Wzorce Projektowe w PHP

Karol Kreft, Zawiercie 28.10.2022

Organizacja

- 10:00 - 19:00 - warsztaty
- 14:00 - 15:00 - obiad
- 19:00 - kolacja
- 5-10 minut przerwy co godzinę
- **Mob programming** z zegarem i zmianą co 3 minuty i 30 sekund
- Robimy tyle ile zdążymy
- Feedback

Agenda

- Wstęp
  - Czym są wzorce projektowe?
- Czym są wzorce strukturalne?
- Przykłady wzorców strukturalnych
- Czym są wzorce behavioralne?
- Przykłady wzorców behavioralnych
- Czym są wzorce kreacyjne?
- Przykłady wzorców kreacyjnych

- Co dalej?
- BONUS

Cel warsztatów

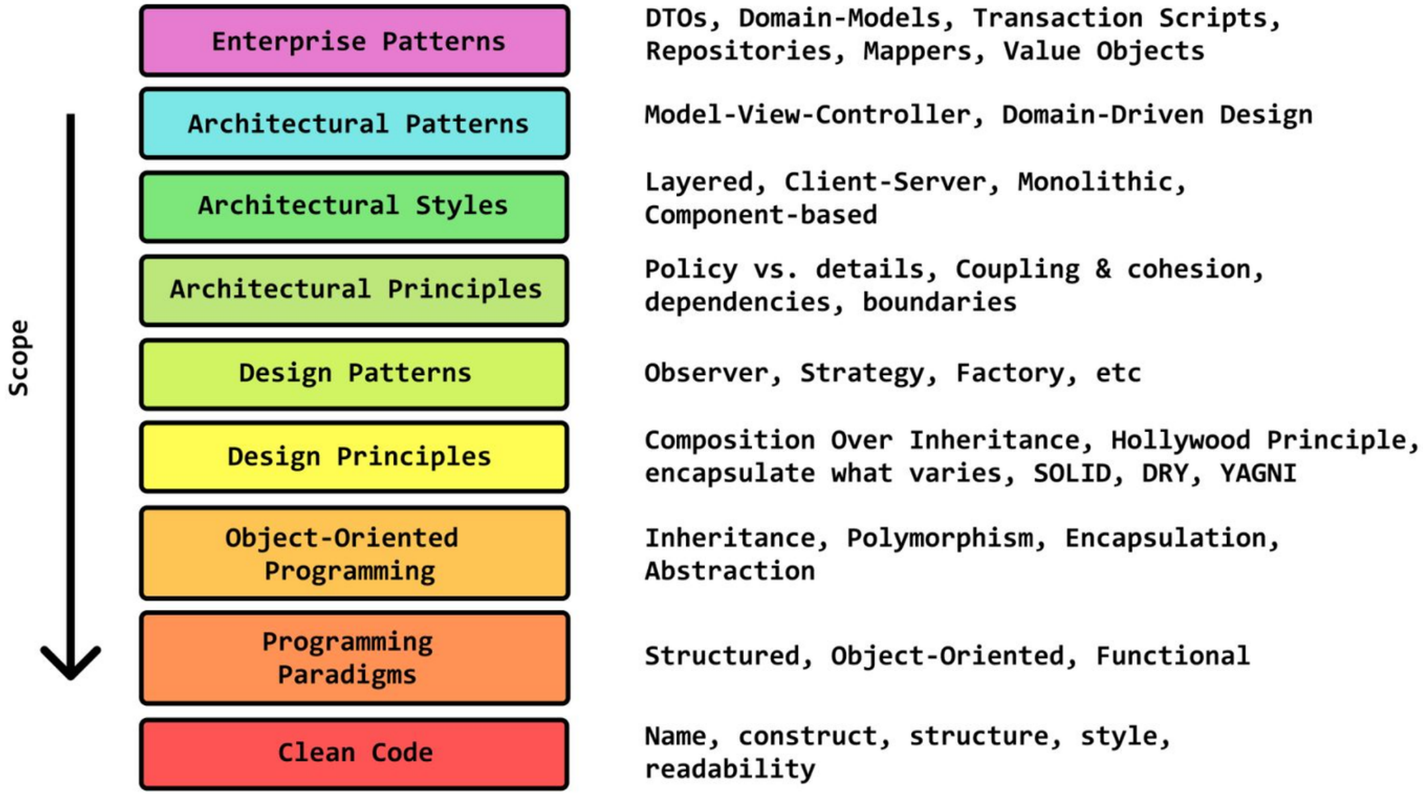
- bliższe poznanie wybranych wzorców projektowych - behawioralnych i strukturalnych
  - zastosowanie wzorców, jaki jest ich cel i jakie problemy rozwiązują
  - użycie w praktyce (praca z kodem)

Cel poboczne

- poznanie wybranych wzorców kreacyjnych
- praca z PHP 8.1
- poznanie techniki mob programmingu w praktyce

The Software Design & Architecture Stack

Khalil Stemmler
@stemmlerjs

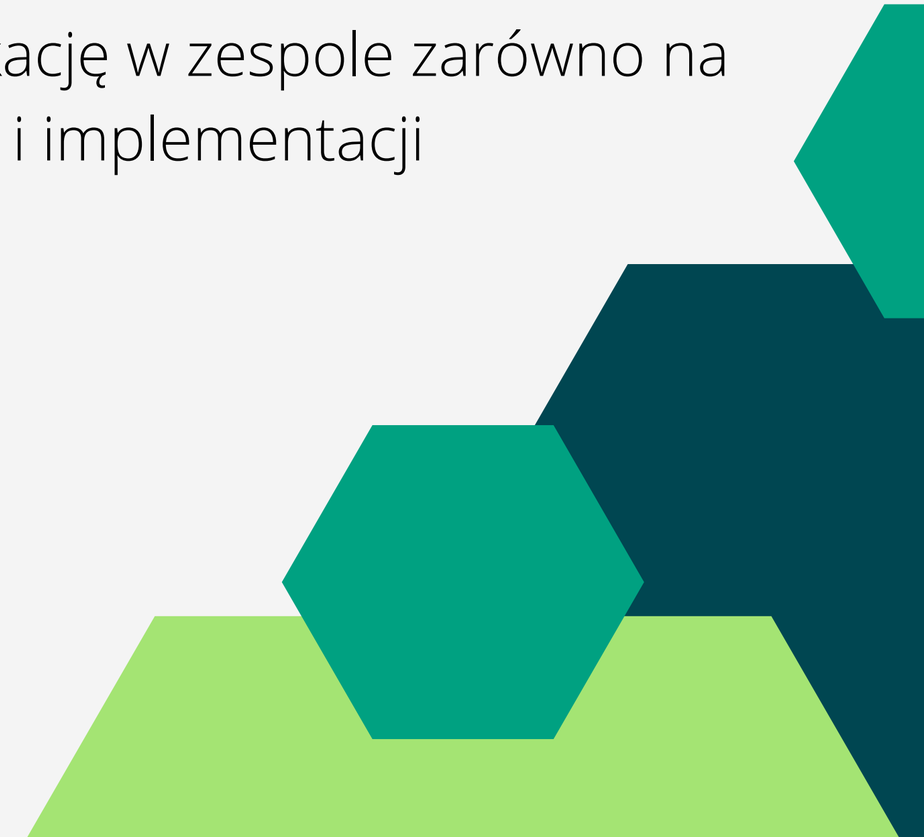


Kilka złotych zasad

- **kompozycja nad dziedziczenie**
  - kompozycja pozwala sterować zależnościami w run-timie, oznacza to większą elastyczność
  - rozwiązaniem problemu z dziedziczeniem nigdy nie jest więcej dziedziczenia
- typujemy abstrakcję zamiast implementacji

Czym są wzorce projektowe?

- Katalog rozwiązań na często spotykane problemy napotykane podczas tworzenia oprogramowania
- Stosowanie wzorców projektowych ułatwia komunikację w zespole zarówno na poziomie planowania i proponowania rozwiązań jak i implementacji
- 3 typy wzorców projektowych
  - Strukturalne
  - Behawioralne
  - Kreacyjne



Czym są wzorce strukturalne?

- Cel: łączenie obiekty w takie struktury by maksymalizować ich elastyczność i efektywność

Dekorator

- Gdy chcemy wykonać coś więcej niż pozwala obecna implementacja
- Obecna implementacja musi pozostać bez zmian
- Nowe rozwiązanie będzie korzystać ze starej implementacji

Kompozyt

- Gdy potrzebujemy wspólnego interfejsu dla grupy obiektów jak dla pojedynczego obiektu z tej grupy
- Dobrze się sprawdza się dla struktur drzewiastych takich jak ACL, drzewa dokumentów, katalogi

Fasada

- Gdy potrzebujemy uproszczonego interfejsu dla biblioteki, innego modułu etc.
- Dzięki uproszczeniu mamy dostęp tylko do niezbędnych operacji z punktu widzenia klienta

Adapter

- Gdy potrzebna jest współpraca pomiędzy niekompatybilnymi interfejsami

https://symfony.com/doc/current/http_client.html#psr-18-and-psr-17

Bridge

- Gdy potrzeba zbudować jasną hierarchię zależności, której elementy można rozwijać niezależnie od siebie
  - Do odseparowania abstrakcji od implementacji

Czym są wzorce behawioralne?

- Cel: Określenie komunikacji pomiędzy obiektami

NullObject

- Gdy potrzeba zadbać o domyślne zachowania dla opcjonalnych zależności

Command

- Gdy żądanie możemy przedstawić jako obiekt i chcemy decydować o tym kiedy powinno zostać zrealizowane
  - zwykle nie natychmiast
  - kolejność jest ważna
  - opcjonalnie procesowanie żądanie może zostać przerwane lub jego efekt cofnięty

Strategia

- Gdy decydujemy o wyborze właściwego algorytmu
  - wybieramy spośród algorytmów o spójnym interfejsie

Obserwator

- Gdy chcemy by zmiana w jednym obiekcie została ogłoszona w kilku innych
  - publish/subscriber

Czym są wzorce kreacyjne?

- Cel: Efektywne tworzenie nowych obiektów i zadbanie o ich niezbędne atrybuty

Co dalej?

- https://refactoring.guru/design-patterns
- https://koddlo.pl/kategoria/wzorce-projektowe/ **Krystian Żądło**
- https://designpatternsphp.readthedocs.io/en/latest

Value Object

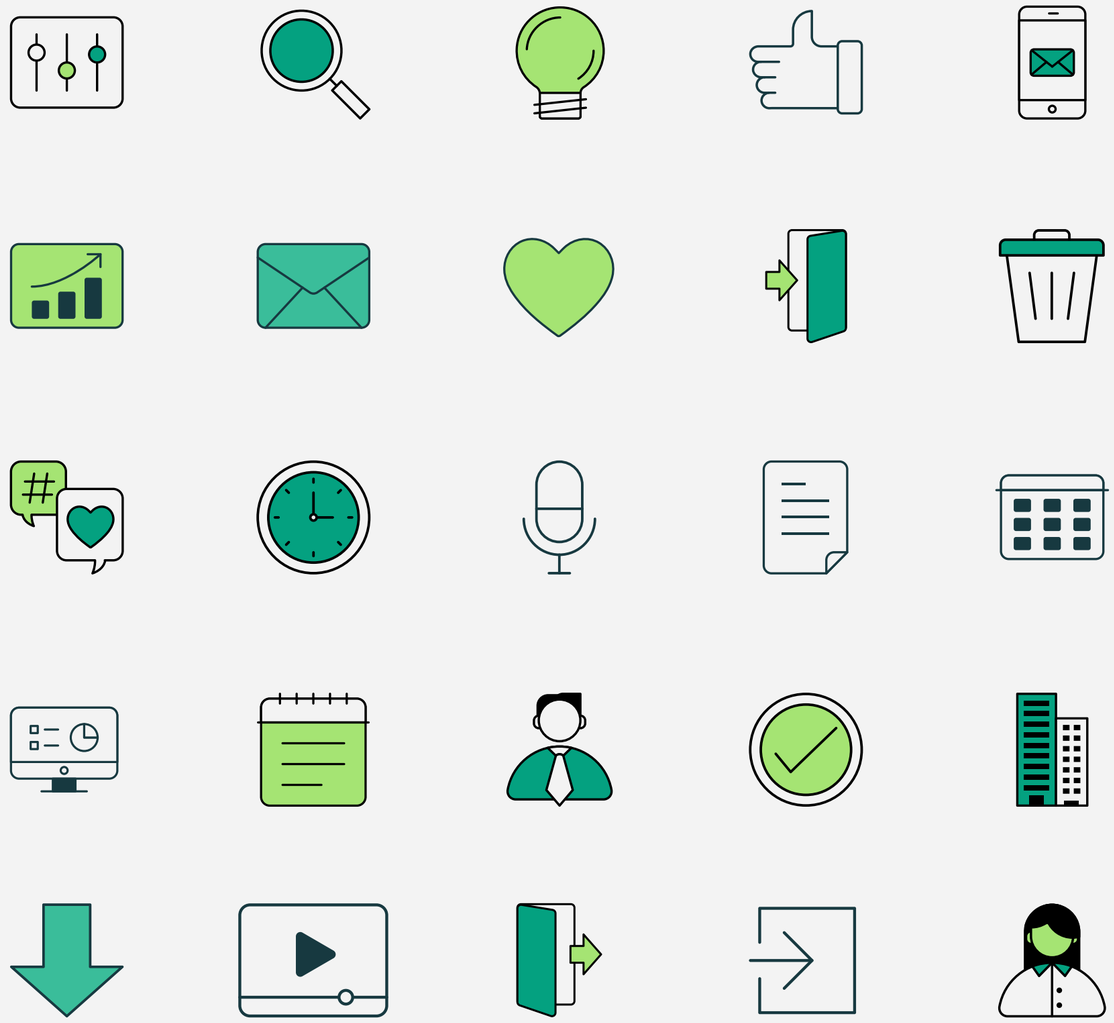
- nie posiada tożsamości
  - przy porównaniu wykorzystuje się jego wartości
- nie podlega edycji przez cały cykl życia
- często stanowi uzupełnienie innego obiektu
  - rzadko występuje samodzielnie


phpcon
POLAND

Napisz tutaj inspirującą deklarację misji.

Dodaj krótkie wytłumaczenie.

Strona zasobów



Użyj tych zasobów w swojej prezentacji Canva. Udanego projektowania! Usuń tę stronę przed rozpoczęciem prezentacji.