

# OJAV-18 - JAVA SE 21 PROGRAMMING COMPLETE

Categoria: **Java**

## INFORMAZIONI SUL CORSO



Durata:  
5 Giorni



Categoria:  
Java



Qualifica Istruttore:  
Oracle Certified  
Professional



Dedicato a:  
Sviluppatore



Produttore:  
Oracle

## OBIETTIVI

This course provides an in-depth coverage for all core Java topics and most frequently used APIs. It also offers a set of practical exercises teaching how to build a fully functioning Java application from scratch.

## PREREQUISITI

This course is intended for students with some programming experience.

## CONTENUTI

### 1 - Introduction to Java

- Objectives 1-2
- What Is Java? 1-3
- How Java Works 1-4
- Object-Oriented Principles 1-5
- Classes and Objects 1-6
- Classes 1-7
- Objects 1-8
- Inheritance 1-9
- Java APIs 1-10
- Java Keywords, Reserved Words, and Special Identifiers 1-11
- Java Naming Conventions 1-12
- Java Basic Syntax Rules 1-13
- Defining a Java Class 1-14
- Accessing Classes Across Packages 1-15
- Implementing Encapsulation with Access Modifiers 1-17
- Creating a Main Application Class 1-18
- Compiling a Java Program 1-19
- Executing a Java Program 1-20
- Comments and Documentation 1-21
- Code Snippets in Javadoc 1-23
- External Snippets 1-24

- Summary 1-25
- Practices for Lesson 1: Overview 1-26

## **2 - Primitive Types, Operators, and Flow Control Statements**

- Objectives 2-2
- Java Primitives 2-3
- Declaring and Initializing Primitive Variables 2-4
- Primitive Declarations and Initializations: Restrictions 2-5
- Java Operators 2-6
- Assignment and Arithmetic Operators 2-7
- Arithmetic Operations and Type Casting 2-8
- More Mathematical Operations 2-9
- Binary Number Representation 2-10
- Bitwise Logical Operators 2-11
- Equality, Relational, and Conditional Operators 2-12
- Short-Circuit Evaluation 2-13
- Flow Control Using if/else Construct 2-14
- Ternary Operator 2-15
- Flow Control Using switch Construct 2-16
- Switch -> No Fall-Through Syntax 2-17
- Switch Expressions yield a Value 2-18
- Switch Statements and Expressions Summary 2-19
- Using JShell (REPL Tool) 2-20
- JShell Commands 2-21
- Code Snippets 2-22
- Summary 2-23
- Practices for Lesson 2: Overview 2-24

## **3 - Text, Date, Time, and Numeric Objects**

- Objectives 3-2
- String Initialization 3-3
- String Operations 3-4
- String Indexing 3-5
- Mutable Text Objects 3-6
- Text Blocks 3-7
- Spaces, Lines, and Quotes 3-9
- Wrapper Classes for Primitives 3-10
- Representing Numbers Using BigDecimal Class 3-11
- Method Chaining 3-12
- Local Date and Time API 3-13
- More Local Date and Time Operations 3-14
- Instants, Durations, and Periods 3-16
- Zoned Date and Time 3-18
- Representing Languages and Countries 3-19
- Formatting and Parsing Numeric Values 3-21
- Formatting and Parsing Date and Time Values 3-23
- Localizable Resources 3-26
- Formatting Message Patterns 3-27

- Formatting and Localization: Summary 3-29
- Summary 3-31
- Practices for Lesson 3: Overview 3-32

#### **4 - Classes and Objects**

- Objectives 4-2
- UML: Introduction 4-3
- Modeling Classes 4-4
- Modeling Interactions and Activities 4-6
- Designing Classes 4-7
- Creating Objects 4-8
- Defining Instance Variables 4-9
- Defining Instance Methods 4-10
- Object Creation and Access: Example 4-11
- Local Variables and Recursive Object Reference 4-12
- Local Variable Type Inference 4-14
- Defining Constants 4-16
- Static Context 4-17
- Accessing Static Context 4-18
- Combining Static and Final 4-19
- Other Static Context Use Cases 4-20
- IntelliJ IDE: Introduction 4-21
- Summary 4-22
- Practices for Lesson 4: Overview 4-23

#### **5 - Improved Class Design**

- Objectives 5-2
- Overload Methods 5-3
- Variable Number of Arguments 5-4
- Defining Constructors 5-5
- Reusing Constructors 5-6
- Access Modifiers: Summary 5-7
- Defining Encapsulation 5-8
- Defining Immutability 5-9
- Constants and Immutability 5-10
- Enumerations 5-11
- Complex Enumerations 5-13
- Java Memory Allocation 5-14
- Parameter Passing 5-15
- Java Memory Cleanup 5-16
- Summary 5-17
- Practices for Lesson 5: Overview 5-18

#### **6 - Implement Inheritance and Use Records**

- Objectives 6-2
- Extending Classes 6-3
- Object Class 6-4
- Reusing Parent Class Code Through Inheritance 6-5

- Instantiating Classes and Accessing Objects 6-6
- Rules of Reference Type Casting 6-7
- Verifying Object Type Before Casting the Reference 6-8
- Pattern Matching for instanceof 6-9
- Reference Code Within the Current or Parent Object 6-10
- Defining Subclass Constructors 6-11
- Class and Object Initialization: Summary 6-12
- Overriding Methods and Using Polymorphism 6-13
- Reusing Parent Class Logic in Overwritten Method 6-15
- Defining Abstract Classes and Methods 6-16
- Defining Final Classes and Methods 6-17
- Sealed Classes and Interfaces 6-18
- Overriding Object Class Operations: toString 6-20
- Overriding Object Class Operations: equals 6-21
- Override Object Class Operations: hashCode 6-22
- Comparing String Objects 6-23
- Java Records 6-24
- Custom Record Constructors 6-25
- Record Patterns 6-26
- Pattern Matching for switch 6-27
- Factory Methods 6-28
- Summary 6-29
- Practices for Lesson 6: Overview 6-30

## 7 - Interfaces and Generics

- Objectives 7-2
- Java Interfaces 7-3
- Multiple Inheritance Problem 7-5
- Implement Interfaces 7-6
- Default, Private, and Static Methods in Interfaces 7-7
- Interface Hierarchy 7-8
- Default Methods Inheritance 7-9
- Interface Is a Type 7-10
- Functional Interfaces 7-11
- Generics 7-12
- Use Generics 7-13
- Examples of Java Interfaces: java.lang.Comparable 7-15
- Examples of Java Interfaces: java.util.Comparator 7-16
- Examples of Java Interfaces: java.lang.Cloneable 7-17
- Composition Pattern 7-18
- Summary 7-19
- Practices for Lesson 7: Overview 7-20

## 8 - Arrays and Loops

- Objectives 8-2
- Arrays 8-3
- Combined Declaration, Creation, and Initialization of Arrays 8-4
- Multidimensional Arrays 8-5

- Copying Array Content 8-6
- Arrays Class 8-7
- Loops 8-8
- Processing Arrays by Using Loops 8-9
- Complex for Loops 8-10
- Embedded Loops 8-11
- Break and Continue 8-12
- Summary 8-13
- Practices for Lesson 8: Overview 8-14

## **9 - Collections**

- Objectives 9-2
- Introduction to Java Collection API 9-3
- Java Collection API Interfaces 9-4
- Java Collection API Implementation Classes 9-5
- Java Collection API Interfaces and Implementation Classes 9-6
- Create List Object 9-7
- Manage List Contents 9-8
- Create Set Object 9-9
- Manage Set Contents 9-11
- Create Deque Object 9-12
- Manage Deque Contents 9-13
- Create HashMap Object 9-15
- Manage HashMap Contents 9-16
- Iterate Through Collections 9-17
- Sequenced Collections 9-18
- Other Collection Behaviors 9-20
- Use java.util.Collections Class 9-21
- Access Collections Concurrently 9-22
- Prevent Collections Corruption 9-23
- Legacy Collection Classes 9-25
- Summary 9-26
- Practices for Lesson 9: Overview 9-27

## **10 - Nested Classes and Lambda Expressions**

- Objectives 10-2
- Types of Nested Classes 10-3
- Static Nested Classes 10-6
- Member Inner Classes 10-7
- Local Inner Classes 10-9
- Anonymous Inner Classes 10-10
- Anonymous Inner Classes and Functional Interfaces 10-12
- Understand Lambda Expressions 10-13
- Define Lambda Expression Parameters and Body 10-14
- Use Method References 10-15
- Default and Static Methods in Functional Interfaces 10-16
- Use Default and Static Methods of the Comparator Interface 10-17
- Use Default and Static Methods of the Predicate Interface 10-18

- Summary 10-19
- Practices for Lesson 10: Overview 10-20

## 11 - Java Streams API

- Objectives 11-2
- Characteristics of Streams 11-3
- Create Streams Using Stream API 11-4
- Stream Pipeline Processing Operations 11-5
- Using Functional Interfaces 11-6
- Primitive Variants of Functional Interfaces 11-7
- Bi-argument Variants of Functional Interfaces 11-9
- Perform Actions with Stream Pipeline Elements 11-10
- Perform Filtering of Stream Pipeline Elements 11-11
- Perform Mapping of Stream Pipeline Elements 11-12
- Join Streams Using flatMap Operation 11-13
- Other Intermediate Stream Operations 11-14
- Short-Circuit Terminal Operations 11-15
- Process Stream Using count, min, max, sum, average Operations 11-16
- Aggregate Stream Data using reduce Operation 11-17
- General Logic of the collect Operation 11-19
- Using Basic Collectors 11-20
- Perform a Conversion of a Collector Result 11-21
- Perform Grouping or Partitioning of the Stream Content 11-22
- Mapping and Filtering with Respect to Groups or Partitions 11-23
- Parallel Stream Processing 11-25
- Parallel Stream Processing Guidelines 11-26
- Restrictions on Parallel Stream Processing 11-27
- Splitter 11-28
- Splitter Characteristics 11-29
- Summary 11-30
- Practices for Lesson 11: Overview 11-31

## 12 - Exception Handling, Logging, and Debugging

- Objectives 12-2
- Using Java Logging API 12-3
- Logging Method Categories 12-4
- Guarded Logging 12-5
- Log Writing Handling 12-6
- Logging Configuration 12-7
- Describe Java Exceptions 12-8
- Create Custom Exceptions 12-9
- Throwing Exceptions 12-10
- Catching Exceptions 12-11
- Exceptions and the Execution Flow 12-12
- Helpful NullPointerExceptions 12-13
- Example Throwing an Unchecked Exception 12-14
- Example Throwing a Checked Exception 12-15
- Handling Exceptions 12-16

- Resource Auto-Closure 12-17
- Suppressed Exceptions 12-18
- Handle Exception Cause 12-19
- Java Debugger 12-20
- Debugger Actions 12-21
- Manipulate Program Data in Debug Mode 12-22
- Validate Program Logic Using Assertions 12-23
- Normal Program Flow with No Exceptions 12-24
- Program Flow Producing a Runtime Exception 12-26
- Program Flow Catching Specific Checked Exception 12-28
- Program Flow Catching Any Exceptions 12-30
- Summary 12-32
- Practices for Lesson 12: Overview 12-33

### **13 - Java IO API**

- Objectives 13-2
- Java Input-Output Principals 13-3
- Java Input-Output API 13-4
- Reading and Writing Binary Data 13-5
- Basic Binary Data Reading and Writing 13-6
- Reading and Writing Character Data 13-7
- Basic Character Data Reading and Writing 13-8
- Connecting Streams 13-9
- Standard Input and Output 13-10
- Using Console 13-11
- Understand Serialization 13-13
- Serializable Object Graph 13-14
- Object Serialization 13-15
- Serialization of Sensitive Information 13-16
- Customize Serialization Process 13-17
- Serialization and Versioning 13-18
- Working with Filesystems 13-19
- Constructing Filesystem Paths 13-20
- Navigating the Filesystem 13-21
- Analyze Path Properties 13-22
- Set Path Properties 13-23
- Create Paths 13-24
- Create Temporary Files and Folders 13-25
- Copy and Move Paths 13-26
- Delete Paths 13-27
- Handle Zip Archives 13-28
- Represent Zip Archive as a FileSystem 13-29
- Access HTTP Resources 13-31
- Summary 13-32
- Practices for Lesson 13: Overview 13-33

### **14 - Java Concurrency and Multithreading**

- Objectives 14-2

- Java Concurrency Concepts 14-3
- Implement Threads 14-4
- Thread Life Cycle 14-5
- Interrupt Thread 14-6
- Block Thread 14-7
- Make Thread Wait Until Notified 14-8
- Common Thread Properties 14-10
- Create Executor Service Objects 14-11
- Manage Executor Service Life Cycle 14-14
- Implementing Executor Service Tasks 14-16
- Locking Problems 14-18
- Writing Thread-Safe Code 14-19
- Ensure Consistent Access to Shared Data 14-21
- Nonblocking Atomic Actions 14-22
- Ensure Exclusive Object Access Using Intrinsic Locks 14-23
- Intrinsic Lock Automation 14-24
- Nonblocking Concurrency Automation 14-25
- Alternative Locking Mechanisms 14-26
- CPU Versus IO Bound Concurrent Tasks 14-27
- Virtual Threads API 14-28
- Virtual Thread Operations 14-29
- Summary 14-30
- Practices for Lesson 14: Overview 14-31

## **15 - Modules and Deployment**

- Objectives 15-2
- Compile, Package, and Execute Nonmodular Java Applications 15-3
- Nonmodular Java Characteristics 15-4
- What Is a Module? 15-5
- Java Modules 15-7
- Java Module Categories 15-8
- Define Module Dependencies 15-9
- Export Module Content 15-10
- Module Example 15-11
- Open Module Content 15-12
- Open an Entire Module 15-13
- Produce and Consume Services 15-14
- Services Example 15-15
- Multi-Release Module Archives 15-16
- Compile and Package a Module 15-17
- Execute a Modularized Application 15-18
- Migrating Legacy Java Applications Using Automatic module 15-19
- Create Custom Runtime Image 15-20
- Execute Runtime Image 15-22
- Optimize a Custom Runtime Image 15-23
- Check Dependencies 15-24
- Summary 15-25
- Practices for Lesson 15: Overview 15-26



## **A - Annotations**

- Objectives A-2
- Annotations: Introduction A-3
- Design Annotations A-4
- Apply Annotations A-5
- Dynamically Discover Annotations A-6
- Document the Use of Annotations A-7
- Annotations that Validate Design A-8
- Deprecated Annotation A-9
- Suppress Compiler Warnings A-10
- Varargs and Heap Pollution A-11
- Summary A-12

## **B - Java Database Connectivity**

- Objectives B-2
- Java Database Connectivity (JDBC) B-3
- JDBC API Structure B-4
- Manage Database Connections B-5
- Create and Execute Basic SQL Statements B-6
- Create and Execute Prepared SQL Statements B-7
- Create and Execute Callable SQL Statements B-8
- Process Query Results B-9
- Control Transactions B-11
- Discover Metadata B-12
- Customize ResultSet B-13
- Set Up ResultSet Type B-14
- Set Up ResultSet Concurrency and Holdability B-16
- Summary B-17

## **C - Java Security**

- Objectives C-2
- Security as Nonfunctional Requirement C-3
- Security Threats C-4
- Denial of Service (DoS) Attack C-5
- Define Security Policies C-6
- Changes in the Security API C-8
- Secure File System and I/O Operations C-9
- Best Practices for Protecting Your Code C-10
- Erroneous Value Guards C-11
- Protect Sensitive Data (Part 1) C-12
- Protect Sensitive Data (Part 2) C-13
- Prevent SQL Injections C-14
- Prevent JavaScript Injections C-15
- Prevent XML Injections C-16
- Discover and Document Security Issues C-17
- Summary C-18

## D - Advanced Generics

- Objectives D-2
- Compiler Erases Information About Generics D-3
- Generic and Raw Type Compatibility D-4
- Generics and Type Hierarchy D-5
- Wildcard Generics D-6
- Upper Bound Wildcard D-7
- Lower Bound Wildcard D-8
- Collections and Generics Best Practices D-9
- Summary D-10

## E - Java Applications on Oracle Cloud

- Objectives E-2
- Cloud Application Requirements E-3
- Cloud Application Runtime Infrastructure E-4
- Cloud Java Application Servers E-5
- Package and Deploy Cloud Application E-6
- Optimise deployment with GraalVM E-7
- HTTP Protocol Basics E-9
- REST Service Conventions and Resources E-11
- Configure and Launch REST Service Application Using Helidon SE E-12
- Summary E-13

## F - Miscellaneous Java Topics

- Objectives F-2
- Builder Design Pattern F-3
- Singleton Design Pattern F-4
- Java Regular Expression API F-5
- Regular Expressions: Character Classes F-6
- Regular Expressions: Quantifiers F-7
- Regular Expressions: Boundaries F-8
- File IO Watch Service F-9
- Summary F-11

## INFO

**Materiale didattico:** Materiale didattico ufficiale Oracle in formato digitale

**Costo materiale didattico:** incluso nel prezzo del corso a Calendario

**Natura del corso:** Operativo (previsti lab su PC)