Kod szkolenia: **J/DDD**

Tytuł szkolenia: **Domain Driven Design (DDD) w procesie tworzenia aplikacji na przykładzie rozwiązań JEE**

Dni: 3

Opis:

**Adresaci szkolenia**

Szkolenie adresowane jest do programistów, projektantów oraz architektów systemów JEE, pragnących poznać metody projektowania oraz implementacji rozwiązań z obszaru aplikacji JEE z wykorzystaniem podejścia DDD.

**Cel szkolenia**

Celem szkolenia jest zapoznanie uczestnika z alternatywnym sposobem tworzenia aplikacji JEE w stosunku do podejścia najczęściej spotykanego (anemiczny model danych)

Kluczowe zagadnienia:

- Architektura oraz warstwowość aplikacji JEE
- Co to jest DDD i czy rzeczywiście warto z tego podejścia skorzystać
- DDD a anemiczny model danych – zalety oraz wady każdego z nich
- Metodologie prowadzenia projektu i ich wpływ na końcowy kształt systemu
- Persystencja modelu danych oraz potencjalny problem z odwzorowaniem ValueObject w bazie danych

**Mocne strony szkolenia**

Szkolenie nie skupia się tylko i wyłącznie na zaprezentowaniu koncepcji związanych z DDD. Jednym z ważniejszych jego elementów jest zaprezentowanie DDD w porównaniu z najczęściej spotykanym podejściem przy implementacji modelu danych tj. implementacją tzw. anemicznego modelu danych. Co więcej, obecnie coraz więcej projektów jest realizowanych w podejściu zwinnym (Agile), warto więc przybliżyć nieco powiązania pomiędzy wyborem metodologii prowadzenia projektu a faktyczną architekturą jaka w tym procesie powstaje (zostanie uwzględnione również podejście z wykorzystaniem tzw. metodologii „ciężkich"). Generalnie, nie istnieje jedno idealne rozwiązanie. Uczestnictwo w niniejszym szkoleniu pozwala jednak z zupełnie innej perspektywy spojrzeć na proces projektowania i budowy rozwiązań JEE jak również zdobyć wiedzę, gdzie i kiedy dane rozwiązanie jest optymalne, a kiedy warto pomyśleć o czymś innym. Przykładowa aplikacja budowana będzie w oparciu o JSF 2, EJB 3 oraz JPA (Hibernate). Na życzenie istnieje możliwość przeprowadzenia

szkolenia z wykorzystaniem innego zestawu technologii.

**Wymagania:**

Od uczestników szkolenia wymagana jest umiejętność programowania w języku Java (do poznania na kursie J/JP), znajomość podstawowych zagadnień związanych z modelowaniem obiektowym (do poznania na kursie UML/BASE) oraz podstawowych koncepcji związanych z programowaniem aplikacji JEE (do poznania na kursie J/EE6), w szczególności znajomości JSF 2, EJB 3 oraz JPA (Hibernate).

**Parametry szkolenia**

3*8 godzin (3*7 godzin netto) wykładów i warsztatów.

Wielkość grupy: maks. 8-10 osób.

## Program szkolenia:

1. Architektura aplikacji JEE – przegląd najważniejszych koncepcji i założeń
    I. Warstwowość aplikacji JEE w różnym ujęciu
        i. Architektura trójwarstwowa (klient, aplikacja, baza danych)
        ii. Warstwowość w aplikacjach JEE – architektura aplikacji (model danych, warstwa prezentacji, warstwa logiki, warstwa dostępu do bazy danych)
    II. Anemiczny model danych
    III. Alternatywa do anemicznego modelu danych – Domain Driven Design
    IV. Obiektowość a model danych
        i. Low Coupling
        ii. Cohesion
    V. Metodologia prowadzania projektu i jej wpływ na proces budowy rozwiązania
        i. Ciężkie metodologie np. Prince2 i podejście „ Waterfall ”
        ii. Metodologie zwinnej tj. „ Agile ”
2. Który model dominuje w praktyce ?
    I. Zalety i wady anemicznego modelu danych
    II. Zalety i wady DDD
    III. Wybór modelu danych oraz kluczowych założeń dotyczących architektury w praktyce
        i. Czy tylko kwestie techniczne są istotne ?
        ii. Na co zwracać uwagę ?
3. DDD - wprowadzenie
    I. Inny sposób myślenia, technologie to rzecz wtórna
    II. Najważniejsza pojęcia
        i. Ubiquitous Language
        ii. Entities
        iii. Value Objects
        iv. Agregaty

v. Repozytoria
vi. Domain Services
vii. Fabryki
viii. Bounded Context

III. Rozwój aplikacji w modelu DDD
i. Faza tworzenia aplikacji
A. Różnice w podejściu do fazy modelowania, budowy rozwiązania w zależności od przyjętych założeń architektonicznych oraz metodologii prowadzenia projektu
ii. Faza testowania aplikacji
A. DDD a Test Driven Design (TDD)
B. DDD a Behavior Driven Design (BDD)
C. Faza testowania w kontekście wybranej architektury rozwiązania oraz metodologii prowadzenia projektu
iii. Faza utrzymania i rozwoju
A. Korzyści z wykorzystania DDD od POCZĄTKU projektu
a. Zależność fazy utrzymania i rozwoju od efektów faz poprzednich
b. DDD a testowalność
c. DDD a rozszerzalność (np. Open-Close Principle)
d. Zgodność implementacji z oczekiwaniami klienta

4. Implementacja przykładowej aplikacji w w podejściu DDD
I. Implementacja modelu danych z wykorzystaniem Entities, Value Objects oraz Agregatów
i. Różnica między Entity a ValueObject
ii. Odwzorowanie Entity oraz ValueObject na poziomie bazy danych
iii. Wykorzystanie Agregatów
II. Repozytoria w kontekście komunikacji z bazą danych
i. JPA a repozytoria
ii. Dependency Injection
III. Wdrożenie warstwy logiki
i. Miejsce i rola warstwy usług w kontekście faktu, iż logika jest implementowana na poziomie obiektów biznesowych/domenowych (np. Application/Domain/Infrastructure Services)
ii. Bezstanowość usług
IV. Implementacja warstwy interfejsu oraz złożenie projektu w całość
5. DDD – wsparcie poprzez dostępne wzorce projektowe
I. Builder
II. State Machine
III. Chain of Responsibility
IV. Supple Design
6. Podsumowanie