

Opleiding : UML 2, analyse en ontwerp

Praktijkcursus - 4d - 28u00 - Ref. UML
Prijs : 2100 € V.B.

UML is de standaard geworden voor het modelleren van IT-systemen (vooral als de code is georganiseerd volgens het objectparadigma). Na uitleg over de verschillende UML-diagrammen laat deze cursus in praktische termen zien hoe UML in elke fase van een IT-project kan worden gebruikt.

Pedagogische doelstellingen

Aan het einde van de training is de deelnemer in staat om:

- ✓ De verschillende fasen van objectmodellering in UML begrijpen.
- ✓ Begrijpen hoe de verschillende UML-diagrammen worden weergegeven en waarom ze nuttig zijn.
- ✓ Functionele vereisten vertalen met behulp van UML-diagrammen.
- ✓ Ontdek de belangrijkste modelleerbenaderingen.

Doelgroep

Projectmanagers, analisten, ontwerpers, softwarearchitecten en ontwikkelaars die betrokken zijn bij de ontwikkeling van objectgebaseerde projecten.

Voorafgaande vereisten

Basiskennis van software engineering. Ervaring met applicatieanalyse en -ontwerp gewenst.

Praktische modaliteiten

Praktisch werk

Diverse uitgebreide casestudies geven inzicht in de verschillende fasen van objectmodellering in UML. Verschillende omgevingen uit de industrie en de Open Source gemeenschap worden ter demonstratie gepresenteerd.

Opleidingsprogramma

DEELNEMERS

Projectmanagers, analisten, ontwerpers, softwarearchitecten en ontwikkelaars die betrokken zijn bij de ontwikkeling van objectgebaseerde projecten.

VOORAFGAANDE VEREISTEN

Basiskennis van software engineering. Ervaring met applicatieanalyse en -ontwerp gewenst.

VAARDIGHEDEN VAN DE CURSUSLEIDER

De deskundigen die de cursus leiden zijn specialisten op het betreffende vakgebied. Zij werden geselecteerd door onze pedagogische teams zowel om hun vakkennis als hun pedagogische vaardigheden voor elke cursus die zij geven. Zij hebben minstens vijf tot tien jaar ervaring in hun vakgebied en oefenen of oefenden verantwoordelijke bedrijfsfuncties uit.

BEOORDELINGSMODALITEITEN

De cursusleider beoordeelt de pedagogische vooruitgang van de deelnemer gedurende de gehele cursus aan de hand van meerkeuzevragen, praktijksituaties, praktische opdrachten, ...

De deelnemer legt ook van tevoren en naderhand een test af ter bevestiging van de verworven kennis.

1 De Object-benadering

- De belangrijkste concepten van de Object-georiënteerde aanpak begrijpen.
- Herinnering aan programmeerparadigma's (logisch, imperatief, objectgeoriënteerd, enz.).
- Objecten: identiteit, toestand en gedrag. Relaties met de echte wereld en computersystemen.
- Populariteit en voordelen van objectgeoriënteerde software.
- Abstractie, inkapseling, classificatie. Klassen en instanties. Abstracte klassen. Het concept van overerving.
- Methoden en het versturen van berichten tussen objecten. Polymorfisme. Overloading en herdefinitie.

Oefening

Demonstraties van objectgeoriënteerde ontwikkelomgevingen en toepassingen die de behandelde concepten presenteren.

2 Objectanalyse en -ontwerp, inleiding tot UML

- Waarom modelleren? Het spectrum van analyse en ontwerp begrijpen.
- Modelleren van bedrijfsdomeinen en IT-oplossingen. Het model, een centraal artefact in het projectproces.
- Analyse en ontwerp van een IT-oplossing. Impact van programmeertalen.
- Evolutie naar objectanalyse/ontwerp. De voordelen.
- Algemene presentatie van UML. Evolutie en doelstellingen. Standpunten van de architect.
- Het hart van UML: de verschillende diagramtypen (statisch en dynamisch).
- Presentatie van verschillende modelbenaderingen.
- UML-uitbreidingen: stereotypen, profielen, beperkingen, enz.

Praktisch werk

Presentatie van casestudies. Analyse van bedrijfsgebieden.

3 UML use case diagram

- Vastleggen en beschrijven van de functionele vereisten van de applicatie.
- Belangrijkste doelstellingen en gebruik. De functionaliteiten van het systeem beschrijven.
- Diagramelementen: use cases, actoren en systeemgrens. Constructiefasen.
- Hoe identificeer je de belanghebbenden? Hoe beschrijf je een use case? Scenario's.
- Formaten, pre- en postvoorwaarden, relatietypen.

Oefening

Eisen vastleggen en uitdrukken met use case diagrammen. Praktijksituaties.

PEDAGOGISCHE EN TECHNISCHE MIDDELEN

- De gebruikte pedagogische middelen en cursusmethoden zijn voornamelijk: audiovisuele hulpmiddelen, documentatie en cursusmateriaal, praktische oefeningen en correcties van de oefeningen voor praktijkstages, casestudies of reële voorbeelden voor de seminars.
- Na afloop van de stages of seminars verstrekt ORSYS de deelnemers een evaluatievragenlijst over de cursus die vervolgens door onze pedagogische teams wordt geanalyseerd.
- Na afloop van de cursus wordt een presentielijst per halve dag verstrekt, evenals een verklaring van de afronding van de cursus indien de stagiair alle sessies heeft bijgewoond.

TOEGANGSMODALITEITEN EN TERMIJNEN

De inschrijving dient 24 uur voor aanvang van de cursus plaatsgevonden te hebben.

TOEGANKELIJKHEID VOOR MINDERVALIDEN

Is voor u speciale toegankelijkheid vereist? Neem contact op met mevr. FOSSE, contactpersoon voor mindervaliden, via het adres psh-accueil@ORSYS.fr om uw verzoek en de haalbaarheid daarvan zo goed mogelijk te bestuderen.

4 UML-diagrammen voor statische modellering

- Toon een overzicht van het systeem, de onderdelen en hun relaties.
- Klassendiagram: rol, gebruik, voorbeelden.
- Hoe identificeer je nuttige klassen?
- Een klasse in UML: naam, attributen en bewerkingen. Zichtbaarheid (public, private en protected).
- Relaties tussen klassen (associatie, generalisatie, aggregatie en compositie).
- Veelheid, rollen, beperkingen, enz.
- Objectdiagrammen en klasse-instanties.
- Samengesteld structuurdiagram.

Oefening

Verwerf de nodige knowhow in klasseontwerp en objectmodellering door middel van casestudy's.

5 UML-diagrammen voor dynamische modellering

- Systeemevolutie en interacties tussen objecten.
- Sequentiediagram: interacties tussen objecten in de tijd. Synchrone en asynchrone berichten.
- Globaal interactiediagram.
- Communicatieschema: rol van objecten, interacties, verwerkingscompetitie, etc.
- Overgangstatusdiagram: mogelijke toestanden van een object en gebeurtenissen die overgangen triggeren.
- Activiteitendiagram: stroom van activiteiten om een operatie uit te voeren, objecten die verantwoordelijk zijn voor deze activiteiten.
- Tijddiagram: variaties in de tijd.
- Notaties en voorbeelden.

Oefening

Casestudies voltooien door systeemdynamica te modelleren. Context van het gebruik van dynamische diagrammen.

6 Systeemhardware en softwarearchitectuur

- Architectuurmodellen. Organisatie in lagen. Sub-systemen.
- Pakketten en hun relaties.
- Componentendiagram: organisatie van code in modules, afhankelijkheden.
- Inzetdiagram: fysieke inzet van het systeem (machines, netwerken, enz.).

Oefening

Definiëren van de architectuur van de voorgestelde oplossingen. Definitieve nesting van UML-diagrammen.

7 Geavanceerde concepten, hulpmiddelen

- Gegevensmodellen. Object/relatieve mapping. Andere modellen.
- Terugkerende problemen oplossen: Design Patterns (singleton, adapter, enz.) en hun rol.
- Kaders, hergebruik.
- Inleiding tot modelgestuurde ontwikkeling (MDA - MDD).
- Modelleringsworkshops: genereren van rapporten, code, enz. XMI-indeling voor het uitwisselen van UML-modellen tussen AGL's.
- UML en projectmethoden. De verschillende benaderingen. UML integreren.
- Inleiding tot het Unified Process (UP), Y-cyclus, iteraties, fasen en activiteiten.
- Andere mogelijkheden (XP...).

Oefening

De geavanceerde functies van de AGL testen.

Data en plaats

KLAS OP AFSTAND

2026 : 24 maa., 19 mei, 19 mei, 6 okt., 6 okt.,
8 dec., 8 dec.

PARIS LA DÉFENSE

2026 : 19 mei, 6 okt., 8 dec.