

Kod szkolenia: **UML/ANA**

Tytuł szkolenia: **UML2 dla analityków**

Dni: 4

## Opis:

### Adresaci Szkolenia:

Szkolenie profilowane jest przede wszystkim dla analityków, którzy chcą modelować aplikacje, organizacje i procesy za pomocą UML2. Przydatne jest również dla testerów, którzy muszą weryfikować poprawność działania aplikacji w oparciu o dokumentację w UML2.

Szkolenie jest również doskonałe dla programistów i testerów, którzy mają nadzieję na awans w kierunku analityka.

Na szkoleniu poruszane są również podstawy architektury.

### Cel szkolenia:

Celem szkolenia jest zdobycie umiejętności analitycznych - tak w zakresie analizy wymagań, modelowania organizacji i procesów (analiza biznesowa), jak i analizy tworzonej aplikacji (analiza systemowa) z użyciem notacji UML2. Szkolenie profilowane jest na analizę, niemniej zawiera również podstawy architektury, czy omówienie różnic między analizą a projektowaniem, dzięki czemu uczestnicy poznają wszystkie obszary UML2.

Ponieważ celem jest zdobycie umiejętności analitycznych a nie samo poznanie języka UML, duży nacisk w szkoleniu jest nałożony na modelowanie w trakcie ćwiczeń.

### Wymagania:

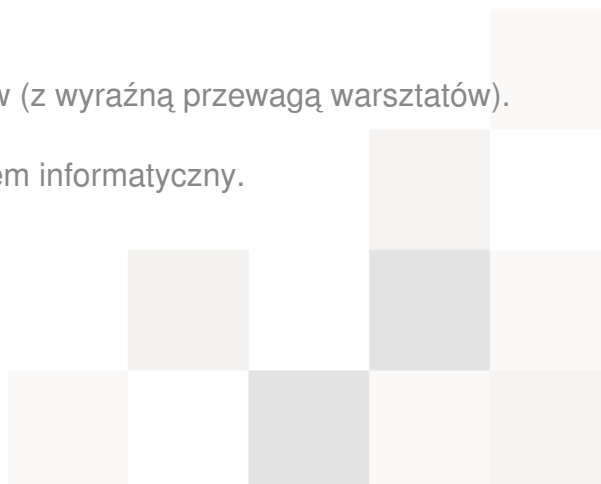
Szkolenie opisuje proces analizy od samych podstaw łącznie z wprowadzeniem do obiektowości, tak więc nie ma żadnych wymagań wstępnych stawianych uczestnikom.

### Parametry szkolenia:

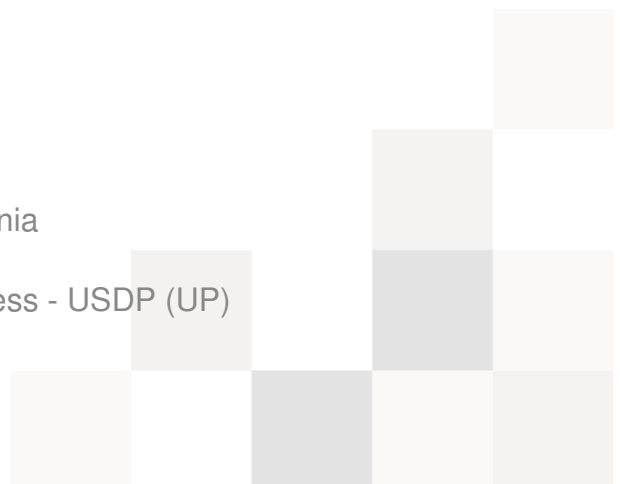
4\*8 godzin (4\*7 godzin netto) wykładów i warsztatów (z wyraźną przewagą warsztatów).

W trakcie warsztatów projektowany jest prosty system informatyczny.

Wielkość grupy: maks. 8-10 osób.

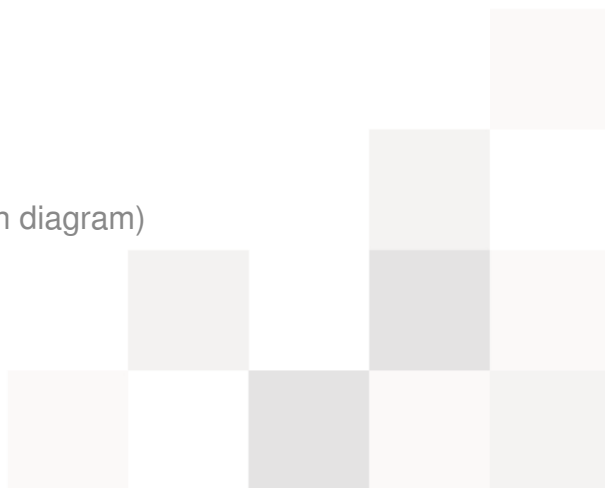


1. Wprowadzenie do UML
  - I. Projektowanie obiektowości
    - i. Klasa (Class)
    - ii. Obiekt (Object)
    - iii. Dziedziczenie (Inheritance)
    - iv. Atrybut (Attribute)
    - v. Metoda (Method)
    - vi. Interfejs (Interface)
    - vii. Enkapsulacja (Encapsulation)
    - viii. Polimorfizm (Polymorphism)
    - ix. Przeciążanie metod (Overloading)
    - x. Nadpisywanie metod (Overriding)
  - II. Czym jest modelowanie obiektowe
  - III. Czym jest a czym nie jest UML
  - IV. Rozwój UML
  - V. Podstawowe elementy UML
    - i. Podstawowe kwalifikatory
      - A. Klasa (Class)
      - B. Interfejs (Interface)
      - C. Obiekt (Object)
      - D. Aktor (Actor)
      - E. Przypadek Użycia (Use Case)
      - F. Komponent (Component)
      - G. Węzeł (Node)
    - ii. Relacje (Relationships)
      - A. Asocjacja (Association)
      - B. Asocjacja (Association)
      - C. Zależność (Dependency)
      - D. Realizacja (Realization)
    - iii. Diagramy (Diagrams)
    - iv. Pakiety (Package)
    - v. Komentarze (Note)
    - vi. Mechanizmy rozszerzenia
      - A. Stereotypy (Stereoype)
      - B. Etykiety (Tagged Values)
      - C. Ograniczenia (Constraints)
  - VI. Przegląd diagramów w UML 2.x
  - VII. Diagramy UML 2.x a 1.x
  - VIII. Diagram a model UML
  - IX. Perspektywy 4+1
  - X. Zastosowania UML
2. UML a metodyki wytwarzania oprogramowania
  - I. Metodyka kaskadowa
  - II. Unified Software Development Process - USDP (UP)



- i. Założenia
  - ii. Wymiary
  - iii. Fazy
    - A. Rozpoczęcie (Inception)
    - B. Opracowanie (Elaboration)
    - C. Budowa (Construction)
    - D. Wdrożenie (Transition)
  - iv. Zadania
  - v. Iteracje
  - vi. Proces
  - III. Rational Unified Process (RUP)
  - IV. Extreme Programming (XP)
  - V. Scrum
3. Modelowanie procesów biznesowych w organizacji
- I. Procesy biznesowe
  - II. Po co modelować procesy biznesowe
  - III. Języki opisu procesów biznesowych
  - IV. Diagram aktywności (activity diagram)
    - i. Modelowanie procesów i algorytmów w UML
    - ii. Aktywność (activity)
    - iii. Akcja (action)
    - iv. Przepływ sterowania (control flow)
      - A. Przepływy warunkowe (guard)
    - v. Przepływy równoległe (fork, join)
    - vi. Początek i koniec
      - A. Węzeł początkowy (initial node)
      - B. Węzeł końca aktywności (activity final node)
      - C. Węzeł końca przepływu (flow final node)
    - vii. Decyzje i złączenia (decision, merge)
    - viii. Przepływ obiektu (object flow)
      - A. Stan obiektu
      - B. Ograniczenia na obiekcie
    - ix. Wyjątki (exception)
    - x. Sygnały
      - A. Generowanie sygnału (send signal action)
      - B. Odbiór sygnału (accept event action)
      - C. Sygnał czasowy
    - xi. Partycja (activity partition)
    - xii. Waga (weight) w przepływie sterowania (control flow)
    - xiii. Zaawansowane reprezentacje przepływu obiektów (object flow)
      - A. Pin wejściowy i wyjściowy
      - B. Pin o charakterze wyjątku
      - C. Pin w przepływie strumieniowym
      - D. Anonimowa notacja pinów
    - xiv. Specyfikacja złączenia (join specification) dla przepływów równoległych
    - xv. Wyjątki

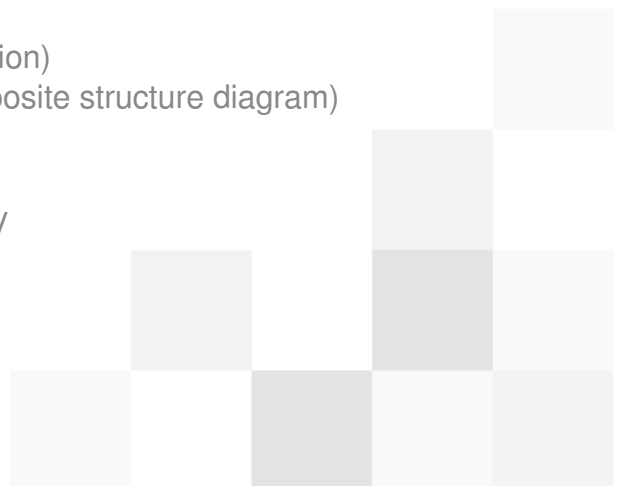
- xvi. Stereotypy obiektów
  - A. Bufor centralny (central buffer)
  - B. Składnica danych (datastore)
- xvii. Obszar przerywalny (interruptible region)
  - A. Sygnały a obszar przerywalny
- xviii. Region rozszerzenia (expansion region)
  - A. Iteracja (iterative)
  - B. Przetwarzanie równoległe (parallel)
  - C. Strumień (stream)
- 4. Analiza wymagań
  - I. Czym są wymagania i po co analiza
  - II. Trudności związane z analizą wymagań
  - III. 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
- 5. Model przypadków użycia (use case model)
  - I. Przypadek użycia
  - II. Kroki i czynności w analizie
  - III. Dokumentacja przypadków użycia
    - i. Opis przypadków użycia
    - ii. Scenariusze
  - IV. Diagram przypadków użycia (use case diagram)
    - i. Aktor (actor)
    - ii. Przypadek użycia (Use Case)
    - iii. Asocjacja (association)
    - iv. Zawieranie przypadków użycia <<include>>
    - v. Rozszerzanie przypadków użycia <<extend>>
    - vi. Generalizacja
    - vii. Granica kontekstu
    - viii. Dokumentowanie scenariusza
- 6. Tworzenie modelu analitycznego
  - I. Po co i kiedy warto
  - II. Stereotypy analityczne
    - i. <<boundary>>
    - ii. <<control>>
    - iii. <<entity>>
  - III. Zasady modelowania analitycznego
  - IV. Diagram komunikacji (communication diagram)
    - i. Obiekt (object)
    - ii. Komunikat (message)



- A. Synchroniczny (synchronous message)
  - B. Asynchroniczny (asynchronous message)
  - C. Zwrotny (return message)
  - D. Utworzenie obiektu <<create>>
  - E. Zniszczenie obiektu <<destroy>>
  - iii. Kolejność komunikatów
    - A. Notacja zagnieżdżona
  - V. Diagram sekwencji (interaction diagram)
    - i. Linia życia (life line)
    - ii. Komunikat (message)
    - iii. Rodzaje komunikatów
      - A. Synchroniczny (synchronous message)
      - B. Asynchroniczny (asynchronous message)
      - C. Zwrotny (return message)
      - D. Utworzenie obiektu <<create>>
      - E. Zniszczenie obiektu <<destroy>> i destruction event
      - F. Komunikat odnaleziony (found message)
      - G. Komunikat zgubiony (lost message)
    - iv. Ośrodek sterowania (execution specification)
    - v. Bloki złożone (combined fragment)
      - A. Alternatywy (alternatives) – alt
      - B. Pętla (loop) – loop
      - C. Współbieżność (parallel) – par
      - D. Przerwanie (break) - break
      - E. Blok krytyczny (critical) - critical
      - F. Blok odwołania (interaction use) - ref
    - vi. Bloki złożone a notacja w UML 1.X
      - A. Dawniej alternatywy
      - B. Dawniej pętle (iteracja)
      - C. Dawniej współbieżność
  - VI. Diagramy komunikacji a diagramy sekwencji – przekształcanie
  - VII. Analiza CRC
    - i. Karta CRC
    - ii. Kroki analizy CRC
      - A. Identyfikacja klas
      - B. Wykonanie scenariuszy
    - iii. Analiza CRC a rzeczywistość
  - VIII. Diagram przeglądu interakcji (interaction overview diagram)
    - i. Podstawowe elementy
      - A. Przepływ sterowania
      - B. Początek
      - C. Koniec
      - D. Zakończenie przepływu
      - E. Fragment interakcji
      - F. Wystąpienie interakcji
7. Statyczne aspekty modelu analitycznego



- I. Różnice między modelem projektowym a analitycznym
- II. Diagram klas (class diagram)
  - i. Klasa (class)
    - A. Elementy klasy (atributy, metody)
    - B. Widoczność (visibility)
    - C. Atrybuty i metody statyczne
  - ii. Uogólnienie (generalization)
  - iii. Klasy abstrakcyjne (abstract class)
    - A. Metody abstrakcyjne
  - iv. Interfejs (interface)
  - v. Realizacja (realization)
  - vi. Relacja zależności
    - A. Stereotypy zależności
      - a. <<refine>>
      - b. <<trace>>
      - c. <<derive>>
      - d. <<instanciate>>
      - e. <<send>>
      - f. <<call>>
  - vii. Asocjacja (association)
    - A. Cechy asocjacji
      - a. Nazwa asocjacji (name)
      - b. Rola (role)
      - c. Nawigowalność (navigability)
      - d. Wielokrotność (multiplicity)
    - B. Asocjacja zwrotna i wielokrotna
    - C. Rodzaje asocjacji
      - a. Asocjacja (association)
      - b. Agregacja (aggregation)
      - c. Kompozycja (composition)
    - D. Klasa asocjacyjna (association class)
    - E. Asocjacja kwalifikowana (qualified association)
  - viii. Zbiory dziedziczenia (generalization set)
  - ix. Ograniczenia uogólnienia
    - A. complete
    - B. incomplete
    - C. disjoint
    - D. overlapping
- III. Diagram obiektów (object diagram)
  - i. Instancja (instance specification)
- IV. Diagram struktur połączonych (composite structure diagram)
  - i. Część (part)
    - A. Port
    - B. Interfejs udostępniany
    - C. Interfejs wymagany
    - D. Złączenie (assembly)



- ii. Połączenie
  - iii. Współpraca (collaboration)
    - A. Rola (role)
    - B. Przypisanie roli
    - C. Element współpracujący
  - iv. Wystąpienie współpracy (collaboration use)
8. Diagram maszyny stanowej (state machine diagram)
- I. Stan (state)
  - II. Stan złożony
  - III. Przejście (transition)
    - i. Proste
    - ii. Automatyczne
    - iii. Zwrotne
    - iv. Wewnętrzne
  - IV. Rodzaje zdarzeń
    - i. Czasowe (time event)
      - A. Względne (after)
      - B. Bezwzględne (at)
    - ii. Zdarzenie zmiany stanu (change event - when)
    - iii. Wywołanie (call event)
    - iv. Sygnał (signal)
    - v. Dowlone (all)
  - V. Sygnatura przejścia
  - VI. Pseudo-stan (pseudostate)
    - i. Ropoczęcie (initial)
    - ii. Zakończenie (final)
    - iii. Zniszczenie (terminate)
    - iv. Wybór (choice)
    - v. Punkt węzłowy (junction)
    - vi. Rozwidlenie (fork)
    - vii. Złączenie (join)
  - VII. Czynności w stanie
    - i. entry
    - ii. do
    - iii. exit
    - iv. Przejście wewnętrzne
  - VIII. Zdarzenia odroczone (deffer)
  - IX. Pseudo stany w stanie złożonym
    - i. Wejście i wyjście (entry/exit point)
    - ii. Płytko i głęboka historia (shallow/deep history)
  - X. Stany współbieżne
9. Diagram harmonogramowania (timing diagram)
- I. Komunikaty (messages)
  - II. 2 notacje diagramów harmonogramowania
  - III. Przesyłanie komunikatów
10. Model architektoniczny



- I. Czym jest architektura systemu
  - II. Cele tworzenia architektury
  - III. Diagram komponentów (component diagram)
    - i. Komponent (component)
    - ii. Komponenty zagnieżdżone
    - iii. Interfejs (interface)
      - A. Interfejs wymagany (required interface)
      - B. Interfejs dostarczany (provided interface)
    - iv. Złączenie (assembly)
    - v. Porty
    - vi. Konektory
  - IV. Diagram wdrożenia (deployment diagram)
    - i. Węzeł (node)
    - ii. Łącze (communication path)
      - A. Łącze kierunkowe
      - B. Liczność łącza
    - iii. Model wdrożenia na diagramach wdrożenia
    - iv. Po co model wdrożenia
    - v. Artefakt
    - vi. Stereotypy artefaktów
      - A. <<file>>
      - B. <<document>>
      - C. <<library>>
      - D. <<executable>>
      - E. <<script>>
      - F. <<source>>
  - V. Diagram pakietów (package diagram)
    - i. Pakiet
    - ii. Zagnieżdżanie (nest)
    - iii. Przestrzeń nazw
    - iv. Importowanie (package import)
      - A. <<import>>
      - B. <<access>>
11. Elementy zaawansowane UML
- I. Meta Object Facility (MOF)
    - i. 4-warstwowa budowa modelu
  - II. Pakiet CORE - UML 2 Infrastructure
    - i. primitive types
    - ii. abstractions
    - iii. basic
    - iv. constructs
  - III. Profile UML
    - i. Profil (profile)
    - ii. Meta-Klasa (metaclass)
    - iii. Stereotyp (stereotype)
    - iv. Rozszerzenie (extension)





- v. Użycie profilu (profile application)
- IV. Język ograniczeń OCL
  - i. Zastosowanie OCL
  - ii. Składnia OCL
  - iii. Kontekst wyrażenia
  - iv. Zastosowanie OCL na diagramach UML
- V. Model Driven Architecture (MDA)
  - i. Modele MDA
    - A. Computation independend model (CIM)
    - B. Platform independend model (PIM)
    - C. Platform specific model (PSM)
    - D. Platform model
  - ii. Transformacja

