

Kod szkolenia: **UML2/RP**

Tytuł szkolenia: **Podstawy języka UML2 w realnych projektach**

Dni: 3

## Opis:

### Adresaci Szkolenia:

Szkolenie adresowane jest do osób, które chciałyby poznać podstawy UML2. Przede wszystkim od strony analizy, ale również z wyszczególnieniem różnic między analizą i projektowaniem oraz przeglądem diagramów architektonicznych.

### Cel szkolenia:

Celem szkolenia jest zdobycie od podstaw umiejętności modelowania w notacji UML. Szkolenie skupia się na najważniejszych i najczęściej używanych elementach języka UML od strony analitycznej, ze świadomym pominięciem tych sporadycznie używanych. Dzięki temu maksymalnie skutecznie wykorzystuje się czas na rozwijanie wśród uczestników umiejętności posługiwania się językiem UML. Dla uzupełnienia wiedzy przedstawiane jest także podejście projektowe oraz omówione są diagramy architektoniczne.

W naszej ofercie znajdują się również dłuższe szkolenia z UML, na których kompleksowo omawiamy modelowanie analityczne, projektowe czy architektoniczne (4-5 dni). Natomiast te szkolenie ma na celu przekazanie umiejętności modelowania w ograniczonym czasie (3 dni), stąd też skupia się tylko na najbardziej praktycznych diagramach.

### Wymagania:

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

### Parametry szkolenia:

3\*8 godzin (3\*7 godzin netto) wykładów i warsztatów (z wyraźną przewagą warsztatów). W trakcie warsztatów, oprócz prostych ćwiczeń, analizowany i projektowany jest prosty system informatyczny.

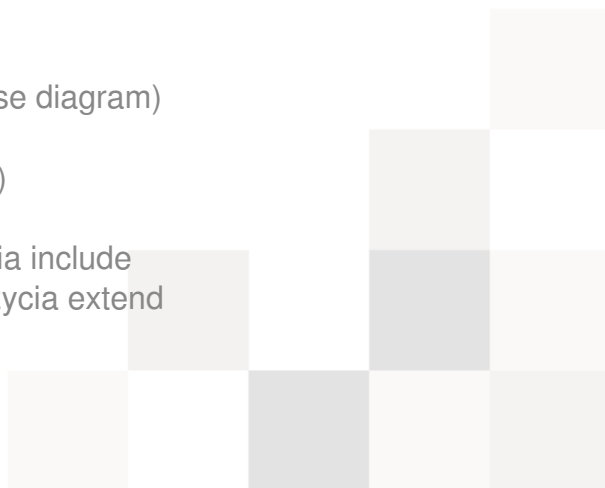
Wielkość grupy: maks. 8-10 osób.

### Program szkolenia:

1. Wprowadzenie do UML



- I. Projektowanie obiektowości
    - i. Klasa (Class)
    - ii. Obiekt (Object)
    - iii. Dziedziczenie (Inheritance)
    - iv. Atrybut (Attribute)
    - v. Metoda (Method)
    - vi. Enkapsulacja (Encapsulation)
    - vii. Polimorfizm (Polymorphism)
    - viii. Przeciążanie metod (Overloading)
    - ix. 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. Komentarze (Note)
    - v. Mechanizmy rozszerzenia
      - A. Stereotypy (Stereotype)
      - B. Etykiety (Tagged Values)
      - C. Ograniczenia (Constraints)
  - VI. Diagram a model UML
2. 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 (dziedziczenie)
  - vii. Granica kontekstu
  - viii. Dokumentowanie scenariusza
3. Modelowanie procesów biznesowych w organizacji
- I. Proces biznesowy
  - 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, decision, merge)
      - B. Przepływy równoległe (fork, join)
    - v. Początek i koniec
      - A. Węzeł początkowy (initial node)
      - B. Węzeł końca aktywności (activity final node)
      - C. Węzeł końca przepływu (flow final node)
    - vi. Partycja (activity partition)
    - vii. Przepływ obiektu (object flow)
      - A. Stan obiektu
      - B. Ograniczenia na obiekcie
    - viii. Sygnały
      - A. Generowanie sygnału (send signal action)
      - B. Odbiór sygnału (accept event action)
      - C. Sygnał czasowy
4. Modelowanie statyczne
- I. Różnice między modelem projektowym a analitycznym
  - II. 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. Przykładowe stereotypy zależności
        - a. instanciate
        - b. 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)
- 5. Tworzenie modelu analitycznego
  - I. Po co i kiedy warto
  - II. Stereotypy analityczne
    - i. boundary
    - ii. control
    - iii. entity
  - III. Zasady modelowania analitycznego
  - IV. 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
    - iv. Bloki złożone (combined fragment)
      - A. Alternatywy - alt
      - B. Pętla - loop
      - C. Współbieżność - par
      - D. Przerwanie - break
    - v. Bloki złożone a notacja w UML 1.X
      - A. Dawniej alternatywy
      - B. Dawniej pętla
      - C. Dawniej współbieżność
- 6. Modelowanie stanowe
  - I. Diagram maszyny stanowej (state machine diagram)
    - i. Stan (state)
    - ii. Stan złożony
    - iii. Przejście (transition)
      - A. Proste
      - B. Automatyczne
      - C. Zwrotne
      - D. Wewnętrzne
    - iv. Rodzaje zdarzeń
      - A. Czasowe (time event)
        - a. Względne (after)
        - b. Bezwzględne (at)



- B. Zdarzenie zmiany stanu (change event - when)
    - C. Wywołanie (call event)
    - D. Sygnał (signal)
    - E. Dowlone (all)
  - v. Sygnatura przejścia
  - vi. Pseudo-stan (pseudostate)
    - A. Ropoczęcie (initial)
    - B. Zakończenie (final)
    - C. Zniszczenie (terminate)
    - D. Wybór (choice)
    - E. Punkt węzłowy (junction)
    - F. Rozwidlenie (fork)
    - G. Złączenie (join)
  - vii. Czynności w stanie
    - A. entry
    - B. do
    - C. exit
    - D. Przejście wewnętrzne
- 7. Podstawy modelu architektonicznego
  - 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. Port i delegacja delegate
  - IV. Diagram wdrożenia (deployment diagram)
    - i. Węzeł (node)
    - ii. Łącze (communication path)
      - A. Łącze kierunkowe
      - B. Liczność łącza
- 8. Przegląd pozostałych diagramów UML (OPCJONALNE)
  - I. Diagram obiektów (Object)
  - II. Diagram struktur połączonych (Composite Structure)
  - III. Diagram pakietów (Package)
  - IV. Diagram harmonogramowania (Timing)
  - V. Diagram komunikacji (Communication)
  - VI. Diagram przeglądu interakcji (Interaction Overview)

