

Kod szkolenia: **UGC**

Tytuł szkolenia: **Projektowanie aplikacji JEE z użyciem wzorców projektowych i notacji UML**

Dni: 5

## Opis:

### Adresaci Szkolenia:

Szkolenie adresowane jest zarówno do programistów jak i projektantów, którzy chcą tworzyć bardziej niezawodne, elastyczne i wydajne oprogramowanie, przy wykorzystaniu technologii JEE.

### Cel szkolenia:

Celem szkolenia jest zdobycie umiejętności projektowania aplikacji JEE w notacji UML, przy użyciu najlepszych, sprawdzonych i wydajnych rozwiązań opisanych w szablonach wzorców projektowych GOF i Core J2EE (z uwzględnieniem zmian w specyfikacji JEE). A tym samym obniżenie kosztów utrzymania tworzonego oprogramowania.

Na szkoleniu uczestnicy uczą się myśleć kategoriami gotowych rozwiązań, z perspektywy takich parametrów systemowych jak elastyczność, wydajność, niezawodność, utrzymanie, skalowalność i innych. Jednocześnie w czasie starannie zaprojektowanych ćwiczeń nabierają wprawy w posługiwaniu się tą wiedzą i przekuwając ją w umiejętności. Umiejętności tak tworzenia dobrych rozwiązań jak i wykrywania zagrożeń.

Jednocześnie w trakcie szkolenia przestrzega się również przed typowymi błędami określanymi jako antywzorce.

### Wymagania:

Szkolenie wprowadza w każdy aspekt niezbędny do projektowania aplikacji z użyciem wzorców GOF i Core J2EE. Tak więc nie ma wymagań wstępnych.

### Parametry szkolenia:

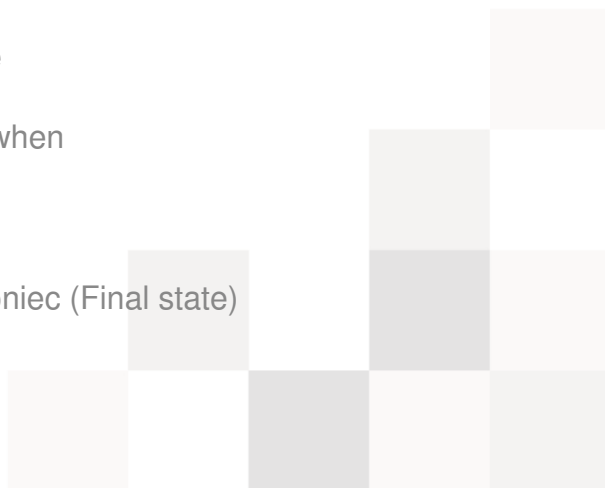
5\*8 godzin (5\*7 godzin netto) wykładów i warsztatów.

Wielkość grupy: maks. 8-10 osób.

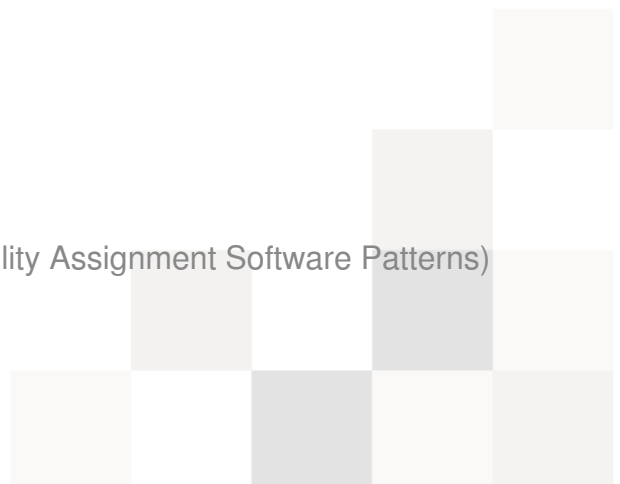
### Program szkolenia:



1. Metodyki wytwarzania oprogramowania
  - I. Metodyka kaskadowa
  - II. Unified Software Development Process - USDP (UP)
    - i. Wymiary
    - ii. Iteracje
    - iii. Fazy
      - A. Rozpoczęcie (Inception)
      - B. Opracowanie (Elaboration)
      - C. Budowa (Construction)
      - D. Wdrożenie (Transition)
  - III. Rational Unified Process (RUP)
  - IV. Extreme Programming (XP)
  - V. Scrum
2. Podstawy UML i analizy
  - I. Czym jest UML
  - II. Przegląd diagramów UML (i podstawowe kwalifikatory)
    - i. Diagram klas
      - A. Klasa (Class)
      - B. Interfejs (Interface)
    - ii. Diagram Obiektów
      - A. Obiekt (Object) nazwany
      - B. Obiekt anonimowy
    - iii. Diagram Pakietów
      - A. Pakiet (Package)
      - B. Stereotypy access i import
    - iv. Diagram komponentów
      - A. Komponent (Component)
      - B. Interfejsy wymagane i udostępniane
    - v. Diagram wdrożenia
      - A. Węzeł (Node)
      - B. Komponent (Component)
    - vi. Diagram struktur połączonych
      - A. Część (Part)
      - B. Port (Ports)
    - vii. Diagram przypadków użycia
      - A. Aktor (Actor)
      - B. Przypadek użycia (Use Case)
    - viii. Diagram maszyny stanów
      - A. Stan (state)
      - B. Przejścia wewnętrzne
      - C. do/entry/exit
      - D. Zdarzenia, while, at, when
    - ix. Diagram aktywności
      - A. Czynność (Activity)
      - B. Partycja (Partition)
      - C. Start (Initial state) i koniec (Final state)



- D. Decyzja (Decision)
- E. Rozwidlenie (Fork) i złączenie (Join)
- F. Obszar przerywania (Interruptable Activity Region) i przerywanie (Interrupt flow)
- x. Diagram komunikacji
  - A. Obiekty
  - B. Komunikaty
- xi. Diagram sekwencji
  - A. Obiekty
  - B. Komunikaty
  - C. Linia życia
  - D. Obszar wykonania (Execution Specification)
- xii. Diagram zależności czasowych
  - A. Notacja stanowa (State Timeline)
  - B. Notacja wartości (Value Timeline)
- xiii. Diagram przeglądu interakcji
  - A. Referencja do diagramu (Diagram Frame)
  - B. Wystąpienie interakcji (Interaction Occurrence)
- III. Mechanizmy rozszerzenia UML
  - i. Stereotypy (Stereotype)
  - ii. Etykiety (Tagged Values)
  - iii. Ograniczenia (Constraints)
- IV. Wybrane diagramy UML
  - i. Model przypadków użycia
    - A. Analiza wymagań i Diagram Przypadków użycia
    - B. Dokumentacja Przypadków Użycia
  - ii. Diagramy klas
    - A. Elementy diagramów klas
    - B. Identyfikacja klas – analiza CRC
  - iii. Diagram Sekwencji
    - A. Elementy diagramów sekwencji
    - B. Uproszczenie tworzenia diagramów sekwencji – model analityczny
      - a. <<boundary>>
      - b. <<entity>>
      - c. <<control>>
- 3. Podstawy projektowania obiektowego i wprowadzenie do wzorców
  - I. Enkapsulacja
  - II. High Cohension
  - III. Loose Coupling
  - IV. Command-Query Separation
  - V. Java i dziedziczenie
  - VI. Wprowadzenie do wzorców
    - i. Rodzaje wzorców
    - ii. GRASP (General Responsibility Assignment Software Patterns)
      - A. Information Expert



- B. Creator
- C. Controller
- D. Polymorphism
- E. Pure Fabrication
- F. Indirection
- G. Protected Variations
- iii. S.O.L.I.D (SOLID-ne programowanie)
  - A. Single Responsibility Principle
  - B. Open-Close Principle
  - C. Liskov Substitution Principle
  - D. Interface Segregation Principle
  - E. Dependency Inversion Principle

## 4. Wzorce GOF

### I. Wzorce konstrukcyjne

- i. Abstract Factory
- ii. Builder
- iii. Factory Method
- iv. Prototype
- v. Singleton

### II. Wzorce strukturalne

- i. Adapter
- ii. Bridge
- iii. Composite
- iv. Decorator
- v. Façade
- vi. Flyweight
- vii. Proxy

### III. Wzorce behawioralne

- i. Chain of responsibility
- ii. Command
- iii. Interpreter
- iv. Iterator
- v. Mediator
- vi. Memento
- vii. Observer
- viii. State
- ix. Strategy
- x. Template Method
- xi. Visitor

## 5. Wprowadzenie do JEE

### I. Modele aplikacji

- i. Web Centric
- ii. Application Centric
- iii. Enterprise

### II. Krótkie omówienie podstawowych technologii JEE

- i. JSTL



- ii. JSF
- iii. JAAS
- iv. JTA
- v. JPA
- vi. JNDI
- vii. JMS
- III. Krótkie omówienie komponentów JEE i kontenerów
  - i. Servlety
  - ii. Strony JSP
  - iii. EJB (2.x i 3.0)
    - A. Session Statefull EJB
    - B. Session Stateless EJB
    - C. Entity EJB
    - D. Message Driven Bean (MDB)
  - iv. WebService
- IV. Architektura komponentowa i wielowarstwowa
- 6. Wzorce Core J2EE - Wzorce warstwy prezentacji
  - I. Intercepting Filter
  - II. Front Controller
  - III. Application Controller
  - IV. Context Object
  - V. View Helper
  - VI. Composite View
  - VII. Dispatcher View
  - VIII. Service To Worker
- 7. Wzorce Core J2EE - Wzorce warstwy biznesowej
  - I. Business Delegate
  - II. Service Locator
  - III. Session Façade
  - IV. Application Service
  - V. Business Object
  - VI. Composite Entity
  - VII. Transfer Object (TO)
  - VIII. Transfer Object Assembler
  - IX. Value List Handler
- 8. Wzorce Core J2EE - Wzorce warstwy integracji
  - I. Data Access Object (DAO)
  - II. Service Activator
  - III. Domain Store
  - IV. Web Service Broker
- 9. Wzorce architektoniczne
  - I. Dependency Injection
  - II. MVC (Model View Control)
  - III. 3 modele aplikacji według SUN
    - i. Web-centric
    - ii. Application-centric



- iii. Enterprise
- IV. Architektura typu Enterprise e JEE
- V. Architektura wielowarstwowa (Layers Pattern)
- VI. Wzorce infrastruktury
  - i. Redundancja ścieżek
  - ii. Replikacja
  - iii. Równoważnie Obciążenia (Load Balancing)
  - iv. Klastry (Clustering)
  - v. Wysoka dostępność (HA – Height Availability; Failover)
  - vi. Forward Proxy Cache
- 10. Przegląd wybranych antywzorców
  - I. Golden Hammer
  - II. Lava Flow
  - III. Spaghetti Code
  - IV. Busy Waiting
  - V. Hardcoding
  - VI. Object Orgy
  - VII. Circular Dependency
  - VIII. The Blob (God Object)

