

Design Patterns, implementatie

Praktijkcursus van 5 dagen - 35u

Ref : DES - Prijs 2024 : € 2 970 excl. BTW

Deze cursus traint u in applicatieontwerp en moderne ontwerppraktijken zoals testgedreven ontwikkeling en refactoring. Talloze casestudies leren u hoe u schaalbare en herbruikbare applicaties maakt, rekening houdend met de belangrijkste ontwerp patronen.

PEDAGOGISCHE DOELSTELLINGEN

Na afloop van de opleiding kan de cursist:

De fundamentele principes van objectgeoriënteerd ontwerp begrijpen

De basisregels voor het opdelen van een applicatie in pakketten toepassen

De principes van klassenbouw toepassen in een toepassing

Testgestuurde ontwikkeling in de praktijk brengen

De belangrijkste ontwerp patronen implementeren

HANDS-ON WORK

De workshops die de cursisten uitvoeren, zijn in de taal van hun keuze (Python, Java, C++, C# of VB.Net).

HET PROGRAMMA

laatste update: 04/2024

1) Presentatie ontwerp

- Een herinnering aan de basisprincipes van OOP en UML.
- De voordelen van UML voor ontwerp.
- De uitdagingen van ontwerpen.
- Overerving gebruiken. Voor- en nadelen.

2) Fundamentele principes van object-georiënteerd ontwerp

- De principes van openen/sluiten (OCP) en Liskov substitutie (LSP).
- Concept van polymorfisme, zwakke koppeling en sterke cohesie.
- De impact van objectontwerp op projecten.

Verdeling van verantwoordelijkheden tussen klassen.

3) Principes voor klassenbouw

- Afhankelijkheidsbeheer met afhankelijkheidsinversie (DIP).
- De ogenschijnlijke complexiteit verminderen door interfaces te scheiden (ISP).
- Verdeling van verantwoordelijkheden met GRASP.

4) Principes van pakketorganisatie

- Het pakket: een ontwerpunit voor levering/hergebruik (REP) en gemeenschappelijk hergebruik (CRP).
- De verdeling van pakketten. CCP.
- Organisatie tussen pakketten.

Pakketten samenstellen en prioriteren.

5) Testgestuurde ontwikkeling

- Test Driven Development (TDD) versus Model Driven Engineering (MDE).

DEELNEMERS

Ontwerpers, ontwikkelaars, architecten of projectmanagers.

VOORAFGAANDE VEREISTEN

Kennis van een objectgeoriënteerde taal.

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.

- Schrijven van testgevallen en testsuites.

Verantwoordelijkheden toewijzen aan softwarecomponenten met behulp van de TDD-aanpak.

6) Principes van ontwerppatronen

- Ontwerppatronen voor hergebruik van ervaring.
 - Toepassingsgebied, voordelen en beperkingen van Design Patterns.
 - Reageren op terugkerende problemen.
 - De oprichtingspatronen van Gamma en GoF: creatie-, gedrags- en structuurpatronen.
 - Refactoren. Waarom refactoren?
 - Aanpassing van de codepresentatie en klassenalgoritmen. Herontwerp.
- Voorbeeld van ontwerpen, refactoren en programmeren met GoF-patronen.*

7) Softwarearchitectuur en architectuurpatronen

- Van vereisten naar architectuur.
- Architecturale stijlen.
- Distributiepatronen (Client Server Style, Data Bus Patroon, Blackboard, Repository).
- Systeemontwerppatronen (MVC, gelaagde architectuur, Plug-in stijl, Pipeline).

8) Ontwikkelingsproces

- Ontwerp in een iteratief en incrementeel proces.
- Het Agile-manifest. XP, Scrum.

DATA

Neem contact met ons op