



Wzorce Projektowe

Marcin Szupke

Autor: Marcin Szupke

Prawa do korzystania z materiałów posiada Software Development Academy

Wzorzec projektowy

Wzorzec projektowy to rozwiązanie problemu w pewnym kontekście



Wzorzec projektowy

Wzorzec projektowy (ang. design pattern) – uniwersalne, sprawdzone w praktyce rozwiązanie często pojawiających się, powtarzalnych problemów projektowych. Pokazuje powiązania i zależności pomiędzy klasami oraz obiektami i ułatwia tworzenie, modyfikację oraz pielęgnację kodu źródłowego. Jest opisem rozwiązania, a nie jego implementacją. Wzorce projektowe stosowane są w projektach wykorzystujących programowanie obiektowe.





Klasyfikacja wzorców

Klasyfikacja według rodzaju wzorca i opisu tego, co robi. Według tego kryterium wzorce możemy podzielić na trzy rodziny

- Kreacyjne (konstruktywne) – opisują proces tworzenia nowych obiektów. Ich zadaniem jest tworzenie, inicjalizacja oraz konfiguracja obiektów oraz innych typów danych
- Strukturalne – opisują struktury powiązanych ze sobą obiektów
- Czynnościowe (behavioralne, operacyjne) – opisują zachowanie i odpowiedzialność współpracujących ze sobą obiektów



Klasyfikacja wzorców

Klasyfikacja na kategorie według ich zakresów. Kategoryzacja polega na określeniu czy wzorzec dotyczy klas czy obiektów

- Klasowe – opisują statyczne związki między klasami
- Obiektowe – opisują dynamiczne związki między obiektami



Wzorce kreacyjne

- Budowniczy (obiektowy)
- Fabryka abstrakcyjna (obiektowy)
- Metoda wytwarzająca (klasowy)
- Prototyp (obiektowy)
- Singleton (obiektowy)

Autor: Marcin Szupke

Prawa do korzystania z materiałów posiada Software Development Academy



Wzorce strukturalne

- Adapter (klasowy oraz obiektowy)
- Dekorator (obiektowy)
- Fasada (obiektowy)
- Kompozyt (obiektowy)
- Most (obiektowy)
- Pełnomocnik (obiektowy)
- Pyłek (obiektowy)



Wzorce czynnościowo / operacyjne

- Łącuch zobowiązań / odpowiedzialności (obiektowy)
- Polecenie (obiektowy)
- Interpreter (klasowy)
- Iterator (obiektowy)
- Mediator (obiektowy)
- Pamiątka / memento (obiektowy)
- Metoda szablonowa (ang. Template method) (klasowy)
- Obserwator (obiektowy)
- Stan (obiektowy)



Wzorce czynnościowo / operacyjne

- Strategia (obiektowy)
- Odwiedzający / wizytator (ang. Visitor) (obiektowy)
- Zabór zasobu jest inicjalizacją (obiektowy)

Autor: Marcin Szupke

Prawa do korzystania z materiałów posiada Software Development Academy

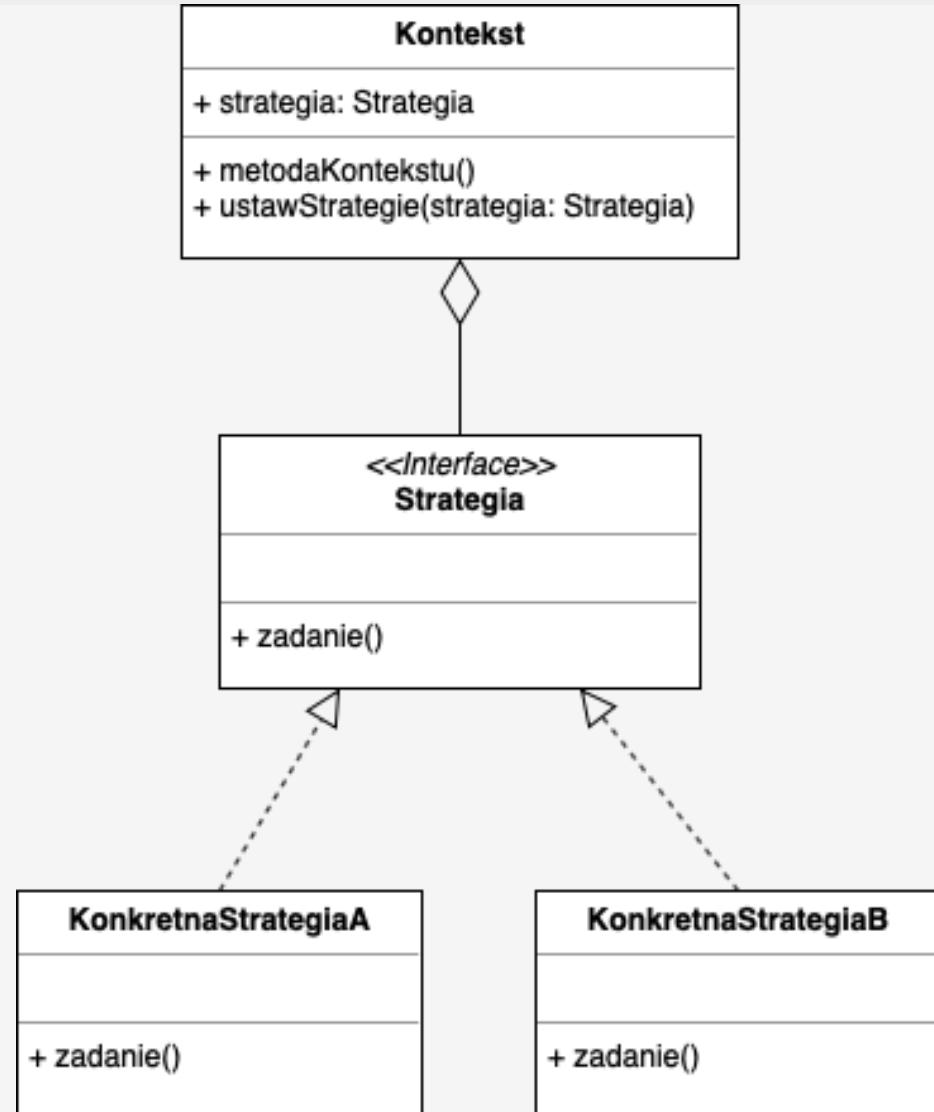
Strategia

Wzorzec Strategia definiuje rodzinę algorytmów, pakuje je jako osobne klasy i powoduje, że są one w pełni wymienne. Zastosowanie tego wzorca pozwala na to, aby zmiany w implementacji algorytmów przetwarzania były całkowicie niezależne od strony klienta, który z nich korzysta





Strategia



Autor: Marcin Szupke

Prawa do korzystania z materiałów posiada Software Development Academy

Obserwator

Wzorzec obserwator definiuje pomiędzy obiektami relację jeden-do-wielu w taki sposób, że kiedy wybrany obiekt zmienia swój stan, to wszystkie jego obiekty zależne zostają o tym powiadomione i automatycznie zaktualizowane

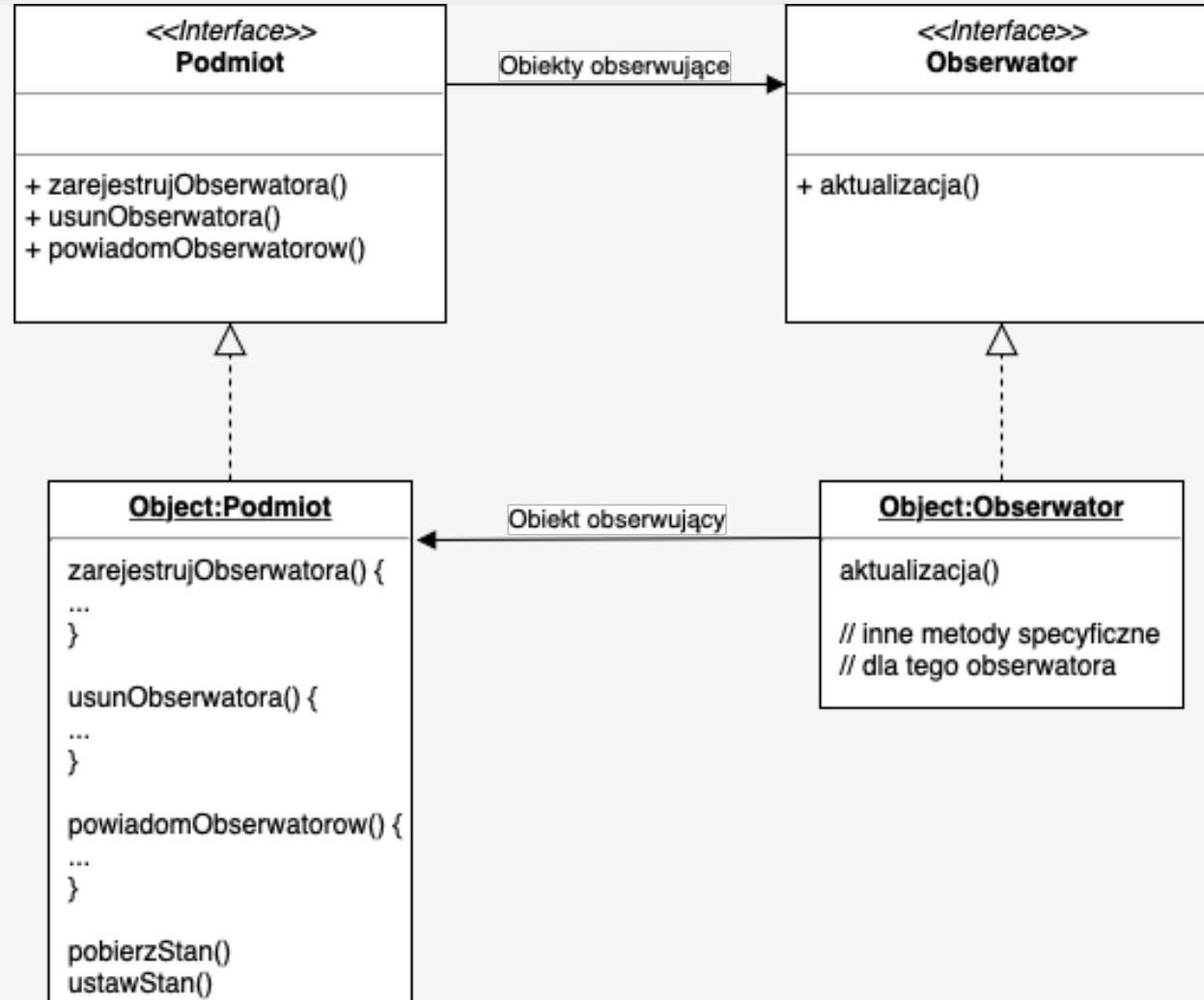


Autor:

Prawa do korzystania z materiałów posiada Software Development Academy



Obserwator



Autor: Marcin Szupke

Prawa do korzystania z materiałów posiada Software Development Academy

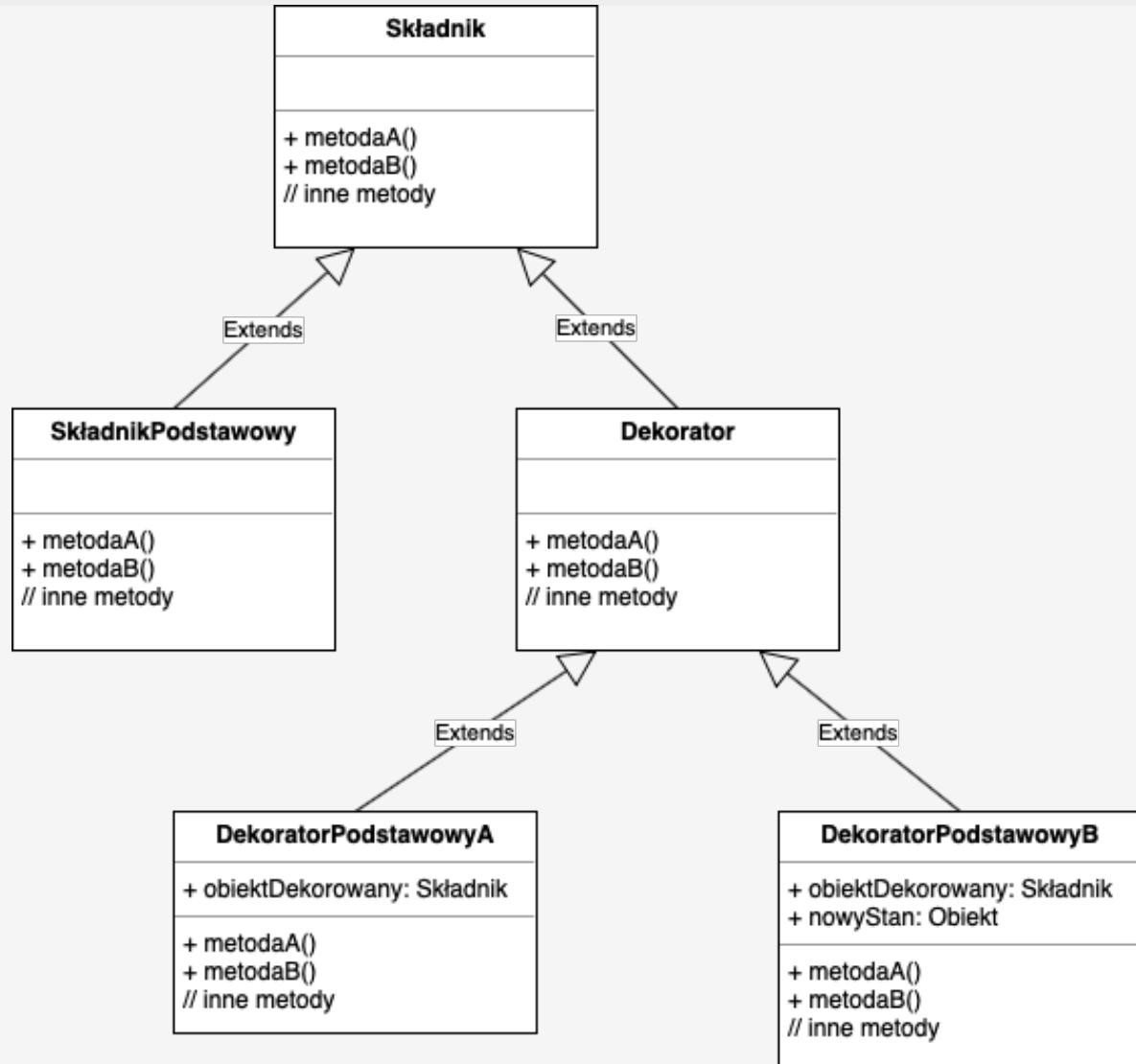
Dekorator

Wzorzec Dekorator pozwala na dynamiczne przydzielanie danemu obiekowi nowych zachowań. Dekoratory dają elastyczność podobną do tej, jaką daje dziedziczenie, oferując jednak w zamian znacznie rozszerzoną funkcjonalność.





Dekorator



Autor: Marcin Szupke

Prawa do korzystania z materiałów posiada Software Development Academy

Metoda Fabrykująca (wytwarzająca)

Wzorzec Metoda Fabrykująca definiuje interfejs pozwalający na tworzenie obiektów, ale pozwala klasom podrzędnych decydować, jakiej klasy obiekt zostanie utworzony. Wzorzec Metoda Fabrykująca przekazuje odpowiedzialność za tworzenie obiektów do klas podrzędnych

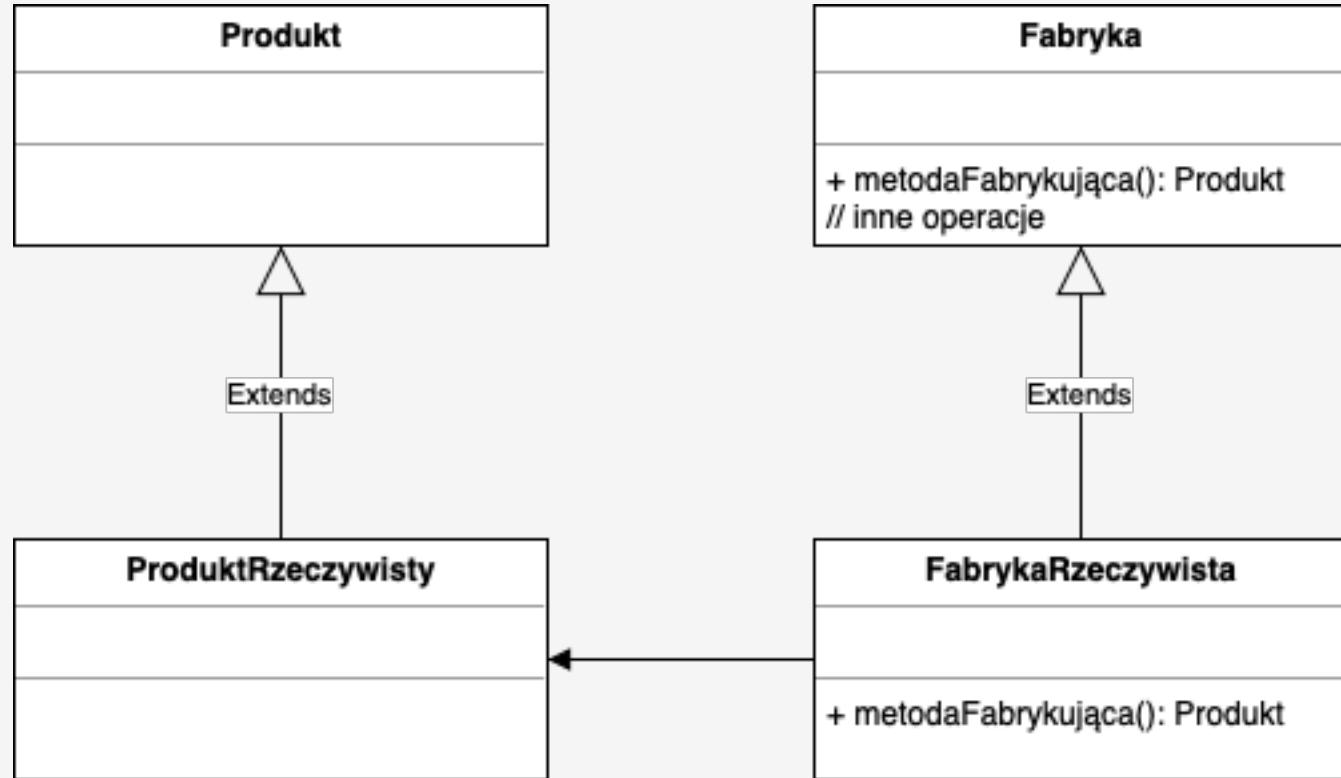


Autor:

Prawa do korzystania z materiałów posiada Software Development Academy



Metoda Fabrykująca (wytwórcza)



Autor:

Prawa do korzystania z materiałów posiada Software Development Academy

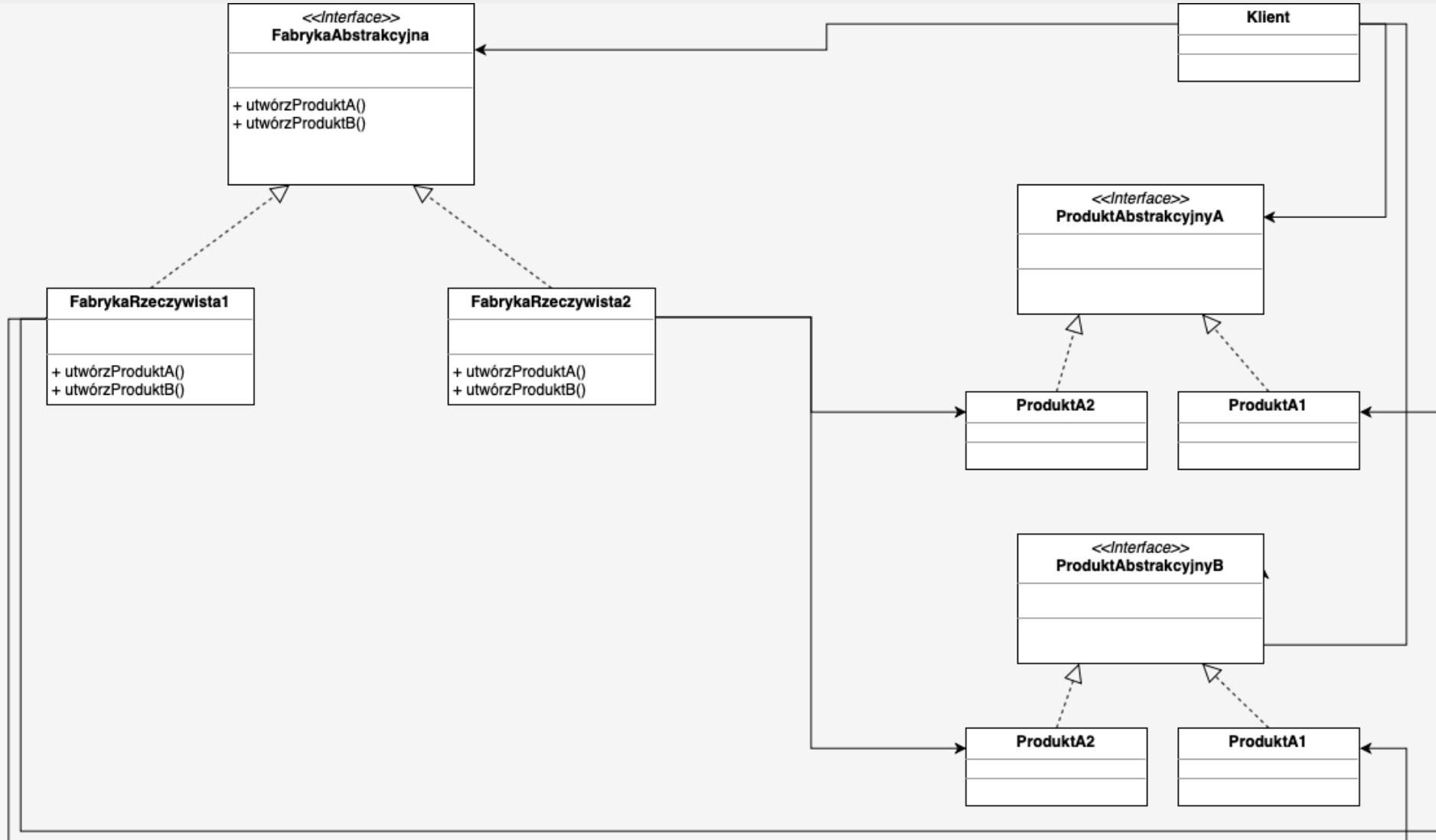
Fabryka abstrakcyjna

Wzorzec Fabryka abstrakcyjna dostarcza interfejs do tworzenia całych rodzin spokrewnionych lub zależnych od siebie obiektów bez konieczności określania ich klas rzeczywistych





Fabryka abstrakcyjna



Autor:

Prawa do korzystania z materiałów posiada Software Development Academy

Singleton

Wzorzec Singleton zapewnia, że dana klasa będzie miała tylko i wyłącznie jedną instancję obiektu i zapewnia globalny punkt dostępu do tej instancji



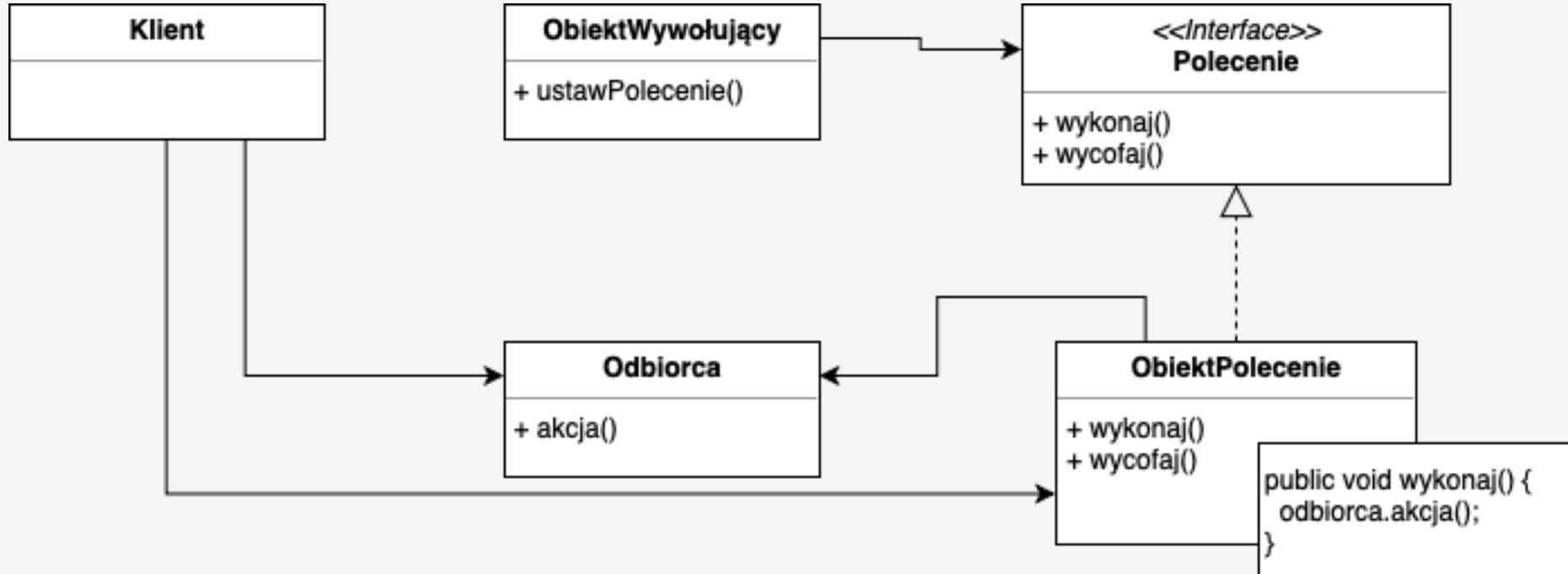
Polecenie

Wzorzec polecenie hermetyzuje żądania w postaci obiektów, co umożliwia parametryzowanie różnych obiektów zróżnicowanymi żądaniami (taki jak na przykład żądania kolejkowania lub rejestracji) oraz obsługiwanie operacji, które można wycofać





Polecenie



Autor:

Prawa do korzystania z materiałów posiada Software Development Academy

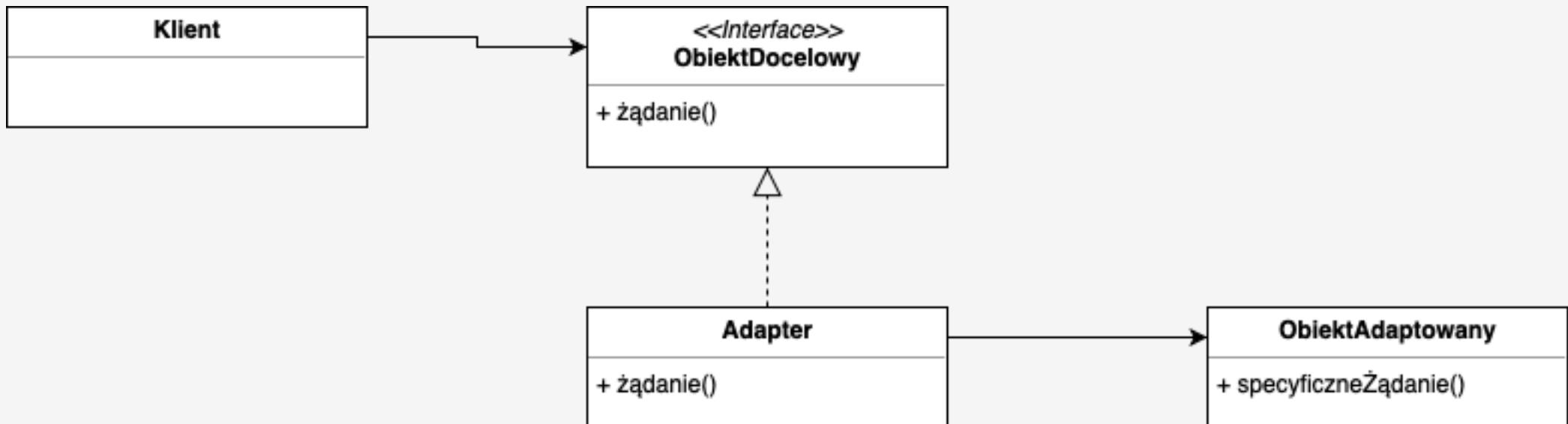
Adapter

Wzorzec Adapter dokonuje konwersji interfejsu danej klasy do postaci innego interfejsu, zgodnego z oczekiwaniami klienta. Adapter pozwala na wzajemną współpracę klas, które ze względu na niekompatybilne interfejsy wcześniej nie mogły ze sobą współpracować





Adapter



Autor:

Prawa do korzystania z materiałów posiada Software Development Academy

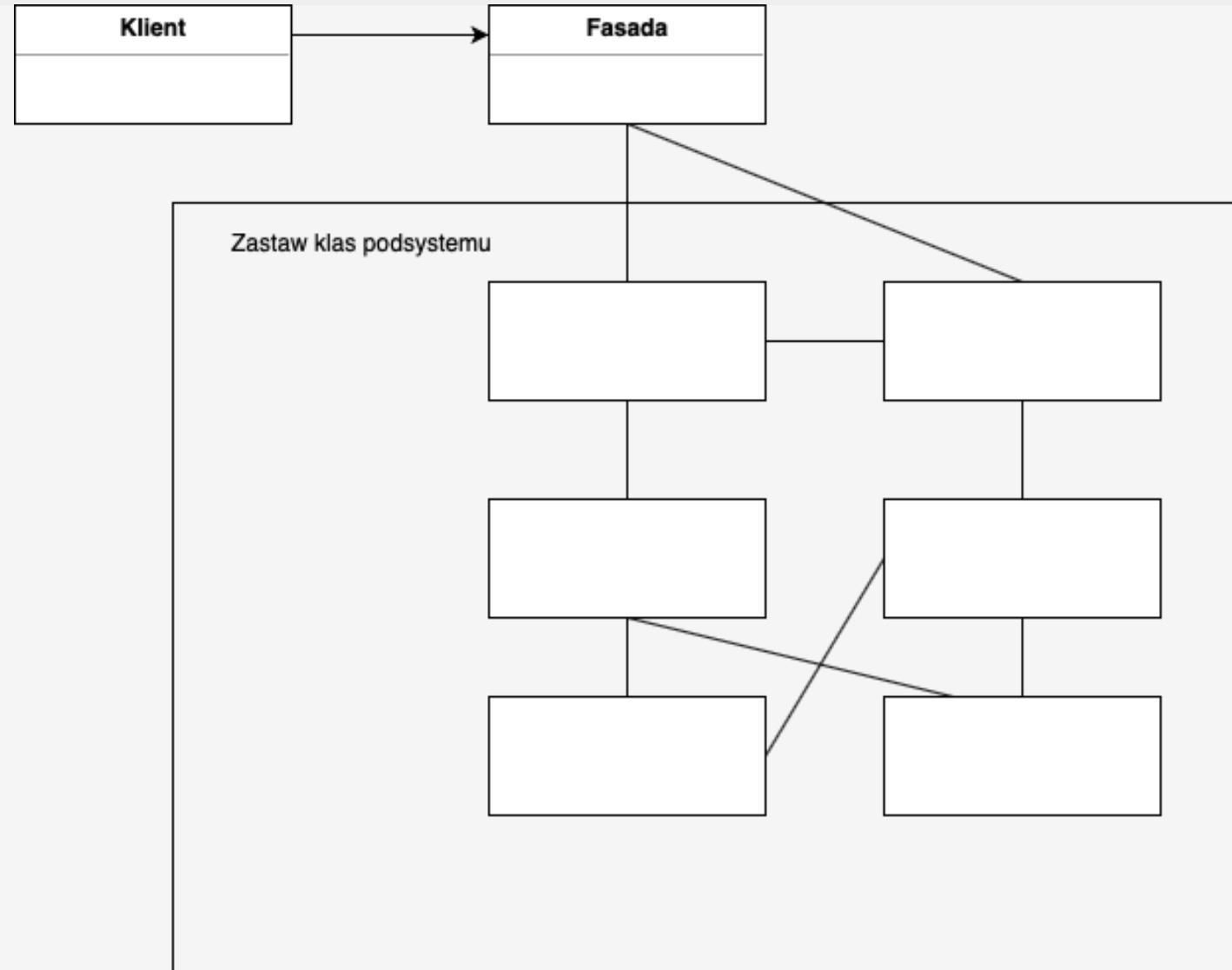
Fasada

Wzorzec Fasada zapewnia jeden, zunifikowany interfejs dla całego zestawu interfejsów określonego podsystemu. Fasada tworzy nowy interfejs wysokiego poziomu, który powoduje, że korzystanie z całego podsystemu staje się zdecydowanie łatwiejsze





Fasada



Autor:

Prawa do korzystania z materiałów posiada Software Development Academy

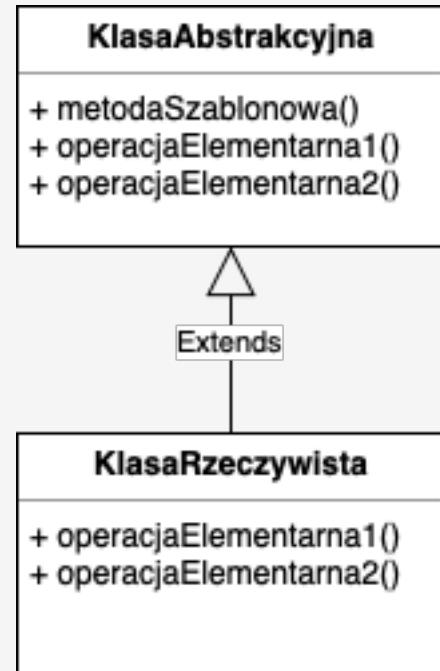
Metoda Szablonowa

Wzorzec Metoda Szablonowa definiuje szkielet danego algorytmu w określonej metodzie, przekazując realizację niektórych jego kroków do klas podrzędnych. Wzorzec ten pozwala klasom podrzędnym na redefiniowanie pewnych kroków algorytmu, ale jednocześnie uniemożliwia zmianę jego struktury





Metoda szablonowa



Autor:

Prawa do korzystania z materiałów posiada Software Development Academy

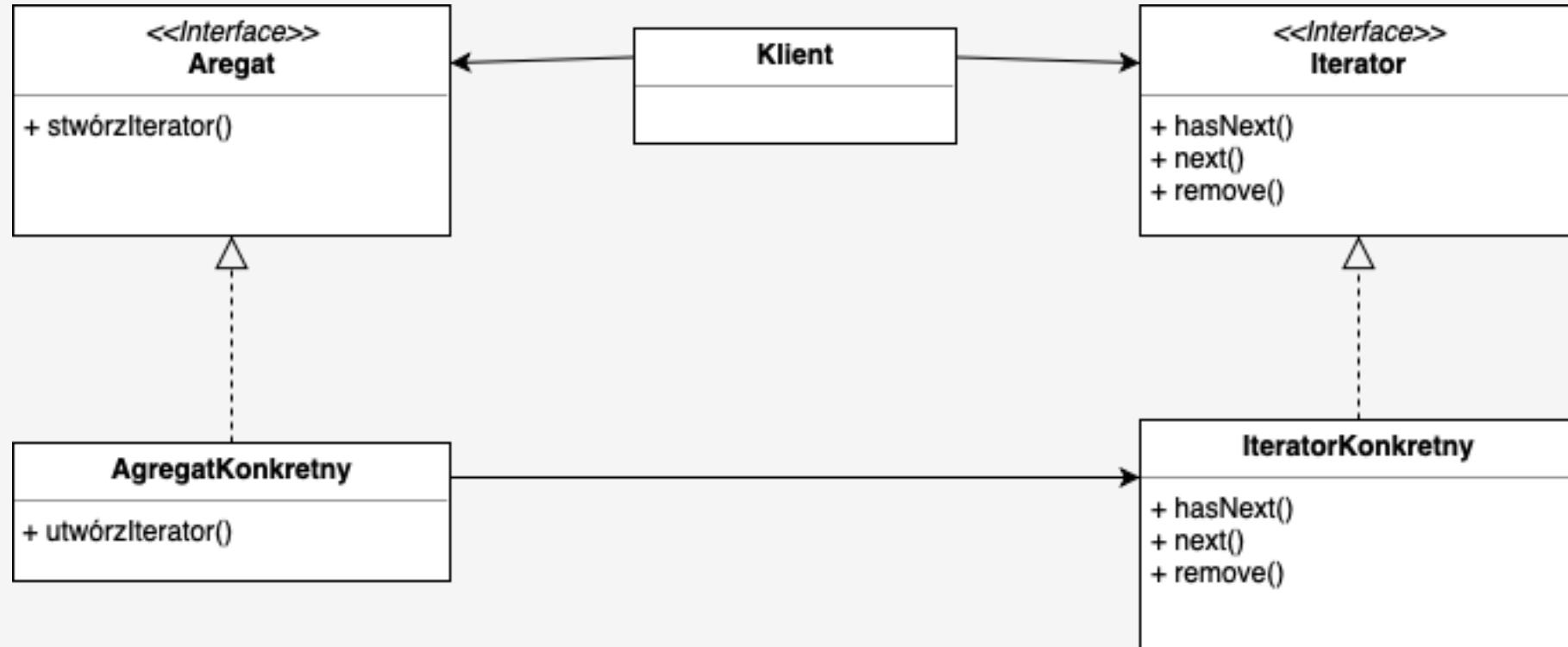
Iterator

Wzorzec Iterator zapewnia metodę dostępu sekwencyjnego do elementów obiektu zagregowanego bez jego reprezentacji wewnętrznej





Iterator



Autor:

Prawa do korzystania z materiałów posiada Software Development Academy

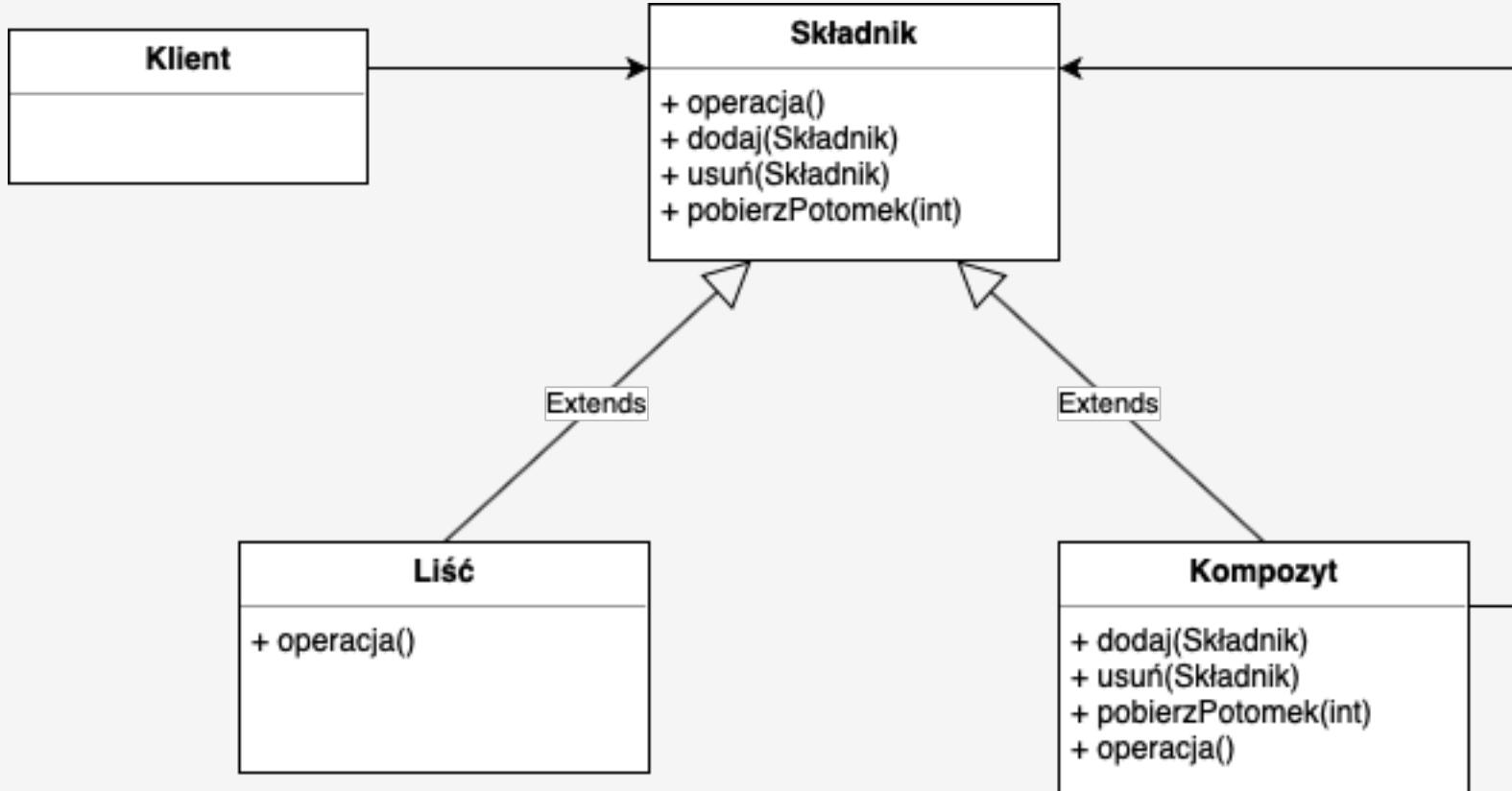
Kompozyt

Wzorzec Kompozyt pozwala łączyć obiekty w struktury drzewiaste, które reprezentują hierarchie część – całość. Jego zastosowanie pozwala klientom jednolicie obsługiwać zarówno pojedyncze obiekty, jak i ich kompozycje





Kompozyt



Autor:

Prawa do korzystania z materiałów posiada Software Development Academy

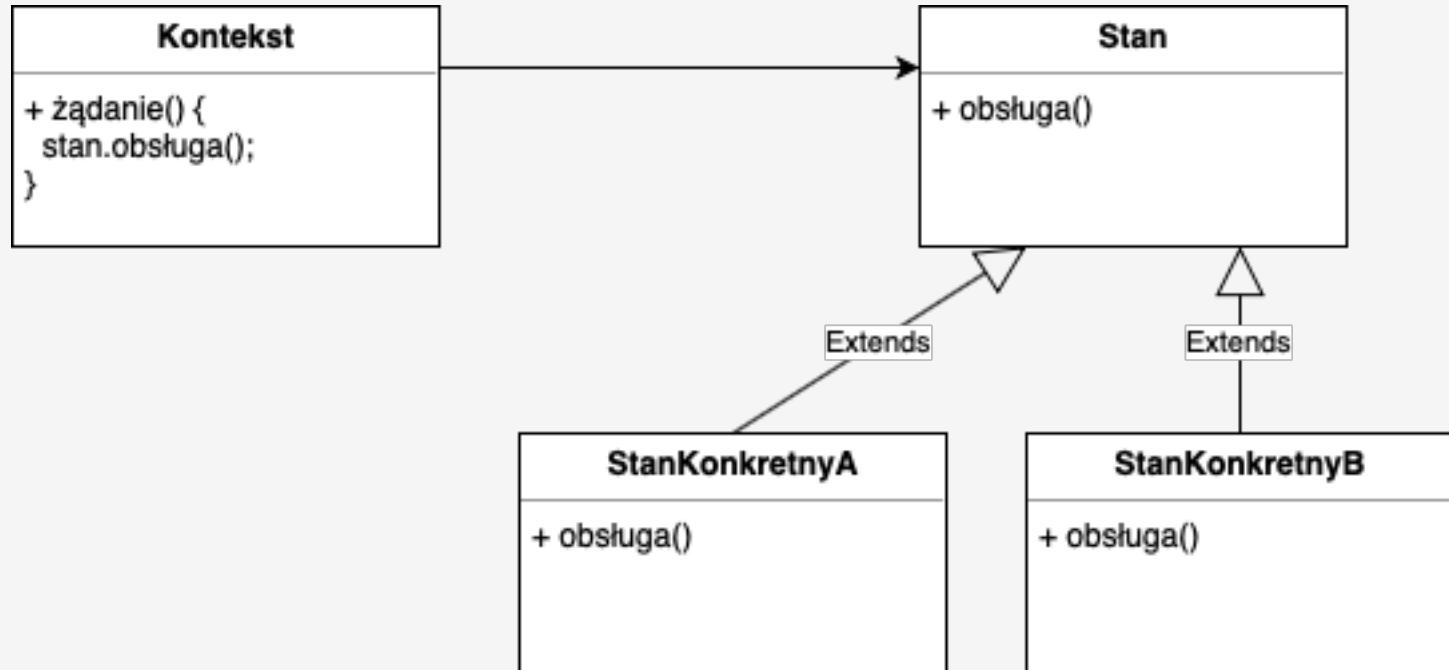
Stan

Wzorzec Stan umożliwia obiektowi zmianę zachowania wraz ze zmianą jego wewnętrznego stanu. Po takiej zmianie funkcjonuje on jak inna klasa





Stan



Autor:

Prawa do korzystania z materiałów posiada Software Development Academy

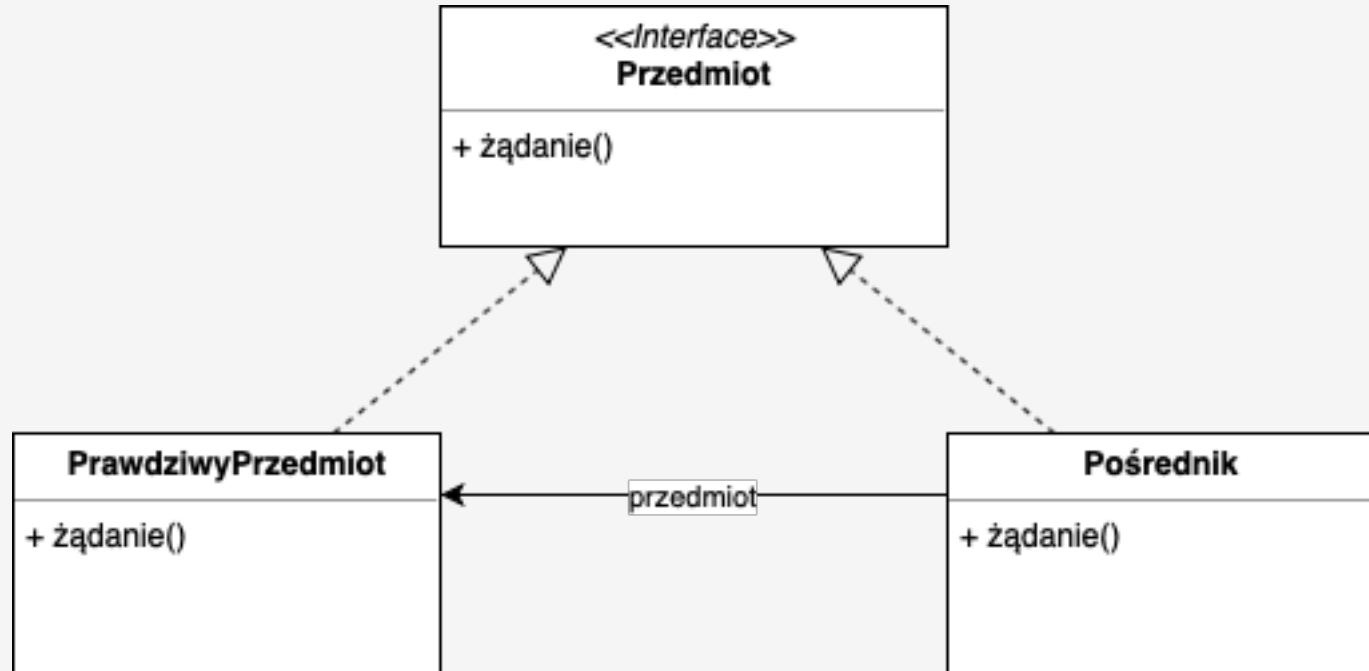
Proxy

Wzorzec Proxy prowadzi do utworzenia obiektu pośredniczącego, który kontroluje dostęp do innego obiektu





Proxy



Autor:

Prawa do korzystania z materiałów posiada Software Development Academy

Most

Wzorzec, którego celem jest oddzielenie abstrakcji od jej implementacji, tak aby mogły się niezależnie od siebie zmieniać.



Budowniczy

Wzorzec, w którym proces tworzenia obiektu podzielony jest na kilka mniejszych etapów, a każdy z nich może być implementowany na wiele sposobów. Dzięki temu możliwe jest tworzenie różnych reprezentacji obiektów w tym samym procesie konstrukcyjnym.



Łańcuch odpowiedzialności

Wzorzec, którego celem jest umożliwienie uniknięcia związania nadawcy i odbiorcy żądania poprzez danie więcej niż jednemu obiekowi szansy obsłużenia tego żądania. Tworzy łańcuch obiektów i przekazuje im kolejno żądanie, aż jakiś obiekt je obsługuje.



Flyweight (pyłek, waga piórkowa)

Wzorzec, którego celem jest efektywna obsługa wielkiej liczby drobnych obiektów poprzez ich współdzielenie.



Interpreter

Wzorzec, którego celem jest definiowanie reprezentacji dla gramatyki zadanego języka, a także interpretera, który wykorzystuje tę reprezentację do interpretowania zdań w danym języku.



Mediator

Wzorzec, którego celem jest kapsułkowanie informacji o tym, jak obiekty współdziałyają. Przyczynia się on do rozluźnienia powiązań między obiektami, gdyż sprawia, że nie odwołują się one do siebie wprost.



Memento (pamiątka)

Wzorzec, którego celem jest zapamiętanie i przekazanie na zewnątrz stanu wewnętrznego obiektu, bez naruszenia kapsułkowania, po to aby mógł być on później przywrócony do zapamiętanego stanu



Prototyp

Wzorzec definiuje rodzaje tworzonych obiektów, używając prototypowego egzemplarza, a także tworzy nowe obiekty, kopując ten prototyp.



Visitor (gość)

Wzorzec, którego celem jest określenie operacji, która ma być wykonana na elementach struktury obiektowej. Umożliwia definiowanie nowej operacji bez modyfikowania klas elementów.

