

Opleiding : Leren programmeren (met Java)

Een beginnersintroductie in de basisprincipes van algoritmische ontwikkeling

Praktijkcursus - 3d - 21u00 - Ref. INJ

Prijs : 1590 € V.B.



4,8 / 5

In deze cursus maak je kennis met klassiek programmeren (van variabelen tot DBMS-toegang) en krijg je een korte inleiding tot objectprogrammering. Veel van de programma's die je zult maken zijn in Java, maar de concepten en methoden die je leert zijn geldig ongeacht de talen die je gebruikt in je toekomstige opdrachten.



Pedagogische doelstellingen

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

- ✓ Programma's structureren volgens een algoritme
- ✓ Het lexicon en de syntaxis van een taal beheersen om een programma te schrijven
- ✓ Een programma compileren en uitvoeren
- ✓ Een programma debuggen en testen
- ✓ Toegang tot een database
- ✓ De belangrijkste principes van objectgeoriënteerd programmeren begrijpen

Doelgroep

Iedereen die moet leren programmeren.

Voorafgaande vereisten

Geen speciale kennis vereist.

Praktische modaliteiten

Praktisch werk

Deze cursus bevat meer dan 60% vereist praktisch werk in Visual Basic (cursus ref INP), Java (cursus ref INJ), C# (cursus ref OGR) of Python (cursus ref THO).

Opleidingsprogramma

DEELNEMERS

Iedereen die moet leren programmeren.

VOORAFGAANDE VEREISTEN

Geen speciale kennis vereist.

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 fundamenteën van programmeren

- Wat is een programma? Wat is een taal? De verschillende paradigma's. Welke taal voor welke toepassing?
- Compilers. Uitvoerbare bestanden.
- De verantwoordelijkheden van een programmeur.
- Wat is een algoritme?
- De behoeften waaraan een algoritme voldoet.
- Het concept pseudotaal.

Praktisch werk

Presentatie van verschillende talen (Java, C#, Visual Basic, C, C++). Schrijven van een eerste algoritme in een pseudotaal.

2 Ontstaan van een eerste programma

- Een eenvoudig programma schrijven: syntaxis en instructies.
- Het programma compileren en uitvoeren.
- Wat is een boekwinkel? Zijn rol, zijn gebruik.

Praktisch werk

De ontwikkel- en uitvoeromgeving ontdekken. Een eerste programma schrijven, compileren en uitvoeren.

3 Programmeerregels

- Naamgevingsconventie.
- Syntaxconventie.
- Commentaar gebruiken. Waarom commentaar geven op ontwikkelingen?
- Verbeter de leesbaarheid van programma's: code inspringen, code uitsplitsen, enz.

4 Variabelen

- Wat is een variabele?
- Waarom een variabele typen?
- Primitieve types: gehele getallen, tekenreeksen, reële getallen, andere.
- Declaratie, definitie en initialisatie van een variabele.
- Constanten.
- Invoer, weergave, toewijzing, typeconversie.
- Organiseer je gegevens in tabelvorm.
- Geavanceerde types: record, matrix, boom.

Praktisch werk

Een aantal eenvoudige programma's schrijven om variabelen te manipuleren.

5 Operatoren en uitdrukkingen

- De verschillende operatoren (vermenigvuldigend, optellend, vergelijking, gelijkheid, logica, toewijzing).
- Combinatie van operatoren.
- Booleaanse uitdrukking.

Praktisch werk

Operatoren en Booleaanse uitdrukkingen verwerken.

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.

6 Besturingsstructuren

- Alternatieve selecties (als, als-dan-niet, case-selectie).
- Instructieblokken (notie van Start... Einde).
- Iteratieve lussen (while-repeat, repeat-until, for-from-to).
- Nesting van instructies.
- De opmerkingen.

Praktisch werk

Gebruik van controlestructuren om een algoritme te implementeren.

7 Procedures en functies

- Definities: procedure, functie.
- Waarom zijn ze essentieel bij het programmeren (herbruikbaarheid, leesbaarheid, enz.)?
- Parameters doorgeven.
- De retourcode van een functie.
- Bewustwording van de grenzen van het doorgeven van de waarde van een variabele.
- Begrip van passeren per adres.
- Functies oproepen.

8 Inleiding tot objectgeoriënteerd programmeren

- Begrippen geassocieerd met Object programmeren: klasse, attribuut, methode, argument.
- Objectmodellering op basis van functionele eisen.
- Inleiding tot goede praktijken in programmaontwerp en -organisatie.

Praktisch werk

Illustratie van objectconcepten.

9 Toegang tot databases

- Organisatie en opslag van gegevens.
- Basisverwerking (verbinding, query's, gegevensherstel).
- Clienttoepassing en gegevensserver.
- Gegevens weergeven en manipuleren in de clienttoepassing.

Praktisch werk

Maak een formulier om informatie te zoeken in een database.

10 Programma onderhouden, debuggen en testen

- Weten hoe je de verschillende foutmeldingen moet lezen en interpreteren.
- Een debugger gebruiken: een programma stap voor stap uitvoeren, breakpoints, variabelen inspecteren tijdens de uitvoering.
- Unit tests plannen.

Praktisch werk

Gebruik van een debugger om de programma-uitvoering te controleren.

Data en plaats