

Java-programmering: omfattande introduktion - 4 dagar

kurser 471

Du får lära dig att

- Skriva, sammanställa och köra Java-program
- Bygga robusta applikationer med Javas objektorienterade funktioner
- Skapa robusta applikationer med Javas klassbibliotek
- Utveckla plattformsoberoende grafiska användargränssnitt (GUI)
- Läs och skriva data med Java streams
- Hämta data från en relationsdatabas med JDBC

Sammanfattning

Javas unika arkitektur gör att programmerare kan utveckla ett enda program som sedan kan köras sömlöst och pålitligt på flera olika plattformar. Denna kurs ger omfattande erfarenhet av språket Java och dess objektorienterade funktioner. skapa kraftfulla konsoll- och GUI-applikationer samt lagra och hämta data från relationsdatabaser.

Vem bör delta

Denna kurs är värdefull för alla som utvecklar Java-program. Tidigare erfarenhet av ett programmeringsspråk som C, PHP eller COBOL förutsätts. Kännedom om webbt tekniker och objekt-koncept underlättar.

Praktiska övningar

En rad praktiska övningar ger dig erfarenhet av att skapa Java-applikationer. Genom en fallstudie utformar och bygger du en komplicerad skrivbords-applikation som utformats efter en välkänd webbplats. Övningarna omfattar att:

- Utveckla en objektorienterad modell med UML-notation
- Skapa Java-objekt och anropa deras metoder
- Strukturera data med samlings-API i Java
- Skapa portabla GUI med Swing-komponenter
- Lägga till händelsehantering till GUI
- Hämta data från en relationsdatabas med JDBC

Java-programmering: omfattande introduktion - 4 dagar

kurser 471

Introduktion till Java-programmering

Javas fördelar

- Plattformsberoende
- Fristående program och servlets

Ett Java-programs struktur

- Sammanställa källkod till bytecode
- Översikt över klassbibliotek

Objektorienterad programmering med Java

Objektparadigm

- Objektorienterad (OO) programmering
- Inkapsling, arv och polymorfism
- OO-analys och design: "Is a"- och "Has a"-förhållanden
- Designa OO-applikationer steg för steg
- Göra ett diagram av objektstrukturen med Unified Modeling Language (UML)

Javas objektorienterade funktioner

- Skapa objekt från klasser
- Aggregation och sammansättning
- Utöka befintliga klasser
- Överlagra och åsidosätta metoder

Java-språkets struktur

Språkets syntax

- Deklarera och initiera variabler
- Programsatser och uttryck
- Deklarera och använda vektorer
- Upcasting, downcasting och autoboxing

Flödeskontroll

- Anropa metoder och överföra parametrar
- Villkorsuttryck och slingor
- Hantera undantag med **try** och **catch**

Definiera klasser

- Fält (instansdata)
- Metoder (funktioner)
- Abstrakta klasser och gränssnitt
- Organisera klasser med paket (packages) och synlighetsmodifierare
- Sammansättning kontra arv

Bygga komponenterna i ett Java-program

- Arbeta med befintliga klasser
- Dra nytta av generiska komponenter med samlingarna API
- Utöka basklasserna
- Utveckla nya klasser
- Sammanställa och avlusa

Utveckla GUI

Grundläggande om användargränssnitt

- Grundläggande GUI-komponenter
- Händelsedrivna programmering

Java är ett varumärke som ägs av Sun Microsystems, Inc.

- Fördelar med portabelt bibliotek för fönsterhantering

Java foundation classes (jfc)

- Fördelar med lättvikts Swing-komponenter
- Utforska Swing-komponentbiblioteket
- Skapa Swing-komponenter: knappar, textfält, rullgardinsmenyer
- Lägga till Swing-komponenter till behållare
- Arrangera Swing-komponenter med layouthanterare
- Dialoger och meddelanderutor

Händelsehantering

- Registrera händelsehanterare
- Inre klasser och toppklasser

Fil I/O och databaser

Java streams

- Streams, Readers och Writers
- Åtkomst till filer
- Formatera text

Filer och kataloger

- Skapa, radera och döpa om filer
- Erhålla katalog- och filinformation

åtkomst av databas med JDBC

- Ansluta till en databas
- Sända SQL-satser
- Hämta och bearbeta data

Utvecklingsverktyg för Java

- Java Development Kit (JDK)
- Kompilator
- Java IDE:s