Object programmeren in C++

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

Deze intensieve cursus heeft twee doelen: de deelnemers kennis laten maken met de methoden en reflexen van objectgeoriënteerd programmeren; en een volledige operationele beheersing van de taal C++. De cursus is het resultaat van vele jaren ervaring in C++ en is opgebouwd rond een strenge lesmethode gebaseerd op een groot aantal progressieve praktische oefeningen. Tijdens deze 5 dagen presenteert de trainer de ontwikkelingen van de standaarden van C++98 tot C++20.

PEDAGOGISCHE DOELSTELLINGEN

Na afloop van de opleiding kan de cursist:

Beheers de syntaxis van de taal C++

De concepten van Object-georiënteerd ontwerp implementeren

De ontwikkeltools gebruiken die bij de C++-taal horen

Beheers de belangrijkste toevoegingen aan de C++ 11 standaard

PEDAGOGISCHE METHODEN

Alle oefeningen bevatten een analyse/ontwerpfase gevolgd door een programmeerfase.

HANDS-ON WORK

Werkstations uitgerust met Visual C++ (Windows) en gcc (Unix). De praktische oefeningen zijn ontworpen om alle elementen van de taal te illustreren en de concepten van objectgeoriënteerd ontwerp systematisch te implementeren.

DEELNEMERS

Ontwikkelaars, ingenieurs en projectmanagers die betrokken zijn bij ontwikkeling.

VOORAFGAANDE VEREISTEN

Goede kennis van een programmeertaal zoals C, Java, Python, C#, VB.NET of PHP.

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: 10/2021

1) C++ syntaxis (verschillen tussen C en C++)

- Gegevens: definitie, initialisatie, gegevenstypes.
- Uitdrukkingen: begrip referentie, castmechanismen.
- Operatoren (: :, nieuw, verwijderen).
- Functies (parameters en retourwaarden doorgeven door verwijzing, standaardwaarden, inlining, overbelasting).
- C-code gebruiken in een C++-programma.
- Verwijzingen (argumenten en retourwaarden).
- Constante types.
- Naamruimten.
- Automatisch" typen met het sleutelwoord auto (C++ 11).

De ontwikkelomgeving onder de knie krijgen en een eenvoudig programma programmeren.

2) Objectgeoriënteerde benadering

- De algemene principes van Objet technieken.
- C++ en objectgeoriënteerd programmeren.
- Een inleiding in objectgeoriënteerde methodologieën.
- Een inleiding tot modellen en UML-notatie (statisch model, dynamisch model, samenwerkingsmodel, scenario).

Toepassing van de concepten op een casestudy, een van de hoofdthema's van de volgende oefeningen.

3) C++ klassen en objecten
- Syntax: velden, methoden, constructeurs.
- Toegangscontrole.
- Zelfverwijzing.
- Statische velden en methodes.
- De functies.
- Vriendmethoden en -klassen.
- Dynamisch arrays van objecten maken.
- Methodologische aspecten: klasontwerp.
- Constructeurs voor kopiëren en verplaatsen (C++11).
- Delegatie van constructeurs (C++ 11).
- Inleiding tot geheugenbeheer (stack, heap, garbage collector, enz.).

Programmeren van de casestudy. Het ontwerpen en bouwen van een hiërarchie van klassen en interfaces.

4) Afleiding en overerving
- Afleidingsprincipe.
- Syntax: afgeleide klassen, constructeurs definiëren.
- Toegangscontrole.
- Polymorfisme implementeren: virtuele functies.
- Hergebruik van code: abstracte klassen.
- Interfaces.
- Meervoudige afleiding.
- Semantische en methodologische aspecten: factoring van de code.

De implementatie van polymorfisme in de casestudy.

5) Uitzonderingen
- Syntax: try-blokken, genereren van uitzonderingen.
- Methodologische aspecten: een uitzonderingshiërarchie opbouwen, uitzonderingen gebruiken.

De introductie van uitzonderingen in de casestudy.

6) Overbelasting van de operator
- Principe van overbelasting.
- Binaire operator overbelasting.
- Speciale overload: de index, functie en conversie operator.
- Overloading van geheugenbeheeroperatoren.
- Overloading van '<<' et '>>' operatoren.

Overloading van een paar eenvoudige operatoren.

7) De modellen
- Klassenmodel. Algemene principes en mechanismen. Model overloading en methode herdefinitie.
- Functiemodel. Algemene principes en mechanismen. Model overbelasting.
- Modellen en overbelasting van de operator.
- Modellen en afleidingsmechanismen.
- De verbeteringen die C++ 11 biedt.

Oefeningen op modellen.

8) I/O- en STL-overzicht
- I/O.
- Het principe van streams en de hiërarchie van input/output klassen.
- Beschrijving van enkele invoer-/uitvoerklassen.
- Overzicht van STL.
- Doelstellingen en principes.
- Beschrijvingen van enkele modellen en klassen.
- Containers, iterators, intervalgebaseerde lussen (C++ 11).

9) Conclusie

- Software levenscyclus: testen, integratie, productie release methode.
- Interactie met andere omgevingen.
- Kritische analyse van C++.
- Evolutie van C++.

DATA

KLAS OP AFSTAND
2024 : 24 jun, 23 sep, 16 dec

BRUSSEL
2024 : 24 jun, 23 sep, 16 dec