

Kod Szkolenia: UML/BASE

Tytuł Szkolenia: UML2 w realnych projektach

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. Szkolenie skupia się na najważniejszych i najczęściej używanych elementach języka UML2.x, ze świadomym pominięciem tych sporadycznie używanych. Dzięki temu maksymalnie skutecznie wykorzystuje się czas szkolenia na rozwijanie wśród uczestników umiejętności posługiwania się UML. Jednocześnie pominięte tutaj elementy są dokładnie omawiane w szkoleniu UML dla zaawansowanych (UML/ZAAW), które trwa 2 dni - tym samym mają Państwo możliwość poprzestania na najważniejszych elementach języka, lub realizacji pełnego cyklu, który może odbyć się w ciągu jednego tygodnia (3 dni podstawy + 2 dni zaawansowane).

Parametry szkolenia:

3*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:

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)
 - v iii. 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 (navigability)
 - 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