psc_ TRANSITION TECHNOLOGIES

Wzorce projektowe

Michał Górecki

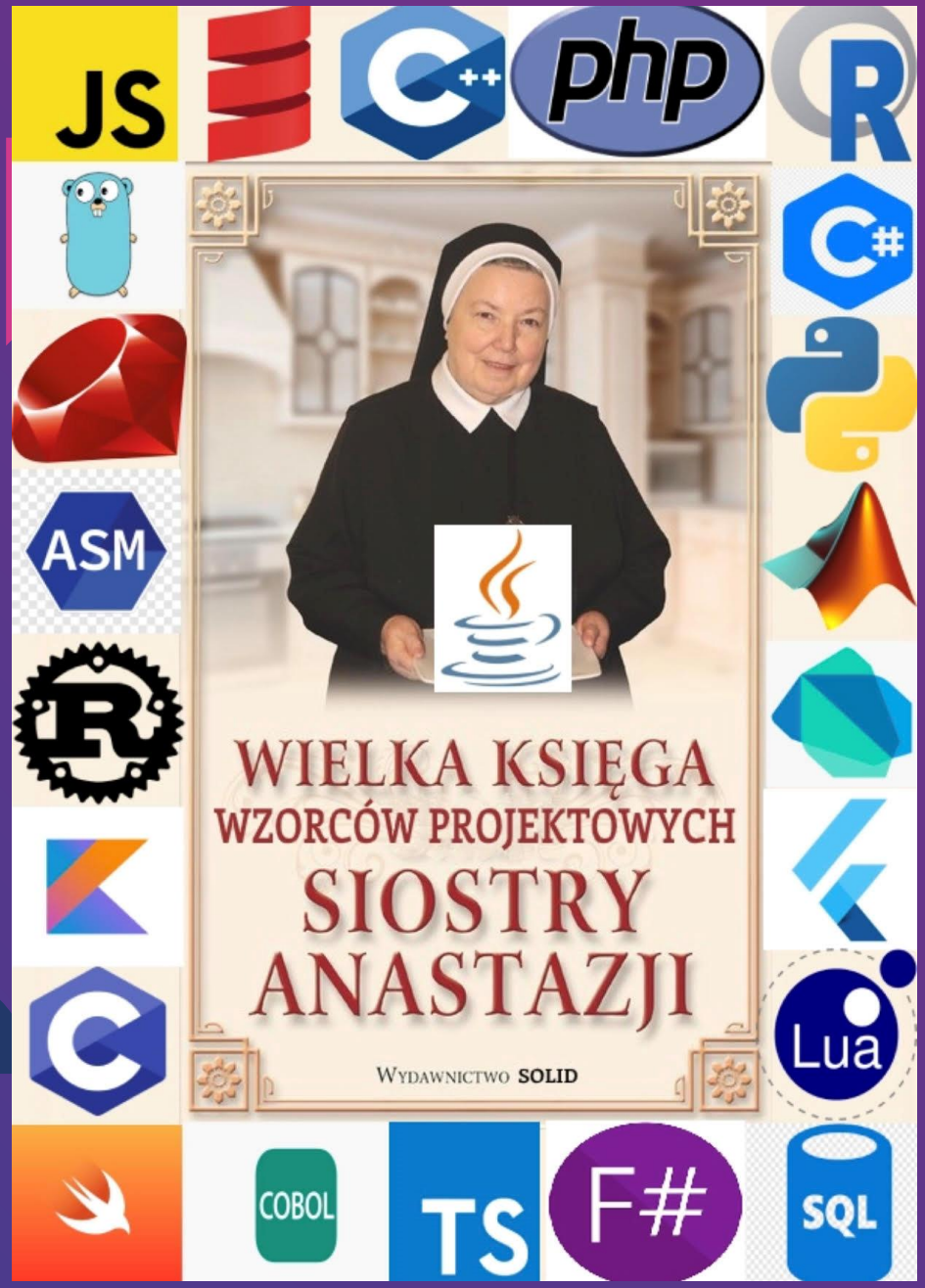
psc_ TRANSITION TECHNOLOGIES

Czym są wzorce projektowe?

> **Wzorzec projektowy** (ang. *design pattern*) – uniwersalne, sprawdzone w praktyce rozwiązanie często pojawiających się, powtarzalnych **problemów projektowych**. (…) ułatwia tworzenie, modyfikację oraz utrzymanie kodu źródłowego. Jest **opisem rozwiązania**, a nie jego implementacją.

psc_ TRANSITION TECHNOLOGIES

Historia pewnej książki

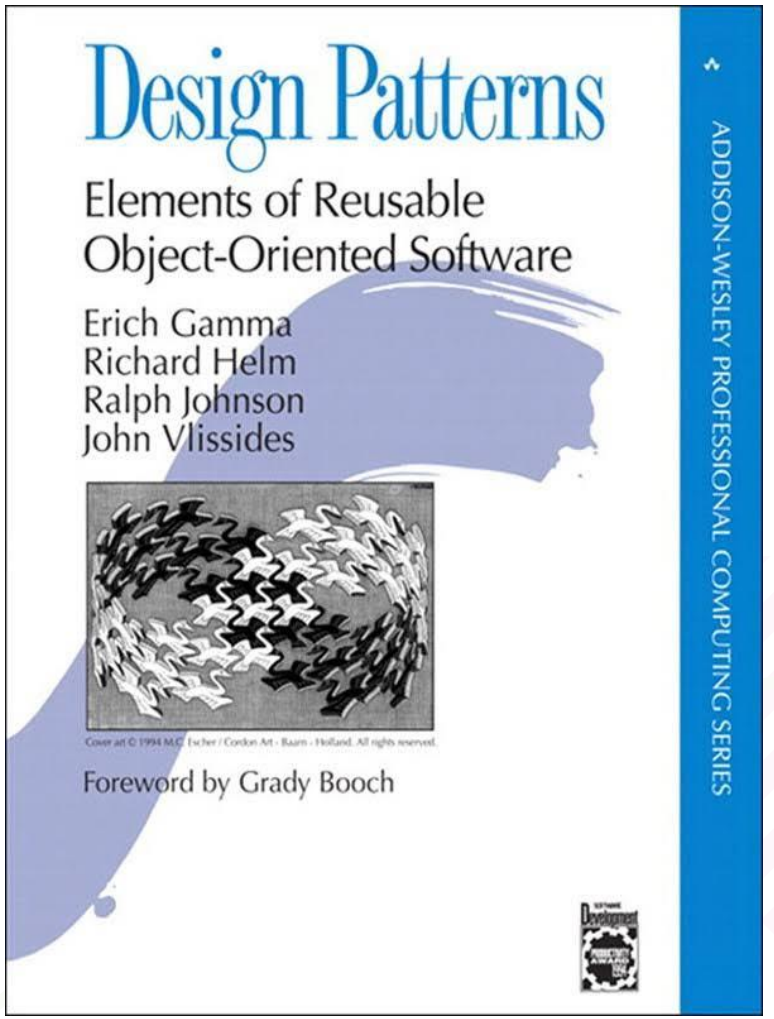


Historia wzorców projektowych

- Termin pochodzi z architektury
- Pomysł rozwinęli Erich Gamma, John Vlissides, Ralph Johnson

  Richard Helm
- W 1994 wydano książkę „*Wzorce projektowe.*

  Elementy oprogramowania obiektowego wielokrotnego użytku”
- 23 wzorce
- Gang of Four



Cel wzorców projektowych

- Opis popularnych problemów
- Wykorzystanie popularnych, przetestowanych rozwiązań
- Uniwersalny dla wszystkich języków
- Wprowadzenie wspólnej terminologii dla programistów

Z czego składa się wzorzec?

- **Nazwa i klasyfikacja**
- **Problem –** scenariusz i intencja
- **Rozwiązanie –** wyjaśnienie, struktura, pseudokod
- **Konsekwencje –** zalety, wady, przykłady użycia

Gang of Four patterns

- Kreacyjne (ang. *Creational*)
- Strukturalne (ang. *Structural*)
- Behawioralne/czynnościowe/operacyjne (ang. *Behavioral*)

Wzorce kreacyjne

- Factory
- Abstract factory
- Builder
- Prototype
- Singleton

Wzorce strukturalne

- Adapter
- Bridge
- Composite
- Decorator
- Facade
- Flyweight
- Proxy

Wzorce behawioralne

- Chain of responsibility
- Command
- Iterator
- Mediator
- Memento
- Observer
- State
- Strategy
- Template method
- Visitor

psc_ TRANSITION TECHNOLOGIES

Inne wzorce

Inne wzorce

- Dependency injection
- Lazy initialization
- MVC

psc_ TRANSITION TECHNOLOGIES

Krytyka wzorców

Krytyka wzorców

- Uzupełniają „niedoskonałości” w językach (np. C++ i Java), które nie występują np. w Lisp
- Mogą być użyte bezmyślnie, bez dostosowania do kontekstu
- Nadużywane przez programistów

psc_ TRANSITION TECHNOLOGIES

Anti-patterns

Anty-wzorce

- *Czasem wzorce projektowe stają się anty-wzorcami*
- *„Golden hammer”* – nabyty raz, używany do wszystkiego
- Powtarzalne złe praktyki
- Zwykle nie są specyficzne dla danego języka

Anty-wzorce: przykłady

- *God class* – klasa która wie i umie wszystko
- Singleton overuse
- Poltergeist
- Copy and paste

psc_ TRANSITION TECHNOLOGIES

Z jakich wzorców korzystałem w projektach?

Repozytorium do ćwiczeń

- https://github.com/michalgorecki/design-patterns-training

psc_ TRANSITION TECHNOLOGIES

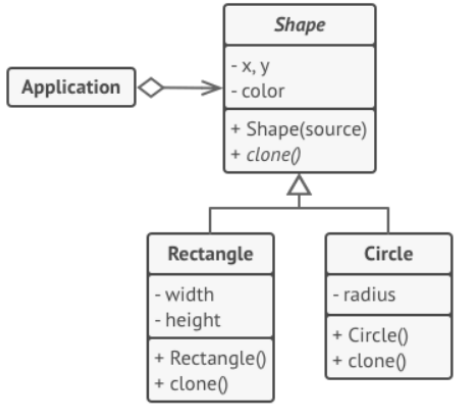
Przykłady wzorców: prototype

Prototype

- **Wzorzec kreacyjny**
- **Problem:** wielokrotne tworzenie identycznych obiektów o takich samych polach
- **Rozwiązanie:** obiekt klonuje sam siebie, zwracając klona
- **Zastosowanie:** 1) gdy bardziej opłacalne jest kopiowanie obiektów

  2) dopiero w runtime wiadomo jakiej konkretnie klasy są obiekty

  3) Chcemy uniknąć hierarchii, w której klasy różnie tworzą obiekty

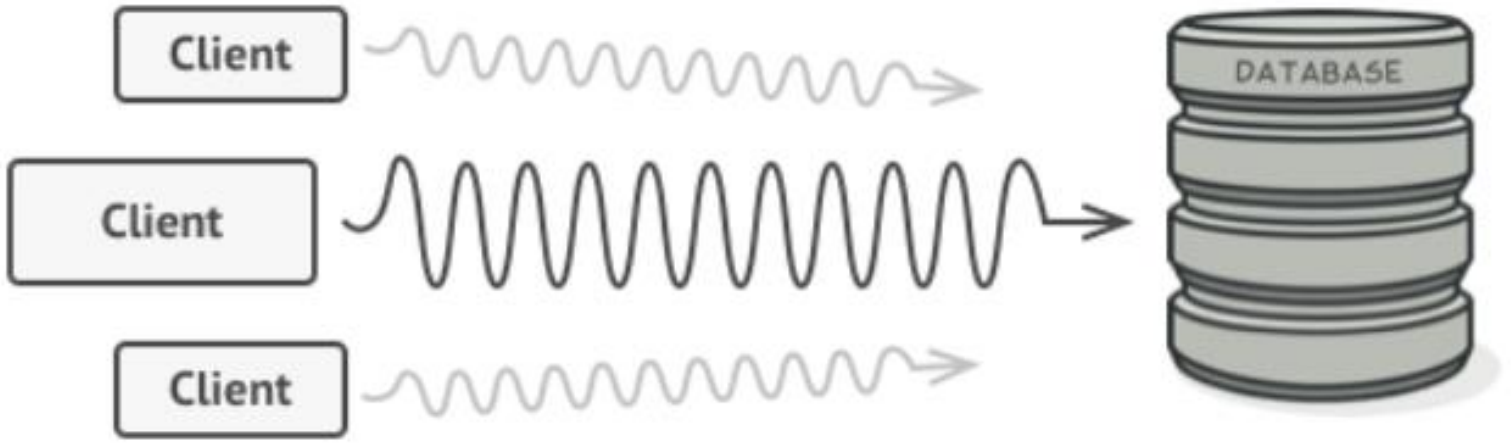


psc_ TRANSITION TECHNOLOGIES

Przykłady wzorców: proxy

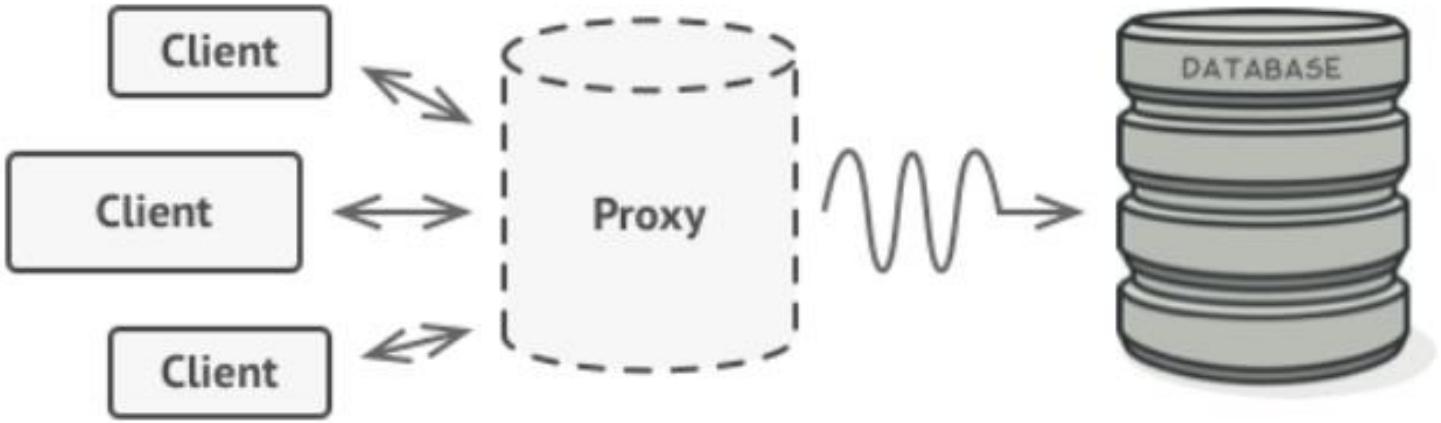
Proxy

- **Wzorzec strukturalny**
- **Problem:** mamy zasób, którego użycie jest kosztowne, np. bazę danych. Chcemy kontrolować dostęp do niej.



Proxy

- **Rozwiązanie:** wprowadzamy pełnomocnika (ang. *Proxy)* między klienta a zasób. Dla klienta interfejs się nie zmienia!
- **Zastosowanie:** chcemy wykonywać dodatkowe operacje „przy okazji”, np. kontrolować dostęp, tworzyć cache, rejestr



IoT@ttpsc.pl
www.ttpsc.com

_Looking forward to build longterm partnership with You •

psc_ TRANSITION TECHNOLOGIES