

Kod szkolenia: **CDDD**

Tytuł szkolenia: **Projektowanie złożonych modeli domen z wykorzystaniem Domain Driven Design**

Dni: 3

## Opis:

### Adresaci Szkolenia:

Szkolenie skierowane jest do programistów, projektantów i architektów, którzy pragną poznać praktycznie i przećwiczyć projektowanie złożonych modeli domen z wykorzystaniem Domain Driven Design.

### Cel szkolenia:

Cel szkolenia obejmuje poznanie i opanowanie następujących zagadnień związanych z DDD:

- Techniki modelowania i pracy w zespole DDD
- Techniki i wzorce taktyczne DDD
- Techniki i wzorce strategiczne DDD
- Techniki refaktoryzacji i wykorzystania wzorców projektowych
- Architektura w DDD i wzorce architektoniczne

### Wymagania:

Uczestnik szkolenia powinien posiadać doświadczenie w programowaniu obiektowym, a także podstawową znajomość zagadnień związanych z analizą i projektowaniem, w tym znajomość UML.

### Parametry szkolenia:

3\*8 godzin (3\*7 godzin netto) wykładów i warsztatów (z wyraźną przewagą warsztatów - 80%). Podczas warsztatów uczestnicy projektują system oraz jego wybrane moduły w praktyce poznając i dyskutując z trenerem koncepcje związane z DDD

### Program szkolenia:

1. Główne założenia DDD
  - I. Stosowalność DDD
  - II. Proces i role w DDD
  - III. Wspólny język



- IV. Architektura a DDD
- 2. Techniki modelowania
  - I. Efektywne modelowanie
    - i. Kruszenie wiedzy (Knowledge crunching)
    - ii. Ciągła nauka (Continuous Learning)
    - iii. Głęboki model (Deep models)
    - iv. Model-Driven Design
  - II. Komunikacja i język
    - i. Wspólny język (Ubiquitous Language)
    - ii. Prowadzenie sesji modelowania
    - iii. Techniki pracy w zespole
- 3. Wzorce taktyczne DDD (Building Blocks)
  - I. Izolacja domeny w architekturze
  - II. Paradygmaty modelu obiektowego
  - III. Asocjacje
  - IV. Encje
  - V. Value objects
  - VI. Serwisy
  - VII. Moduły
  - VIII. Agregaty
  - IX. Fabryki
  - X. Repozytoria i modelowanie pod kątem relacyjnych baz danych
- 4. Techniki ulepszania i refaktoryzacji
  - I. Wzorce projektowe GoF w kontekście DDD
    - i. Saga jako mediator i fasada
    - ii. Specyfikacja jako kompozyt (Composite)
    - iii. Dekorator polityk/strategii (Decorator)
    - iv. Stan dla agregatów (State)
    - v. Łańcuch odpowiedzialności (Chain of Responsibility)
    - vi. Budowniczy (Builder)
  - II. Wzorce projektowania gibkiego (Supple Design)
    - i. Interfejsy intencyjne (Intention-Revealing Interfaces)
    - ii. Funkcje bez efektów ubocznych (Side-Effect-Free Functions)
    - iii. Asercje (Assertions)
    - iv. Kontury koncepcyjne (Conceptual Contours)
    - v. Wolnostojące klasy (Standalone Classes)
    - vi. Domknięcia operacji (Closure of Operations)
    - vii. Projektowanie deklaratywne
- 5. Wzorce strategiczne DDD
  - I. Rodzaje domen (Core, Generic, Support) i techniki projektowania strategicznego domen
  - II. Ustalanie granic i integracji: Bounded Context oraz Context Map, Shared Kernel, Anticorruption Layer, Open Host Service
  - III. Praca międzyzespołowa: Customer/Supplier, Conformist, Separate Ways
- 6. Architektura
  - I. Architektura aplikacji a architektura systemu

- II. Projektowanie strategiczne
- III. Model dużej skali: Capability, Operations, Policy, Decision Support
- IV. Wzorce architektoniczne
  - i. Architektura warstwowa (Multilayered Architecture) z warstwami prezentacji, aplikacji, domeny i infrastruktury
  - ii. Architektura CqRS
  - iii. Architektura zdarzeniowa (Event-Driven Architecture)
  - iv. Architektura SOA

