

# Java SE 21 Programmer

**Nro. Horas: 45**

**Costo para Profesionales: S/ 1000.00 (incluido IGV)**

**Costo para Estudiantes (no egresados): S/ 500.00 (incluido IGV)**

Este es un taller de preparación para la certificación Java SE 21 Developer de Oracle. Este examen consta de **50 preguntas**, una duración de **90 minutos** y se necesita mínimo un **68% de aprobación**. Es de múltiple elección.

## PROPUESTA DEL CURSO

- Clases grabadas online vía zoom.
- Simulacros por tópico
- Simulacros tipo examen de 50 preguntas
- Ejercicios y material por cada clase.

## TOPICOS JAVA 17

Esos son los tópicos que se estudiarán en el taller:

### Handling date, time, text, numeric and boolean values

- ✓ Use primitives and wrapper classes including Math API, parentheses, type promotion, and casting to evaluate arithmetic and boolean expressions
- ✓ Manipulate date, time, duration, period, instant and time-zone objects using Date-Time API
- ✓ Manipulate text, including text blocks, using String and StringBuilder classes

### Controlling Program Flow

- ✓ Create program flow control constructs including if/else, switch statements and expressions, loops, and break and continue statements

### Utilizing Java Object-Oriented Approach

- ✓ Declare and instantiate Java objects including nested class objects, and explain the object life-cycle including creation, reassigning references, and garbage collection
- ✓ Implement inheritance, including abstract and sealed classes. Override methods, including that of Object class. Implement polymorphism and differentiate object type versus reference type. Perform type casting, identify object types using instanceof operator and pattern matching
- ✓ Create classes and records, and define and use instance and static fields and methods, constructors, and instance and static initializers
- ✓ Create and use interfaces, identify functional interfaces, and utilize private, static, and default interface methods
- ✓ Implement overloading, including var-arg methods
- ✓ Create and use enumerations with fields, methods and constructors
- ✓ Understand variable scopes, use local variable type inference, apply encapsulation, and make objects immutable

### Handling Exceptions

- ✓ Handle exceptions using try/catch/finally, try-with-resources, and multi-catch blocks, including custom exceptions

Working with Arrays and Collections

- ✓ Create Java arrays, List, Set, Map, and Deque collections, and add, remove, update, retrieve and sort their elements

Working with Streams and Lambda expressions

- ✓ Use Java object and primitive Streams, including lambda expressions implementing functional interfaces, to supply, filter, map, consume, and sort data
- ✓ Perform decomposition, concatenation and reduction, and grouping and partitioning on sequential and parallel streams

Packaging and deploying Java code and use the Java Platform Module System

- ✓ Define modules and their dependencies, expose module content including for reflection. Define services, producers, and consumers
- ✓ Compile Java code, produce modular and non-modular jars, runtime images, and implement migration using unnamed and automatic modules

Managing concurrent code execution

- ✓ Create worker threads using Runnable and Callable, manage the thread lifecycle, including automations provided by different Executor services and concurrent API
- ✓ Develop thread-safe code, using different locking mechanisms and concurrent API
- ✓ Process Java collections concurrently including the use of parallel streams

Using Java I/O API

- ✓ Read and write console and file data using I/O Streams
- ✓ Create, traverse, read, and write Path objects and their properties using java.nio.file API
- ✓ Serialize and de-serialize Java objects

Accessing databases using JDBC

- ✓ Create connections, create and execute basic, prepared and callable statements, process query results and control transactions using JDBC API

Implementing Localization

- ✓ Implement localization using locales, resource bundles, parse and format messages, dates, times, and numbers including currency and percentage values

Assume the following:

- ✓ Missing package and import statements: If sample code do not include package or import statements, and the question does not explicitly refer to these missing statements, then assume that all sample code is in the same package, or import statements exist to support them.
- ✓ No file or directory path names for classes: If a question does not state the file names or directory locations of classes, then assume one of the following, whichever will enable the code to compile and run:
  - ✓ All classes are in one file
  - ✓ Each class is contained in a separate file, and all files are in one directory
- ✓ Unintended line breaks: Sample code might have unintended line breaks. If you see a line of code that looks like it has wrapped, and this creates a situation where the wrapping is significant (for example, a quoted String literal has wrapped), assume that the wrapping is an extension of the same line, and the line does not contain a hard carriage return that would cause a compilation failure.
- ✓ Code fragments: A code fragment is a small section of source code presented without its context. Assume that all necessary supporting code exists and that the supporting environment fully supports the correct compilation and execution of the code shown and its omitted environment.
- ✓ Descriptive comments: Take descriptive comments, such as "setter and getters go here," at face value. Assume that correct code exists, compiles, and runs successfully to create the described effect.

Candidates are also expected to:

- ✓ Understand the basics of Java Logging API.
- ✓ Use Annotations such as Override, FunctionalInterface, Deprecated, SuppressWarnings, and SafeVarargs.
- ✓ Use generics, including wildcards.

## TOPICOS JAVA 21

Esos son los tópicos que se estudiarán en el taller:

1. Virtual Threads – JEP 444

2. Sequenced Collections – JEP 431

3. Record Patterns – JEP 440
4. Pattern Matching for switch – JEP 441
5. New Methods in String, StringBuilder, StringBuffer, and Character
6. Preview and Incubator Features
7. Other Changes in Java 21

**Prerrequisitos:**

- Conocimientos de Fundamentos de programación

**Formas de Pago:**

Transferencia Bancaria:

**INTERBANK**  
**CUENTA CORRIENTE SOLES**  
**200-3001605915**  
**CCI: 003-200-003001605915-33**

**Yape o PLIN:**

- Al +51 966 025 115 (Miryan Ramírez Ortega)

Pago con Tarjeta de Débito y Crédito

- Solicitar el pago con tarjeta de débito y crédito al +51 966 025 115 (WhatsApp).

**Consultas:**

Si tiene alguna consulta o comentario nos puede escribir al WhatsApp al +51 966 025 115.