Ontwerp oriented object
Ingineering objectsoftware

Praktijkcursus van 4 dagen - 28u
Ref : COB - Prijs 2024 : € 2 390 excl. BTW

Hoe kan men het object oriented ontwerp aanpakken? Hoe kan men overgaan van een functionele benadering naar een objectbenadering? Hoe kan men object oriented programma schrijven met echte schaalbaarheid en herbruikbaarheid? Deze cursus biedt u een conceptuele en praktische beheersing van het objectontwerp.

PEDAGOGISCHE DOELSTELLINGEN

Na afloop van de opleiding kan de cursist:

De principes en specifieke eigenschappen van objectontwerp begrijpen.

Omschakelen van een functionele naar een objectbenadering.

Modelleren van objectsoftware met behulp van UML-notatie.

Vertalen van het UML-model naar objecttaal.

De benaderingen beschrijven met frameworks en componenten.

Leren hoe men de Design Patronen kan implementeren.

HANDS-ON WORK

De behandelde punten zullen worden geïllustreerd met tal van oefeningen die speciaal zijn gekozen om de principes en bijzonderheden van het objectontwerp te benadrukken, van de initiële analyse tot de implementatie in een objecttaal.

DEELNEMERS

Ontwikkelaars, projectleiders die zich willen opleiden in objectgeoriënteerd ontwerp.

VOORAFGAANDE VEREISTEN

Basiskennis van applicatieontwerp en softwareontwikkeling.

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.

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.

HET PROGRAMMA

laatste update: 10/2022

1) Wat kan men verwachten van de Objectbenadering?

- Waarom objecttechnologieën gebruiken?
- De uitdagingen van de nieuwe informatica: modulariteit (plug-ins), herbruikbaarheid, schaalbaarheid.
- Het gebruik van componentenbibliotheken. Hoe beantwoordt de Object-aanpak aan deze uitdagingen?
- Hoe kan men een Object-probleem aanpakken?
- Kennis uit andere vakgebieden van de informatica en andere disciplines.

2) De basisconcepten van de objectbenadering

- De objecten: dualiteit procedure/gegeven.
- Klassen als structuurmodellen en gedrag van objecten, instanties als vertegenwoordigers van klassen.
- De gedefinieerde methoden en procedures in de klassen en gebruikt door de instanties.
- Interacties tussen objecten door het verzenden van berichten. Hoe worden berichten geïnterpreteerd door de objecten?
- De erfenis. Erfenis en typering van variabelen in sterk getypeerde talen (C++, Java).

3) Diagrammen en weergave van objecten met behulp van UML

- De belangrijkste diagrammen (klassendiagrammen, sequentiediagrammen) en hun gebruik voor het objectontwerp.
- De instrumenten voor het beoordelen en weergeven van objecten: toepassing van een modelsysteem van de markt.

4) De grote principes van het objectontwerp

- Wat kan men in de vorm van een object zetten? Principe van reïficatie.
- Criteria die moeten worden toegepast om te beslissen wat in de vorm van een Object moet worden gezet. De te vermijden fouten.
- Hoe kan men een objectsoftware structureren? Principe van modulariteit en ontleding van domeinen.
- Hoe kan men een geheel aan klassen structureren? Abstractie- en classificatiebeginsel.
- Hoe kan men de interactie tussen objecten bedenken? Principe van inkapseling en autonomie.
- Complexe communicatiesystemen analyseren. De algemene aanpak. De te vermijden fouten.
- Toe te passen criteria om over "goede" klassenhiërarchieën te beschikken. De te vermijden fouten.

5) Hoe kan men objectsoftware benaderen?

- De ontwikkelingsprincipes. Van spiraalvormige ontwikkeling tot incrementele ontwikkeling.
- Identificatie van de entiteiten van het domein en beschrijving van de interacties. Hergebruik en schaalbaarheid van programma's.
- Objectontwerp betekent niet het gebruik van een objecttool!
- De te vermijden fouten.

6) Van ontwerp tot implementatie

- UML klasse diagrammen vertalen naar programmeertalen en databases.
- Implementatieprincipes van objectapplicaties. Het belang van de distributie. Algemene client-servermodellen.
- De grote platformen van vandaag: de .NET-technologieën van Microsoft en JEE van SUN.
- Vergelijking van hun sterke en zwakke punten.
- Het belang van de distributie. Klassenbibliotheken. Talen voor het programmeren en gebruiken van componenten.

7) Aanpak met frameworks en componenten

- Het probleem van de levenscyclus van software.
- Ontwikkelings- en onderhoudsproblemen vereisen een softwareaanpak die evolutie mogelijk maakt.
- De frameworks- en componentenbenadering, die gebaseerd is op de Object-gedachte, is een antwoord op deze noodzaak.
- Hoe kan men snel apps ontwerpen en realiseren op basis van frameworks en herbruikbare componenten?
- Hoe kan men softwarecomponenten integreren in een bestaand framework? Hoe kan men frameworks maken?
- Een bestaande applicatie kunnen nemen om deze om te vormen tot framework en zodoende evolutief te maken.
- Grote klassen van frameworks. De huidige componentmodellen.

8) Design patterns

- Hoe kan men de ervaring hergebruiken bij het ontwerpen en ontwikkelen van objectapplicaties?
- Design Patterns of "ontwerppatronen" als softwareoplossingen voortvloeiend uit terugkerende algemene problemen.
- De verschillende soorten Design Patterns. Voorbeeld van Design Patterns.
- Voordelen en beperkingen van Design Patterns.
- Hoe kan men Design Patterns praktisch gebruiken? Leren om Design Patronen te implementeren door de praktijk.

DATA

KLAS OP AFSTAND
2024 : 02 jul, 17 dec

BRUSSEL
2024 : 02 jul, 17 dec