

Tytuł szkolenia: Wzorce projektowe i refaktoryzacja kodu w języku PHP

Kod szkolenia: P-PHP-WP-REF

Wprowadzenie

Szkolenie zaczniemy od wprowadzenia teoretycznego do wymienionych poniżej w agendzie wzorców projektowych następnie na podstawie kodu legacy napisanego w czystym starannym PHPie będziemy refaktorować kod przy użyciu wzorców projektowych i TDD (przy użyciu narzędzi phpspec i phpunit).

Adresaci szkolenia

Od uczestników oczekiwana jest znajomość programowania w języku PHP na poziomie średnio zaawansowanym, umiejętność stosowania znaczników HTML i CSS oraz zagadnień związanych z programowaniem obiektowym.

Cel szkolenia

- Poznanie podstawowych typów zagrożeń i sposobów na ich uniknięcie
- Nabycie umiejętności tworzenia bezpiecznych i wydajnych rozwiązań w PHP
- Zrozumienie wzorców projektowych GOF na bazie praktycznych przykładów w języku PHP
- Poznanie prawidłowego kontekstu użycia wzorców, a także konsekwencji ich stosowania
- Nabycie umiejętności poprawiania jakości kodu poprzez refaktoryzację

Szkolenia uzupełniające

- [Programowanie aplikacji internetowych w oparciu o framework Symfony 2](#)
- [Mapowanie obiektowo-relacyjne z Doctrine](#)
- [Mapowanie obiektowo-relacyjne z Propel](#)
- [Wydajność i bezpieczeństwo aplikacji PHP](#)

Czas i forma szkolenia

- 21 godzin (3 dni x 7 godzin), w tym wykłady i warsztaty praktyczne.

Plan szkolenia

1. Wzorce projektowe

- Definicja i pochodzenie
- Elementy składowe
- Klasyfikacja
- Katalogi wzorców
- Konsekwencje stosowania

2. Wzorce kreacyjne

- Fabryka abstrakcyjna (Abstract Factory)
- Budowniczy (Builder)
- Metoda fabrykująca (Factory Method)

3. Wzorce strukturalne

- Adapter (Adapter)
- Most (Bridge)
- Kompozyt (Composite)
- Specyfikacja (Specification)
- Dekorator (Decorator)
- Fasada (Facade)
- Proxy (Proxy)

4. Wzorce behawioralne

- Łańcuch odpowiedzialności (Chain of Responsibility)
- Interpreter (Interpreter)
- Iterator (Iterator)
- Obserwator (Observer)
- Strategia (Strategy)
- Metoda szablonowa (Template Method)
- Gość (Visitor)
- Pusty Obiekt (NullObject)

5. Refaktoryzacja

- Czym jest refaktoryzacja?
- Przekształcenia kodu oparte o testy jednostkowe
- Narzędzia wspomagające refaktoryzację (phpspec, phpunit)
- Typowy kod wymagający refaktoryzacji
- Prosta refaktoryzacja kodu

6. Refaktoryzacja do wzorców

- Zastępowanie konstruktorów metodami fabrykującymi
- Przenoszenie operacji tworzenia obiektów do fabryk
- Hermetyzacja klasy poprzez wprowadzanie fabryki
- Użycie klasy Builder do hermetyzacji obiektów typu Composite
- Komponowanie metod
- Dodawanie nowych zachowań oraz złączanie źródeł danych przy użyciu wzorca Decorator w Repozytoriach
- Zastępowanie wyrażeń warunkowych przy użyciu wzorców Strategy
- Wprowadzanie wzorca Composite jako warstwa ACL dla repozytoriów (Anti Corruption Layer)
- Pokazanie wzorca Specification Pattern
- Tworzenie metody szablonowej
- Likwidacja bezpośrednich powiadomień przy użyciu wzorca Observer
- Ujednolicanie interfejsów przy użyciu Adapterów i Bridge
- Wprowadzanie Responderów dla CommandHandlerów
- Użycie obiektów pustych (Null Object) na podstawie domyślnych Responderów