Java 8 Programming & Object Orientated Essentials for Developers New to OO (TT2120-J8)

Varighed: 5 Days **Kursus Kode: GK0391** **Leveringsmetode: Company event (Firmakursus)**

Beskrivelse:

This course is geared for developers who have little or no prior working knowledge of object-oriented (OO) programming languages (such as C, COBOL, 4GL, and so forth). Throughout the course, you will learn the best practices for writing great OO programs in Java 8, using sound development techniques, improved features for better performance, and new capabilities for addressing rapid application development. Special emphasis is placed on OO concepts and best practices.

Firmakursus

Med et firmakursus bliver jeres it-kompetenceudvikling målrettet jeres behov. Det betyder, at vi hjælper med at finde og sammensætte det helt rigtige kursusindhold og den helt rigtige form. Kurset kan afvikles hos os eller kunden, standard eller virtuelt.

Målgruppe:

Developers who wish to use Java or who need to reinforce sound OO Java coding practices

Agenda:

- OO programming and its advantages in today's world
- Objects, classes, and OO implementations
- Basic concepts of OO such as encapsulation, inheritance, polymorphism, and abstraction
- Fundamentals of the Java language, its importance, uses, strengths, and weaknesses
- How Java language relates to OO programming and the object model
- Use classes, inheritance and polymorphism
- Use collections, generics, autoboxing, and enumerations
- Use Lambda expressions
- Use the stream application programming interface (API)
- Discover the new date/time API
- Use the Java Database Connectivity (JDBC) API for database access
- Work with annotations
- Java tooling
- **Specific Java 8 features that are covered in the course include:**
- The new date/time API
- Lambda expressionsMethod and constructor references
- The stream API
- Collectors
- The optional class

Forudsætninger:

- Practical experience working with procedural or non-OO languages such as C, COBOL, 4GL, or mainframe

Indhold:

1. Java 8: A First Look

- The Java Platform
- Java Platforms
- Lifecycle of a Java Program
- Responsibilities of JVM
- Documentation and Code Reuse
- Using the JDK
- Setting up Environment
- Locating Class Files
- Compiling Package Classes
- Source and Class Files
- Java Applications
- The Eclipse Paradigm
- Workbench and Workspace
- Views
- Editors
- Perspectives
- Projects

2. OO Concepts

- Object-Oriented Programming
- Real-World Objects
- Classes and Objects
- Object Behavior
- Methods and Messages
- Inheritance, Abstraction, and Polymorphism
- Encapsulation
- Inheritance
- Method Overriding
- Polymorphism

3. Getting Started with Java

- Writing a Simple Class
- Classes in Java
- Class Modifiers and Types
- Class Instance Variables
- Primitives vs. Object References
- Creating Objects
- Adding Methods to the Class
- Passing Parameters into Methods
- Returning a Value from a Method
- Overloaded Methods
- Constructors
- Optimizing Constructor Usage
- Language Statements
- Operators
- Comparison and Logical Operators
- Looping
- Continue and Break Statements
- The Switch Statement
- The For-Each() Loop
- For-Each Loop Restrictions
- Using Strings
- Strings
- String Methods
- String Equality
- StringBuffer
- StringBuilder
- Java 8: String Joiner
- Specializing in a Subclass

4. Essential Java Programming

- Fields and Variables
- Instance vs. Local Variables: Usage Differences
- Data Types
- Default Values
- Block Scoping Rules
- Final and Static Fields
- Static Methods
- Using Arrays
- Arrays
- Accessing the Array
- Multidimensional Arrays
- Copying Arrays
- Variable Arguments
- Java Packages and Visibility
- Class Location of Packages
- The Package Keyword
- Importing Classes
- Executing Programs
- Java Naming Conventions

5. Advanced Java Programming

- Inheritance and Polymorphism
- Polymorphism: The Subclasses
- Upcasting vs. Downcasting
- Calling Superclass Methods from Subclass
- The Final Keyword
- Interfaces and Abstract Classes
- Separating Capability from Implementation
- Abstract Classes
- Implementing an Interface
- Abstract Classes vs. Interfaces
- Exceptions
- Exception Architecture
- Handling Multiple Exceptions
- Automatic Closure of Resources
- Creating Your Own Exceptions
- Throwing Exceptions
- Checked vs. Unchecked Exceptions

6. Java Developer's Toolbox

- Utility Classes
- Wrapper Classes
- The Number Class
- Random Numbers
- Autoboxing/Unboxing
- The Date Class
- Enumerations and static imports
- Enumeration Syntax
- When You Should Use Enumerations
- Using Static Imports
- When You Should Use Static Imports
- Java 8 - The Date/Time API
- The Core Date/Time classes
- Formatting Dates
- Working with Time Zones
- Manipulate Date/Time Values

7. Collections and Generics

- Generics
- Generics and Subtyping
- Bounded Wildcards
- Generic Methods
- Legacy Calls to Generics
- When Generics Should Be Used
- Collections
- Characterizing Collections
- Collection Interface Hierarchy
- Iterators
- The Set Interface
- The List Interface
- Queue Interface
- Map Interfaces
- Using the Right Collection
- Collections and Multithreading

8. Lambda and Streams

- Java 8: Introduction to Lambda
- Functional vs OO Programming
- Anonymous Inner-classes
- Lambda Expression Syntax
- Functional Interfaces
- Method references
- Constructor references
- Java 8: Streams
- Processing Collections of data
- The Stream interface
- Reduction and Parallelism
- Filtering collection data
- Sorting Collection data
- Map collection data
- Find elements in Stream
- Numeric Streams
- Create infinite Streams
- Sources for using Streams
- Java 8: Collectors
- Creating Collections from a Stream
- Group elements in the Stream
- Multi-level grouping of elements
- Partitioning Streams

9. Java Application Development

- Introduction to Annotations
- Annotations Overview
- Working with Java Annotations
- JDBC
- Connecting to the Database
- Statement and PreparedStatement
- ResultSet
- Executing Inserts, Updates, and Deletes
- Controlling Transactions and Concurrency

- Extending a Class
- Casting
- The Object Class
- Default Constructor
- Implicit Constructor Chaining

Additional Information:

This course is about 50% hands-on lab and 50% lecture with extensive programming exercises designed to reinforce core development skills, concepts, and best practices learned in the lessons. Our courses include ample materials and labs to ensure all students are either appropriately challenged, or assisted, at all times-no matter their skill level.

Flere Informationer:

For yderligere informationer eller booking af kursus, kontakt os på tlf.nr.: 44 88 18 00

training@globalknowledge.dk

www.globalknowledge.com/da-dk/

Global Knowledge, Stamholmen 110, 2650 Hvidovre