kodem, a klasami obiektów.



#### Wzorce strukturalne

Wzorce strukturalne - opisują struktury obiektów i ich wzajemnych powiązań.









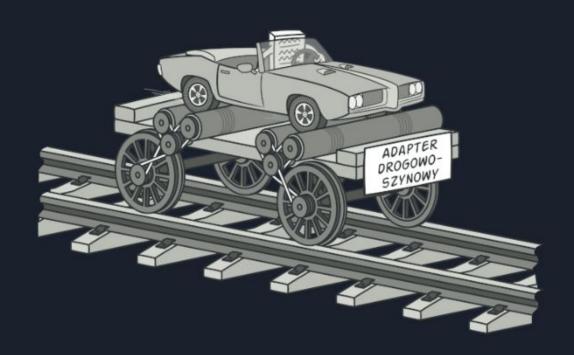






## Adapter

Adapter jest strukturalnym wzorcem projektowym pozwalającym na współdziałanie ze sobą obiektów o niekompatybilnych interfejsach.



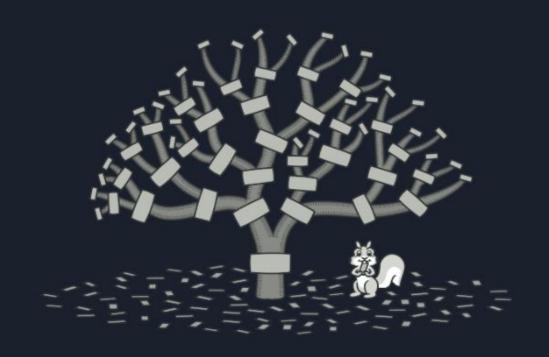
#### Most

Most jest strukturalnym wzorcem projektowym pozwalającym na rozdzielenie dużej klasy lub zestawu spokrewnionych klas na dwie hierarchie — abstrakcję oraz implementację. Nad obiema można wówczas pracować niezależnie.



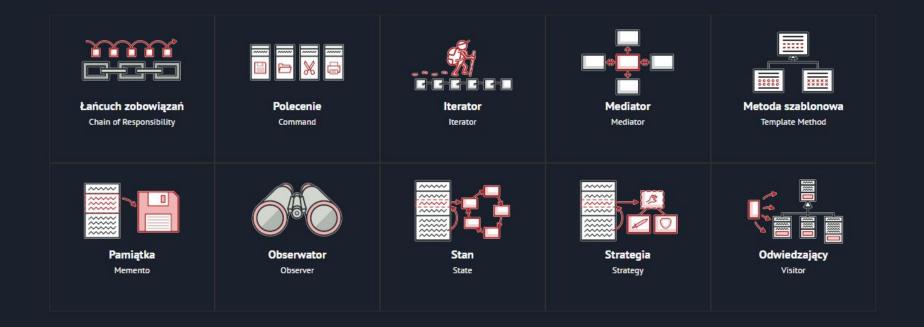
# Kompozyt

**Kompozyt** to strukturalny wzorzec projektowy pozwalający komponować obiekty w struktury drzewiaste, a następnie traktować te struktury jakby były osobnymi obiektami.



#### Wzorce czynnościowe (behawioralne/operacyjne)

**Wzorce czynnościowe (behawioralne lub operacyjne)** - opisują zachowanie i odpowiedzialność obiektów, które współpracują ze sobą.



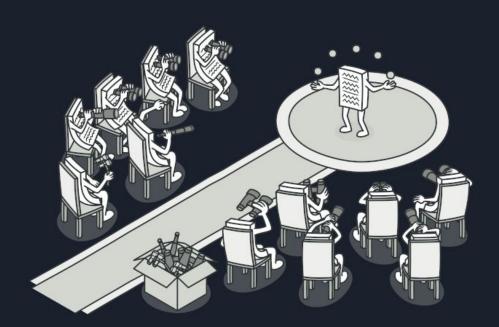
#### Strategia

**Strategia** to behawioralny wzorzec projektowy pozwalający zdefiniować rodzinę algorytmów, umieścić je w osobnych klasach i uczynić obiekty tych klas wymienialnymi.



#### Obserwator

**Obserwator** to czynnościowy (behawioralny) wzorzec projektowy pozwalający zdefiniować mechanizm subskrypcji w celu powiadamiania wielu obiektów o zdarzeniach dziejących się w obserwowanym obiekcie.



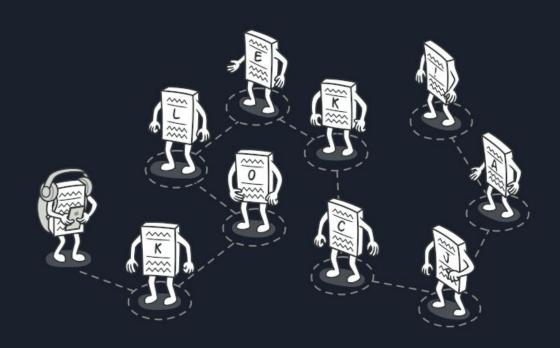
#### Mediator

**Mediator** to behawioralny wzorzec projektowy pozwalający zredukować chaos zależności pomiędzy obiektami. Wzorzec ten ogranicza bezpośrednią komunikację pomiędzy obiektami i zmusza je do współpracy wyłącznie za pośrednictwem obiektu mediatora



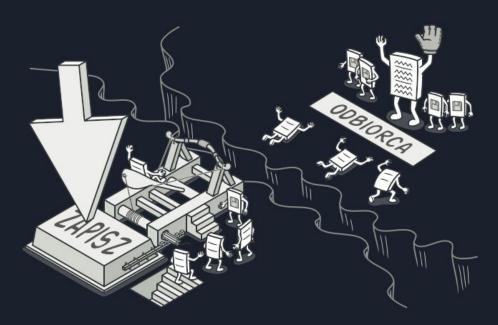
#### Iterator

**Iterator** to behawioralny wzorzec projektowy, pozwalający sekwencyjnie przechodzić od elementu do elementu jakiegoś zbioru bez konieczności eksponowania jego formy (lista, stos, drzewo, itp.).



#### Polecenie

**Polecenie** jest behawioralnym wzorcem projektowym który zmienia żądanie w samodzielny obiekt zawierający wszystkie informacje o tym żądaniu. Taka transformacja pozwala na parametryzowanie metod przy użyciu różnych żądań. Oprócz tego umożliwia opóźnianie lub kolejkowanie wykonywania żądań oraz pozwala na cofanie operacji.



# Źródła

- <a href="https://refactoring.guru/pl/design-patterns">https://refactoring.guru/pl/design-patterns</a>
- <a href="https://pl.wikipedia.org/wiki/Wzorzec projektowy">https://pl.wikipedia.org/wiki/Wzorzec projektowy</a> (informatyka)
- <a href="https://bykowski.pl/wzorce-projektowe">https://bykowski.pl/wzorce-projektowe</a>