

## Kod Szkolenia: UML/ALL

## Tytuł Szkolenia: Modelowanie z użyciem notacji UML2

### Adresaci szkolenia:

Szkolenie adresowane jest do osób, które chciałyby poznać UML2 od podstaw. Zarówno dla programistów jak i przyszłych/początkujących analityków, projektantów czy architektów.

### Cel szkolenia:

Celem szkolenia jest zdobycie od podstaw umiejętności niezbędnych do stworzenia pełnego modelu począwszy od analizy wymagań, poprzez tworzenie modelu przypadków użycia, modelu analitycznego, architektury a skończywszy na samym modelu projektowym.

Łącząc wiedzę i doświadczenie stworzyliśmy dla Państwa szkolenie z UML o innowacyjnym programie. Początkowo szkolenie skupia się na najważniejszych i najczęściej używanych elementach języka UML2.x, po czym omawiane są te zaawansowane elementy diagramów w ramach powrotu po przeanalizowaniu podstaw UML. Dzięki temu maksymalnie skutecznie wykorzystuje się czas szkolenia na rozwijanie wśród uczestników umiejętności posługiwania się UML. Początkowo uczestnicy rozwijają swoje umiejętności modelowania opierając się na elementach najczęściej używanych. A po omówieniu podstaw UML poznają i ćwiczą elementy zaawansowane. W ten sposób dużo głębiej utrwalają sobie zdobytą wiedzę, dużo szybciej opanowują umiejętności modelowania i jednocześnie nabywają świadomość i umiejętność używania elementów zaawansowanych.

Jeśli interesują państwa jedynie podstawy UML, lub jedynie elementy zaawansowane, to zapraszamy do zapoznania się w ramach naszej oferty ze szkoleniem z podstaw "UML w realnych projektach"(3 dni) i ze szkoleniem z elementów zaawansowanych "UML zaawansowany"(2 dni).

### Parametry szkolenia:

5\*7 godzin wykładów i warsztatów w proporcji 1/3. W trakcie warsztatów, oprócz prostych ćwiczeń, analizowany i projektowany jest prosty system informatyczny. Wielkość grupy: maks. 8-10 osób.

### Wymagania:

Szkolenie opisuje proces modelowania od samych podstaw łącznie z wprowadzeniem do obiektowości, tak więc nie ma żadnych wymagań wstępnych stawianych uczestnikom.

### Polecane szkolenia poprzedzające:

sages

Brak



## Program szkolenia:

1. Wprowadzenie do UML
  - Projektowanie obiektowości
    - i. Klasa (Class)
    - ii. Obiekt (Object)
    - iii. Dziedziczenie (Inheritance)
    - iv. Atrybut (Attribute)
    - v. Metoda (Method)
    - vi. Interfejs (Interface)
    - vii. Enkapsulacja (Encapsulation)
    - viii. Polimorfizm (Polymorphism)
    - ix. Przeciążanie metod (Overloading)
    - x. Nadpisywanie metod (Overriding)
  - Czym jest modelowanie obiektowe
  - Czym jest a czym nie jest UML
  - Rozwój UML
  - Podstawowe elementy UML
    - i. Podstawowe kwalifikatory
      1. Klasa (Class)
      2. Interfejs (Interface)
      3. Obiekt (Object)
      4. Aktor (Actor)
      5. Przypadek Użycia (Use Case)
      6. Komponent (Component)
      7. Węzeł (Node)
    - ii. Relacje (Relationships)
      1. Asocjacja (Association)
      2. Asocjacja (Association)
      3. Zależność (Dependency)
      4. Realizacja (Realization)
    - iii. Diagramy (Diagrams)
    - iv. Pakiety (Package)
    - v. Komentarze (Note)
    - vi. Mechanizmy rozszerzenia
      1. Stereotypy (Stereoype)
      2. Etykiety (Tagged Values)
      3. Ograniczenia (Constraints)
  - Przegląd diagramów w UML 2.x
  - Diagramy UML 2.x a 1.x
  - Diagram a model UML
  - Perspektywy 4+1
  - Zastosowania UML
2. UML a metodyki wytwarzania oprogramowania
  - Metodyka kaskadowa
  - Unified Software Development Process - USDP (UP)
    - i. Założenia
    - ii. Wymiary
    - iii. Fazy
      1. Rozpoczęcie (Inception)
      2. Opracowanie (Elaboration)
      3. Budowa (Construction)
      4. Wdrożenie (Transition)
    - iv. Zadania

- v. Iteracje
- vi. Proces
  - Rational Unified Process (RUP)
  - Extreme Programming (XP)
- 3. Modelowanie procesów biznesowych w organizacji
  - Proces biznesowy
  - Po co modelować procesy biznesowe
  - Języki opisu procesów biznesowych
  - Diagram aktywności (activity diagram)
    - i. Modelowanie procesów i algorytmów w UML
    - ii. Aktywność (activity)
    - iii. Akcja (action)
    - iv. Przepływ sterowania (control flow)
      - 1. Przepływy warunkowe (guard)
    - v. Przepływy równoległe (fork, join)
    - vi. Początek i koniec
      - 1. Węzeł początkowy (initial node)
      - 2. Węzeł końca aktywności (activity final node)
      - 3. Węzeł końca przepływu (flow final node)
    - vii. Decyzje i złączenia (decision, merge)
      - viii. Przepływ obiektu (object flow)
        - 1. Stan obiektu
        - 2. Ograniczenia na obiekcie
    - ix. Wyjątki (exception)
    - x. Sygnały
      - 1. Generowanie sygnału (send signal action)
      - 2. Odbiór sygnału (accept event action)
      - 3. Sygnał czasowy
    - xi. Partycja (activity partition)
- 4. Analiza wymagań a model przypadków użycia (use case model)
  - Czym są wymagania i po co analiza
  - Trudności związane z analizą wymagań
  - Proces analizy wymagań
    - i. Analiza problemu
    - ii. Zbieranie wymagań
    - iii. Dokumentacja (opis wymagania)
    - iv. Grupowanie wymagań
    - v. Znajdowanie podobieństw i sprzeczności
    - vi. Definiowanie zależności (diagram zależności)
    - vii. Zatwierdzanie wymagań
      - viii. Ustalanie priorytetów
    - ix. Zarządzanie zmianami
  - Analiza przypadków użycia
    - i. Przypadek użycia
    - ii. Kroki i czynności w analizie
    - iii. Dokumentacja przypadków użycia
      - 1. Opis przypadków użycia
      - 2. Scenariusze
    - iv. Diagram przypadków użycia (use case diagram)
      - 1. Aktor (actor)
      - 2. Przypadek użycia (Use Case)
      - 3. Asocjacja (association)
      - 4. Zawieranie przypadków użycia «include»
      - 5. Rozszerzanie przypadków użycia «extend»

- 6. Generalizacja
- 7. Granica kontekstu
- 8. Dokumentowanie scenariusza
- 5. Tworzenie modelu analitycznego
  - Po co i kiedy warto
  - Stereotypy analityczne
    - i. «boundary»
    - ii. «control»
    - iii. «entity»
  - Zasady modelowania analitycznego
  - Diagram komunikacji (communication diagram)
    - i. Obiekt (object)
    - ii. Komunikat (message)
      - 1. Synchroniczny (synchronous message)
      - 2. Asynchroniczny (asynchronous message)
      - 3. Zwrotny (return message)
      - 4. Utworzenie obiektu «create»
      - 5. Zniszczenie obiektu «destroy»
    - iii. Kolejność komunikatów
      - 1. Notacja zagnieżdżona
  - Diagram sekwencji (interaction diagram)
    - i. Linia życia (life line)
    - ii. Komunikat (message)
    - iii. Rodzaje komunikatów
      - 1. Synchroniczny (synchronous message)
      - 2. Asynchroniczny (asynchronous message)
      - 3. Zwrotny (return message)
      - 4. Utworzenie obiektu «create»
      - 5. Zniszczenie obiektu «destroy» i destruction event
    - iv. Ośrodek sterowania (execution specification)
    - v. Bloki złożone (combined fragment)
      - 1. Alternatywy (alternatives) - alt
      - 2. Pętla (loop) - loop
      - 3. Współbieżność (parallel) - par
    - vi. Bloki złożone a notacja w UML 1.X
      - 1. Dawniej alternatywy
      - 2. Dawniej pętle (iteracja)
      - 3. Dawniej współbieżność
  - Diagramy komunikacji a diagramy sekwencji - przekształcanie
  - Analiza CRC
    - i. Karta CRC
    - ii. Kroki analizy CRC
      - 1. Identyfikacja klas
      - 2. Wykonanie scenariuszy
    - iii. Analiza CRC a rzeczywistość
- 6. Model architektoniczny
  - Czym jest architektura systemu
  - Cele tworzenia architektury
  - Diagram komponentów (component diagram)
    - i. Komponent (component)
    - ii. Komponenty zagnieżdżone
    - iii. Interfejs (interface)
      - 1. Interfejs wymagany (required interface)
      - 2. Interfejs dostarczany (provided interface)

- iv. Złączenie (assembly)
- Diagram wdrożenia (deployment diagram)
  - i. Węzeł (node)
  - ii. Łącze (communication path)
    - 1. Łącze kierunkowe
    - 2. Liczność łącza
- 7. Statyczne aspekty modelu projektowego
  - Różnice między modelem projektowym a analitycznym
  - Statyczne aspekty modelu projektowego
  - Diagram klas (class diagram)
    - i. Klasa (class)
      - 1. Elementy klasy (atrybuty, metody)
      - 2. Widoczność (visibility)
      - 3. Atrybuty i metody statyczne
    - ii. Uogólnienie (generalization)
    - iii. Klasy abstrakcyjne (abstract class)
      - 1. Metody abstrakcyjne
    - iv. Interfejs (interface)
    - v. Realizacja (realization)
    - vi. Relacja zależności
      - 1. Stereotypy zależności
        - a. «refine»
        - b. «trace»
        - c. «derive»
        - d. «instanciate»
        - e. «send»
        - f. «call»
    - vii. Asocjacja (association)
      - 1. Cechy asocjacji
        - a. Nazwa asocjacji (name)
        - b. Rola (role)
        - c. Nawigowalność (navigablility)
        - d. Wielokrotność (multiplicity)
      - 2. Asocjacja zwrotna i wielokrotna
      - 3. Rodzaje asocjacji
        - a. Asocjacja (association)
        - b. Agregacja (aggregation)
        - c. Kompozycja (composition)
      - 4. Klasa asocjacyjna (association class)
  - Diagram obiektów (object diagram)
    - i. Instancja (instance specification)
- 8. Dynamiczne aspekty modelu projektowego
  - Dynamiczne aspekty modelu projektowego
  - Implementacyjny diagram komunikacji
  - Implementacyjny diagram sekwencji
  - Diagram maszyny stanowej (state machine diagram)
    - i. Stan (state)
    - ii. Stan złożony
    - iii. Przejście (transition)
      - 1. Proste
      - 2. Automatyczne
      - 3. Zwrotne
      - 4. Wewnętrzne
    - iv. Rodzaje zdarzeń

- 1. Czasowe (time event)
  - a. Względne (after)
  - b. Bezwzględne (at)
- 2. Zdarzenie zmiany stanu (change event - when)
- 3. Wywołanie (call event)
- 4. Sygnał (signal)
- 5. Dowlone (all)
- v. Sygnatura przejścia
- vi. Pseudo-stan (pseudostate)
  - 1. Ropoczęcie (initial)
  - 2. Zakończenie (final)
  - 3. Zniszczenie (terminate)
  - 4. Wybór (choice)
  - 5. Punkt węzłowy (junction)
  - 6. Rozwidlenie (fork)
  - 7. Złączenie (join)
- vii. Czynności w stanie
  - 1. entry
  - 2. do
  - 3. exit
  - 4. Przejście wewnętrzne
- 9. Zaawansowane elementy diagramów UML
  - Zaawansowane elementy diagramu aktywności (activity diagram)
    - i. Waga (weight) w przepływie sterowania (control flow)
    - ii. Łączniki
    - iii. Zaawansowane reprezentacje przepływu obiektów (object flow)
      - 1. Pin wejściowy i wyjściowy
      - 2. Pin o charakterze wyjątku
      - 3. Pin w przepływie strumieniowym
      - 4. Anonimowa notacja pinów
      - 5. Zestaw pinów
    - iv. Specyfikacja złączenia (join specification) dla przepływów równoległych
    - v. Parametr czynności (activity parameter node)
    - vi. Stereotypy obiektów
      - 1. Bufor centralny (central buffer)
      - 2. Składowa danych (datastore)
    - vii. Obszar przerywalny (interruptible region)
      - 1. Sygnały a obszar przerywalny
    - viii. Region rozszerzenia (expansion region)
      - 1. Iteracja (iterative)
      - 2. Przetwarzanie równoległe (parallel)
      - 3. Strumień (stream)
    - ix. Uproszczony zapis obszaru rozszerzenia
  - Zaawansowane elementy diagramu sekwencji (interaction diagram)
    - i. Komunikaty
      - 1. Komunikat opcjonalny (balking message)
      - 2. Komunikat oczekujący (timeout message)
      - 3. Komunikat odnaleziony (found message)
      - 4. Komunikat zgubiony (lost message)
    - ii. Blok złożony (combined fragment)
      - 1. Alternatywy (alternatives) - alt
      - 2. Pętla (loop) - loop
      - 3. Równoległość (parallel) - par
      - 4. Blok krytyczny (critical) - critical

- 5. Funkcjonalność nieprawidłowa (negative) - neg
- 6. Ścisłe uporządkowanie (strict sequencing) - stricte
- 7. Słabe uporządkowanie (weak sequencing) - seq
- 8. Istotność - consider
- 9. Nieistotność - ignore
- 10. Formuła - assert
- 11. Blok odwołania (interaction use) - ref
- 12. Przerwanie (break) - break
- 13. Opcja (option) - opt
- iii. Bramy
  - 1. Bramy formalne (formal gates)
  - 2. Bramy właściwe (actual gates)
  - 3. Bramy wyrażeniowe (expression gates)
- Zaawansowane elementy diagramu komunikacji (communication diagram)
  - i. Obiekt wielokrotny
  - ii. Poprzednik
  - iii. Współbieżność
  - iv. Klasy aktywne
- Zaawansowane elementy diagramu klas (class diagram)
  - i. Zobowiązania (responsibilities)
  - ii. Klasy szablonowe (template class)
  - iii. Zbiory dziedziczenia (generalization set)
  - iv. Ograniczenia uogólnienia
    - 1. complete
    - 2. incomplete
    - 3. disjoint
    - 4. overlapping
  - v. Asocjacje
    - 1. Ograniczenia końców asocjacji
      - a. Subsets
      - b. Redefined
      - c. Union
      - d. Ordered
      - e. Bag
      - f. Sequence
    - 2. Asocjacja n-arna
    - 3. Asocjacja kwalifikowana (qualified association)
- Zaawansowane elementy diagramu maszyny stanowej (state machine diagram)
  - i. Zdarzenia odroczone (defer)
  - ii. Pseudo stany w stanie złożonym
    - 1. Wejście i wyjście (entry/exit point)
    - 2. Płytki i głęboka historia (shallow/deep history)
  - iii. Stany współbieżne
- 10. Zaawansowane elementy architektury
  - Zaawansowane elementy diagramu komponentów (component diagram)
    - i. Porty
    - ii. Konektory
  - Zaawansowane elementy diagramu wdrożenia (deployment diagram)
    - i. Niskopoziomowe diagramy wdrożenia
    - ii. Szablony architektoniczne
    - iii. Model wdrożenia na diagramach wdrożenia
      - 1. Po co model wdrożenia
      - 2. Artefakt
      - 3. Stereotypy artefaktów

- a. «file»
- b. «document»
- c. «library»
- d. «executable»
- e. «script»
- f. «source»
- 4. Specyfikacja konfiguracji (deployment specification)
- 5. Relacje między artefaktami
  - a. Kompozycji (composition)
  - b. Zależności (dependency)
- 6. Instalacja artefaktów (deployment) «deploy»
- 7. Manifestacja (manifestation) «manifest»
- Diagram pakietów (package diagram)
  - i. Pakiet
  - ii. Zagnieżdżanie (nest)
  - iii. Przestrzeń nazw
  - iv. Importowanie (package import)
    - 1. «import»
    - 2. «access»
  - v. Łączenie (merge)
  - vi. Diagramy pakietów i modelowanie warstw architektury
- 11. Wzorce projektowe GOF
  - Czym jest wzorzec
  - Zalety stosowanie wzorców
  - Niebezpieczeństwa stosowania wzorców
  - Kategorie wzorców
  - Wzorce projektowe GOF
    - i. Wzorce konstrukcyjne (creational patterns)
      - 1. Abstract Factory
      - 2. Builder
      - 3. Factory Method
      - 4. Prototype
      - 5. Singleton
    - ii. Wzorce strukturalne (structural patterns)
      - 1. Adapter
      - 2. Bridge
      - 3. Composite
      - 4. Decorator
      - 5. Façade
      - 6. Flyweight
      - 7. Proxy
    - iii. Wzorce czynnościowe (behavioral patterns)
      - 1. Chain of responsibility
      - 2. Command
      - 3. Interpreter
      - 4. Iterator
      - 5. Mediator
      - 6. Memento
      - 7. Observer
      - 8. State
      - 9. Strategy
      - 10. Template method
      - 11. Visitor
- 12. Nowe diagramy UML 2.0

- Diagram struktur połączonych (composite structure diagram)
    - i. Część (part)
      - 1. Port
      - 2. Interfejs udostępniany
      - 3. Interfejs wymagany
      - 4. Złączenie (assembly)
    - ii. Połączenie
    - iii. Wzorce na diagramach UML
    - iv. Współpraca (collaboration)
      - 1. Rola (role)
      - 2. Przypisanie roli
      - 3. Element współpracujący
    - v. Wystąpienie współpracy (collaboration use)
  - Diagram harmonogramowania (timing diagram)
    - i. Zdarzenia i ograniczenia
      - 1. event
      - 2. time constraint
      - 3. duration constraint
      - 4. time observation
    - ii. Komunikaty (messages)
    - iii. 2 notacje diagramów harmonogramowania
    - iv. Harmonizacja linii zmiany stanów
    - v. Przesyłanie komunikatów
    - vi. Diagramy sekwencji a harmonogramowanie
  - Diagram przeglądu interakcji (interaction overview diagram)
    - i. Podstawowe elementy
      - 1. Przepływ sterowania
      - 2. Początek
      - 3. Koniec
      - 4. Zakończenie przepływu
      - 5. Fragment interakcji
      - 6. Wystąpienie interakcji
    - ii. Bloki złożone (combined fragment) a diagram przeglądu interakcji
      - 1. Alternatywa
      - 2. Współbieżność
      - 3. Iteracja
      - 4. Opcja
13. Inne elementy zaawansowane UML
- Meta Object Facility (MOF)
    - i. 4 warstwowa budowa modelu
  - Pakiet CORE - UML 2 Infrastructure
    - i. primitive types
    - ii. abstractions
    - iii. basic
    - iv. constructs
  - Profile UML
    - i. Profil (profile)
    - ii. Meta-Klasa (metaclass)
    - iii. Stereotyp (stereotype)
    - iv. Rozszerzenie (extension)
    - v. Użycie profilu (profile application)
  - Język ograniczeń OCL
    - i. Zastosowanie OCL
    - ii. Składnia OCL

- iii. Kontekst wyrażenia
- iv. Zastosowanie OCL na diagramach UML
- Model Driven Architecture (MDA)
  - i. Modele MDA
    - 1. Computation independend model (CIM)
    - 2. Platform independend model (PIM)
    - 3. Platform specific model (PSM)
    - 4. Platform model
  - ii. Transformacja