Java SE 21: Programming II Training

Duration: **5 Day**

BİLGİNÇ IT Academy
info@bilginc.com
bilginc.com

Overview

Based on the **Official Oracle Curriculum**, this is the second-level {*training} for developers learning the Java language. It builds upon the foundations laid in the "Java SE: Programming I" course and covers advanced topics such as object-oriented programming (OOP), modular architecture, secure coding, database integration, use of Lambdas, and developing multithreaded applications.

This course uses the current version of Java:**Java SE 21**.

m-slice="1 1 []">Based on the **Official Oracle Curriculum**, this is the second-level course for developers learning the Java language. It builds upon the foundations laid in the "Java SE: Programming I" course and covers advanced topics such as object-oriented programming (OOP), modular architecture, secure coding, database integration, use of Lambdas, and developing multithreaded applications.

This course uses the current version of Java:**Java SE 21**.

Prerequisites

Completion of **Java SE 21: Programming I** or equivalent experience

Who Should Attend

- Java Developers
-
    - Java Developers
    - Data Scientists
    - Application Developers working with Java-based platforms
    - Engineers aiming to upgrade their skills with modern Java features
- Data Scientists
- Application Developers working with Java-based platforms
- Engineers aiming to upgrade their skills with modern Java features

Java SE 21: Programming II Training

Duration: **5 Day**

BİLGİNÇ IT Academy
info@bilginc.com
bilginc.com

What You Will Learn

By the end of this {*training}, you will have gained knowledge and skills in the following areas:

- Develop Java applications using object-oriented principles such as encapsulation, inheritance, and polymorphism
- Execute Java applications from the command line
- Work effectively with the Java Collections Framework
- Use Lambda expressions to filter and process collections
- Manage exceptions with robust exception handling strategies
- Perform file input/output operations
- Manage file systems using the JDK NIO.2 API
- Perform CRUD operations using JDBC and JPA technologies
- Implement parallel operations and concurrency using Lambda expressions
- Apply modular programming practices for scalable application architecture
- Build high-performance, multithreaded applications

Outline

1. Functional Programming

- Lambda expressions and Functional Interfaces
- Built-in functional interfaces: `Predicate`, `Consumer`, `Function`, etc.
- Stream API usage: `stream()`, `filter()`, `map()`
- Writing clean and readable functional code using Java Streams

2. Modular Programming

- Fundamentals of Java's module system (JPMS)
- `module-info.java` syntax and structure
- Defining modules and services in applications
- Migrating legacy projects to a modular structure

3. Stream and Parallel Stream Operations

BİLGİNÇ IT Academy

info@bilginc.com

bilginc.com

- Understanding **Parallel Streams** and their advantages
- Enhancing performance with parallel processing
- Terminal operations using Collectors
- Creating and managing custom streams

4. Concurrency and Multithreading

- Writing multi-threaded applications
- Using ExecutorService, Callable, and Runnable
- Thread synchronization and handling race conditions

5. Java API & Secure Coding Practices

- Core I/O fundamentals: InputStream, OutputStream, Reader, Writer
- Modern file operations with **NIO.2**: Path, Files, DirectoryStream
- Best practices for writing secure Java code

6. Database Applications (JDBC & JPA)

- Connecting to databases using JDBC API
- Preventing SQL injection with PreparedStatement
- Using JPA for Object-Relational Mapping (ORM)
- Executing CRUD operations with hands-on examples

7. Localization

- Implementing locale-based features with Locale, ResourceBundle, DateFormat, and NumberFormat
- Building applications that support multiple languages

8. Collections and Generics

- Overview of Java Collections Framework: List, Set, Map, Queue
- Using Iterator and Iterable interfaces
- Type safety with **Generics**
- Working with Wrapper classes (Integer, Double, Boolean, etc.)

9. Exception Handling and Assertions

Java SE 21: Programming II Training

Duration: **5 Day**

BİLGİNÇ IT Academy
info@bilginc.com
bilginc.com

- Difference between checked and unchecked exceptions
- Creating and handling custom exceptions
- Using try, catch, finally, throw, and throws
- Validating assumptions with assert