SZKOLENIE ZAAWANSOWANE

Wzorce i dobre praktyki na platformie Java

UGC

Czas trwania: 5 dni (40h)

Poprawne projektowanie rozwiązań dla zaawansowanych aplikacji Java w oparciu o zaktualizowane wzorce projektowe z kilku szablonów

Cele szkolenia

- Zdobycie umiejętności projektowania zaawansowanych aplikacji Java w notacji UML, przy użyciu najlepszych, sprawdzonych i wydajnych rozwiązań opisanych w szablonach wzorców projektowych GOF, Core J2EE (z uwzględnieniem zmian w specyfikacji JEE oraz alternatyw w Spring), DDD i nowoczesnych wzorców w skalowalnej architekturze
- Przegląd i przećwiczenie wybranych rozwiązań z wielu katalogów wzorców z uwzględnieniem przydatności projektowej, dydaktycznej i możliwości użycia gotowych rozwiązań
- Obniżenie kosztów utrzymania tworzonego oprogramowania
- Nauka myślenia kategoriami gotowych rozwiązań z perspektywy parametrów systemowych takich jak elastyczność, wydajność, niezawodność, utrzymanie, skalowalność i innych
- Nabranie wprawy w posługiwaniu się zdobytą wiedzą oraz przekucie jej w umiejętność tworzenia dobrych rozwiązań i wykrywania zagrożeń
- Nauka poprawnego dobierania rozwiązań i opisywania ich w UML, a nie nauka programowania ze względu na szerokie spektrum dostępnych na rynku technologii
- Szybkie wdrożenie w tematykę JEE i Spring osób niezaznajomionych ze specyfikacją oraz osiągnięcie wyższej jakości rozwiązań przez osoby już pracujące z tymi technologiami
- Omówienie i ćwiczenie wybranych wzorców GOF jako łatwiejsze wprowadzenie w tematykę zaawansowanych wzorców

Zalety

- Szkolenie przedstawia wzorce dla zaawansowanych nowoczesnych aplikacji Java, przestrzegając przed przestarzałymi elementami szablonów, ucząc modelowania z użyciem wzorców oraz wskazując, gdzie wyręczą nas gotowe rozwiązania
- Praktyka przed teorią - wszystkie szkolenia technologiczne prowadzone są w formie warsztatowej. Konieczna teoria jest wyjaśniana na przykładzie praktycznych zadań
- Konkretne umiejętności - w ramach każdego szkolenia rozwijamy praktyczne umiejętności związane z daną technologią i tematyką
- Nauka z praktykami - wszyscy trenerzy na co dzień pracują w projektach, gwarantuje to dostęp do eksperckiej wiedzy i praktycznego know-how

Dla kogo?

- Programiści i projektanci chcący tworzyć bardziej niezawodne, elastyczne i wydajne oprogramowanie Java

Wymagania

- Znajomość Java w ramach projektowania lub programowania

Program

1. Modelowanie projektu w UML - wybrane diagramy
   a. Diagram klas
      - Klasa i jej elementy
      - Klasy i metody abstrakcyjne
      - Interfejs
      - Relacje
   b. Diagram sekwencji
      - Linia życia
      - Rodzaje komunikatów
      - Bloki złożone: alt, break, loop, par
2. Podstawy projektowania obiektowego i wprowadzenie do wzorców
   a. Enkapsulacja
   b. High Cohesion
   c. Loose Coupling
   d. Command-Query Separation
   e. Wprowadzenie do wzorców
   f. GRASP
   g. S.O.L.I.D
3. Wybrane wzorce GoF
   a. Wzorce konstrukcyjne
   b. Wzorce strukturalne
   c. Wzorce behawioralne
4. Wprowadzenie do JEE i alternatyw w Spring
   a. Modele aplikacji: Web-centric, Application-centric i Enterprise
   b. Wstęp do podstawowych technologii JEE
   c. Przedstawienie komponentów JEE i kontenerów
   d. Architektura komponentowa i wielowarstwowa
5. Wzorce Core J2EE w nowoczesnej Javie
   a. Wzorce warstwy prezentacji
   b. Wzorce warstwy biznesowej
   c. Wzorce warstwy integracji
6. DDD kontra Core J2EE
   a. Podstawowe różnice między szablonami
   b. Przegląd poziomu strategicznego
      - Ubiquitous Language
      - Bounded Context
      - Context Map i relacje między kontekstami
      - ACL a Business Delegate
   c. Przegląd poziomu taktycznego z porównaniem
      - Business Object a Dane w DDD (Entity, Value Object, Aggregate)
      - Application Service a Service
      - DAO a Repository
      - Transfer Object a Domain Event
      - Factory Method i Builder (GoF) a Factory
      - Wzorce taktyczne a Layered Architecture
   d. Współpraca kontekstów
   e. Zakresy publikacji zdarzeń
   f. Aspekty technologiczne
   g. Co jeszcze warto wiedzieć o Domain-driven design
7. Microservices i technologie
   a. Wzorce infrastruktury: Cloud, Load Balancer, Clustering
   b. SOA i ESB jako poprzednicy Microservices
   c. Microservices a SOA
   d. Microservices a Monolith
   e. Client-side a Server-side service discovery
   f. Zalety i problemy Microservices
   g. On premise, IaaS, CaaS, PaaS, FaaS, SaaS
   h. Jak tworzyć architekturę Microservices

i. Polyglot Persistence
j. CQRS i Event Sourcing jako wsparcie Microservices
k. Hexagonal Architecture (Ports And Adapters)
l. Słów kilka o szablonach: Microservices Patterns, POSA, PEAA, Core J2EE, DDD, EIP
