Basisbeginselen van Java en objecten

Praktijkcursus van 5 dagen - 35u
Ref : LJO - Prijs 2024 : € 2 970 excl. BTW

Deze cursus introduceert studenten in de Java-taal en objectprogrammering door geleidelijk alle constructen van de taal te presenteren. De cursus behandelt ook ontwerpproblemen (met behulp van UML-notatie) en de belangrijkste standaardbibliotheken: input/output, collections en grafische interfaces.

PEDAGOGISCHE DOELSTELLINGEN

Na afloop van de opleiding kan de cursist:

De principes van objectgeoriënteerd programmeren toepassen

De syntaxis van de Java-taal onder de knie krijgen

De belangrijkste standaard Java-bibliotheken onder de knie krijgen

Een geïntegreerde ontwikkelomgeving voor programmeren in Java onder de knie krijgen

HANDS-ON WORK

De praktische oefeningen zijn ontworpen om alle elementen van de taal te illustreren en de concepten van objectgeoriënteerd ontwerp te implementeren: alle oefeningen bevatten een analyse/ontwerpfase gevolgd door een programmeerfase.

HET PROGRAMMA

laatste update: 12/2021

1) Objecttechnieken

- De algemene principes van objectmodellering en programmeren.
- Abstractie en inkapseling: interfaces.
- De verschillende vormen van overerving, polymorfisme.
- Inleiding tot UML-modellering: het statische model, het dynamische model, het samenwerkingsmodel, scenario's.

De UML-specificatie van een casestudy, die een van de uitgangspunten zal zijn van de volgende oefeningen.

2) Basis taalconstructies

- Variabelen: declaratie en typering.
- Velden definiëren.
- Methoden: definitie.
- Uitdrukkingen.
- Controle-instructies: voorwaardelijke, lus- en vertakkingsinstructies.
- Tabellen.
- Opgesomde types, autoboxing.
- Compilatie-eenheden en pakketten: controle op zichtbaarheid van klassen, importmechanisme.
- Statische invoer.

Een reeks eenvoudige oefeningen om de ontwikkelomgeving onder de knie te krijgen en een eenvoudig programma te maken. Gebruik van pakketten.

3) Klassen definiëren en instantiëren

- Klassen en objecten.
- Velden en methoden.
- Constructors.

DEELNEMERS

Niet-ervaren ontwikkelaars, ingenieurs, projectmanagers die dicht bij de ontwikkeling staan.

VOORAFGAANDE VEREISTEN

Basiskennis van programmeren. Ervaring met applicatieontwikkeling 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.

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.

- Zelfverwijzing.
- Statische velden en methodes.
- Methoden met een variabel aantal argumenten.
- Methodologische aspecten: klasontwerp.

De casestudy programmeren.

4) De erfenis

- De verschillende vormen van overerving: uitbreiding en implementatie.
- Interfaces en interface-implementatie. Polymorfisme en de implementatie ervan.
- Uitbreiding. Afgeleide klassen, constructors en referenties definiëren. Methodologische aspecten.
- Klassenhiërarchieën bouwen. Codes factoriseren: abstracte klassen.
- Gelijktijdig gebruik van implementatie en uitbreiding. Abstracte klassen.
- Methodologische aspecten: constanten groeperen, diensten specificeren.
- Klassehiërarchieën en interfaces bouwen.
- Algemeenheid.

Het ontwerpen en bouwen van een hiërarchie van klassen en interfaces. Polymorfisme en genericiteit implementeren in de casestudy.

5) Uitzonderingen

- Blokken proberen, uitzonderingen genereren.
- Het selectiealgoritme catch().
- Methodologische aspecten: een uitzonderingshiërarchie opbouwen, uitzonderingen gebruiken.

Introductie van uitzonderingen in de casestudy.

6) Invoer/uitvoer-programmering

- De hiërarchie van invoer-/uitvoerklassen.
- Een paar klassen voor het manipuleren van bestandssystemen.
- Enkele invoer/uitvoer klassen die werken op byte streams en Char streams.
- Toetsenbordinvoer/uitvoer.

Bestanden lezen/schrijven.

7) Grafisch programmeren

- Basisbegrippen: de principes van visualisatie en gebeurtenisbeheer sinds jdk1.1.
- Grafische componenten bekijken: containers en lay-outs.
- Sommige grafische onderdelen: labels, knoppen, tekstzones.
- Gebeurtenisbeheer: Listeners en adapters. Handlers koppelen aan grafische componenten.

Een GUI bouwen.

8) Enkele utiliteitsklassen

- Systeemklassen.
- Containerklassen.

Implementatie van utiliteitsklassen.

DATA

KLAS OP AFSTAND
2025 : 31 mrt, 02 jun, 29 sep, 01 dec

BRUSSEL
2025 : 31 mrt, 02 jun, 29 sep, 01 dec