

Kod szkolenia: **UML/PRO**

Tytuł szkolenia: **UML2 dla projektantów**

Dni: 4

## Opis:

### Adresaci Szkolenia:

Szkolenie profilowane jest przede wszystkim dla projektantów, którzy chcą modelować aplikacje za pomocą UML2. Przydatne jest również dla programistów, którzy muszą wytwarzać oprogramowanie na podstawie dokumentacji w UML2.

Szkolenie jest również doskonałe dla programistów, którzy mają nadzieję na awans w kierunku projektanta.

a szkoleniu poruszane są również podstawy analizy i architektury.

### Cel szkolenia:

Celem szkolenia jest zdobycie umiejętności projektowania aplikacji w UML2. Szkolenie profilowane jest na projektowanie, niemniej zawiera również podstawy analizy, dzięki czemu uczestnicy uczą się jak wytworzyć pełną dokumentację projektową nawet gdy w ramach projektu nie są zaangażowani analitycy, lub model analityczny jest niepełny.

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

### Wymagania:

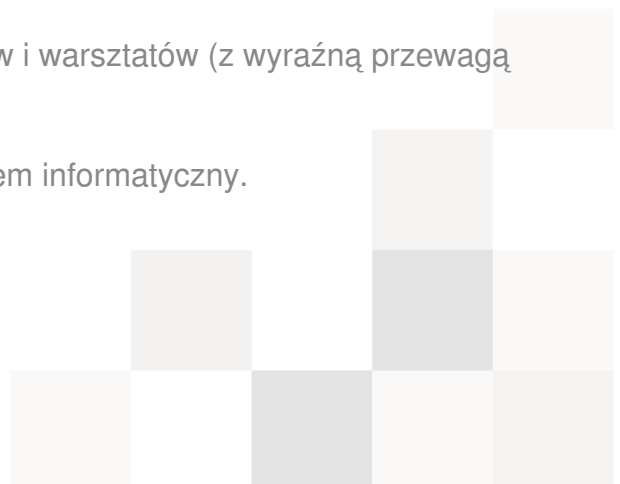
Szkolenie opisuje proces projektowania od samych podstaw łącznie z wprowadzeniem do obiektowości, tak więc nie ma żadnych wymagań wstępnych stawianych uczestnikom.

### Parametry szkolenia:

4\*8 godzin (4\*7 godzin netto) efektywnych 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)

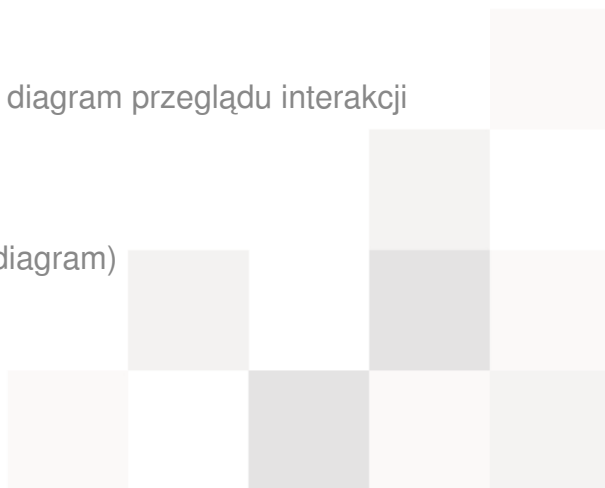


- i. Założenia
  - ii. Wymiary
  - iii. Fazy
    - A. Rozpoczęcie (Inception)
    - B. Opracowanie (Elaboration)
    - C. Budowa (Construction)
    - D. Wdrożenie (Transition)
  - iv. Zadania
  - v. Iteracje
  - vi. Proces
  - III. Rational Unified Process (RUP)
  - IV. Extreme Programming (XP)
  - V. Scrum
3. Model przypadków użycia (use case model) jako podstawa do projektowania
- 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
4. Modelowanie procesów i algorytmów - diagram aktywności (activity diagram)
- I. Aktywność (activity)
  - II. Akcja (action)
  - III. Przepływ sterowania (control flow)
    - i. Przepływy warunkowe (guard)
  - IV. Przepływy równoległe (fork, join)
  - V. Początek i koniec
    - i. Węzeł początkowy (initial node)
    - ii. Węzeł końca aktywności (activity final node)
    - iii. Węzeł końca przepływu (flow final node)
  - VI. Decyzje i złączenia (decision, merge)
  - VII. Przepływ obiektu (object flow)
    - i. Stan obiektu
    - ii. Ograniczenia na obiekcie
  - VIII. Wyjątki (exception)
  - IX. Sygnały
    - i. Generowanie sygnału (send signal action)
    - ii. Odbiór sygnału (accept event action)

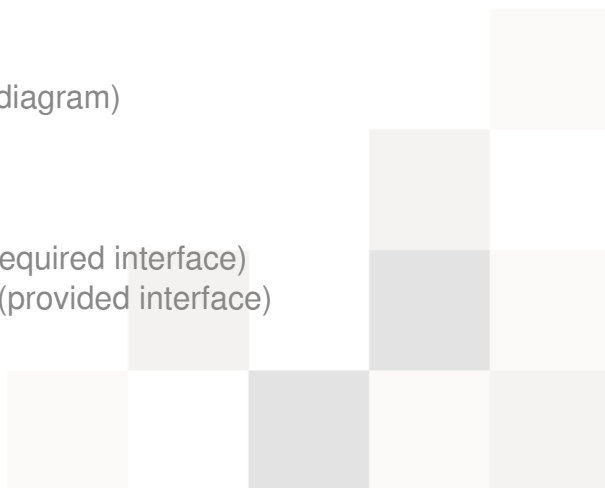
- iii. Sygnał czasowy
- X. Partycja (activity partition)
- XI. Waga (weight) w przepływie sterowania (control flow)
- XII. Zaawansowane reprezentacje przepływu obiektów (object flow)
  - i. Pin wejściowy i wyjściowy
  - ii. Pin o charakterze wyjątku
  - iii. Pin w przepływie strumieniowym
  - iv. Anonimowa notacja pinów
- XIII. Specyfikacja złączenia (join specification) dla przepływów równoległych
- XIV. Wyjątki
- XV. Stereotypy obiektów
  - i. Bufor centralny (central buffer)
  - ii. Składnica danych (datastore)
- XVI. Obszar przerywalny (interruptible region)
  - i. Sygnały a obszar przerywalny
- XVII. Region rozszerzenia (expansion region)
  - i. Iteracja (iterative)
  - ii. Przetwarzanie równoległe (parallel)
  - iii. Strumień (stream)
- 5. Statyczne aspekty modelu projektowego
  - I. Różnice między modelem projektowym a analitycznym
  - II. Statyczne aspekty modelu projektowego
  - III. Diagram klas (class diagram)
    - i. Klasa (class)
      - A. Elementy klasy (atrybuty, 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)
- viii. Klasy szablonowe (template class)
- ix. Zbiory dziedziczenia (generalization set)
- x. Ograniczenia uogólnienia
  - A. complete
  - B. incomplete
  - C. disjoint
  - D. overlapping
- xi. Asocjacje
  - A. Ograniczenia końców asocjacji
    - a. subsets
    - b. redefined
    - c. union
    - d. ordered
    - e. bag
    - f. sequence
  - B. Asocjacja kwalifikowana (qualified association)
- IV. Diagram obiektów (object diagram)
  - i. Instancja (instance specification)
- V. Diagram struktur połączonych (composite structure diagram)
  - i. Część (part)
    - A. Port
    - B. Interfejs udostępniany
    - C. Interfejs wymagany
    - D. Złączenie (assembly)
  - ii. Połączenie
  - iii. Wzorce na diagramach UML
  - iv. Współpraca (collaboration)
    - A. Rola (role)
    - B. Przypisanie roli
    - C. Element współpracujący
  - v. Wystąpienie współpracy (collaboration use)
- 6. Dynamiczne aspekty modelu projektowego
  - I. 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
- II. 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
- III. Diagramy komunikacji a diagramy sekwencji – przekształcanie
- 7. Model analityczny jako opcjonalny etap przejściowy do projektu
  - I. Po co i kiedy warto
  - II. Stereotypy analityczne
    - i. <<boundary>>
    - ii. <<control>>
    - iii. <<entity>>
- 8. Diagram przeglądu interakcji (interaction overview diagram)
  - I. Podstawowe elementy
    - i. Przepływ sterowania
    - ii. Początek
    - iii. Koniec
    - iv. Zakończenie przepływu
    - v. Fragment interakcji
    - vi. Wystąpienie interakcji
  - II. Bloki złożone (combined fragment) a diagram przeglądu interakcji
    - i. Alternatywa
    - ii. Współbieżność
    - iii. Iteracja
- 9. 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. Rozpoczęcie (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. Płytki i głęboka historia (shallow/deep history)
- X. Stany współbieżne
- 10. Diagram harmonogramowania (timing diagram)
  - I. Komunikaty (messages)
  - II. 2 notacje diagramów harmonogramowania
  - III. Przesyłanie komunikatów
- 11. 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
    - A. Po co model wdrożenia
    - B. Artefakt
    - C. Stereotypy artefaktów
      - a. <<file>>
      - b. <<document>>
      - c. <<library>>
      - d. <<executable>>
      - e. <<script>>
      - f. <<source>>
    - D. Specyfikacja konfiguracji (deployment specification)
    - E. Relacje między artefaktami
      - a. Kompozycji (composition)
      - b. Zależności (dependency)
    - F. Instalacja artefaktów (deployment) <<deploy>>
    - G. Manifestacja (manifestation) <<manifest>>
- V. Diagram pakietów (package diagram)
  - i. Pakiet
  - ii. Zagnieżdżanie (nest)
  - iii. Przestrzeń nazw
  - iv. Importowanie (package import)
    - A. <<import>>
    - B. <<access>>
  - v. Łączenie (merge)
- 12. 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

