

Java Programming I y II

Detalles del curso:

Duración: 25 horas

Modalidad: Videoconferencia

Examen: Oficial

Información e inscripción:

685 457 610 - 615 844 193

formacion@laberit.com

LÄBERIT

CENTRO DE
FORMACIÓN TIC.

Curso de Java Programming I y II

Introducción

El presente curso oficial de Oracle, diseñado para otorgar a los participantes una base sólida en el uso de componentes nativos del API de Java SE 8, constituye el segundo paso en el camino hacia el éxito como desarrollador de aplicaciones en la plataforma Java.

Los asistentes aprenderán a diseñar aplicaciones robustas orientadas a objetos en la plataforma Java, a la vez que conocen y ponen en práctica los componentes nativos claves del API de Java SE 8: Collections, Generics, JDBC, Localization, Java NIO.2, The Fork-Join Framework, ...

Dicho curso se enmarca dentro del programa Workforce Development Program (WDP), iniciativa diseñada por Oracle para ayudar a la nueva generación de profesionales TI a aprender las habilidades más demandadas sobre tecnologías Oracle, así como ayudarles en su preparación a los exámenes de certificación asociados al curso, suponiendo un valor añadido y diferenciador en su perfil profesional.

¿A quién va dirigido?

Este curso está dirigido a desarrolladores con experiencia en el uso del lenguaje de programación Java, que desean profundizar en el uso de componentes nativos del API de Java.

También pueden estar interesados en realizar este curso, aquellos desarrolladores ya familiarizados con los componentes principales del API nativo de Java, que se están preparando para obtener la certificación OCP Java SE 8 Programmer y desean profundizar en algunos contenidos.

Objetivos didácticos

Una vez finalizado el curso, el alumno habrá adquirido los conocimientos y habilidades para:

- Diseñar aplicaciones en la plataforma Java usando las APIs nativas más conocidas.
- Poner en práctica los conocimientos sobre orientación a objetos.
- Identificar buenas prácticas para diseñar aplicaciones robustas.
- Utilizar y diseñar expresiones Lambda.
- Almacenar y manipular datos usando colecciones.
- Interactuar con archivos, directorios y el sistema de ficheros.
- Conectar con bases de datos utilizando consultas SQL estándar mediante JDBC.

Curso de Java Programming I y II

- Diseñar aplicaciones multi-thread con alto rendimiento.

Requisitos previos

Para un máximo aprovechamiento del curso, los participantes deberán poseer conocimientos del paradigma de programación orientado a objetos (tanto a nivel teórico como práctico) y experiencia en el uso del lenguaje de programación Java.

Programa

1. Descripción general de la plataforma Java

- Definición de cómo el lenguaje Java logra la independencia de la plataforma
- Diferenciar entre las plataformas Java ME, Java SE y Java EE
- Evaluación de opciones de bases de datos, middleware y bibliotecas de Java
- Definiendo cómo sigue evolucionando el lenguaje Java
- Revisión de clase y sintaxis de Java

2. Creando clases simples de Java

- Creando variables primitivas
- Usando operadores
- Crear y manipular cadenas
- Uso de declaraciones if-else y switch
- Iterando con bucles: while, do-while, for, mejorado para
- Creando arreglos
- Usar campos, constructores y métodos de Java

3. Encapsulación y subclasificación

- Usando la encapsulación en el diseño de clases de Java
- Modelado de problemas comerciales utilizando clases de Java
- Haciendo las clases inmutables
- Creación y uso de subclases de Java
- Métodos de sobrecarga

Curso de Java Programming I y II

- Métodos primordiales, polimorfismo y clases estáticas

4. Uso de niveles de acceso: privado, protegido, predeterminado y público

- Métodos primordiales
- Usando la invocación de método virtual
- Usar varargs para especificar argumentos variables
- Usar el operador instanceof para comparar tipos de objetos
- Usar lances hacia arriba y hacia abajo
- Modelado de problemas comerciales mediante el uso de la palabra clave estática
- Clases abstractas y anidadas

5. Diseñar clases base de propósito general mediante el uso de clases

abstractas

- Construcción de clases y subclases abstractas de Java
- Aplicar la palabra clave final en Java
- Distinguir entre clases de nivel superior y anidadas
- Interfaces y expresiones Lambda

6. Definición de una interfaz Java

- Elegir entre herencia de interfaz y herencia de clase
- Ampliando una interfaz
- Métodos de incumplimiento
- Clases internas anónimas
- Definición de una expresión Lambda
- Colecciones y genéricos

7. Crear una clase genérica personalizada

- Usando el diamante de inferencia de tipo para crear un objeto
- Creando una colección usando genéricos
- Implementación de ArrayList
- Implementar un TreeSet
- Implementando un HashMap

- Implementing a Deque
- Pedido de colecciones
- Colecciones, flujos y filtros

8. Describiendo el patrón Builder

- Iterando a través de una colección usando la sintaxis lambda
- Descripción de la interfaz de Stream
- Filtrar una colección usando expresiones lambda
- Llamar a un método existente usando una referencia de método
- Encadenando múltiples métodos juntos
- Definición de pipelines en términos de lambdas y colecciones
- Interfaces funcionales integradas de Lambda

9. Listado de las interfaces integradas incluidas en java.util.function

- Interfaces principales: predicado, consumidor, función, proveedor
- Usar versiones primitivas de interfaces base
- Usar versiones binarias de interfaces base
- Operaciones Lambda

10. Extraer datos de un objeto usando mapa

- Describir los tipos de operaciones de arroyos
- Describiendo la clase opcional
- Describir el procesamiento diferido
- Ordenar una secuencia
- Guardar resultados en una colección mediante el método de recopilación
- Agrupar y particionar datos usando la clase Collectors
- Excepciones y afirmaciones

11. Definición del propósito de las excepciones de Java

- Uso de las cláusulas catch, multi-catch y finalmente
- Cerrar automáticamente los recursos con una declaración de prueba con recursos
- Reconocer clases y categorías de excepciones comunes
- Creando excepciones personalizadas

- Prueba de invariantes mediante el uso de aserciones
- API de fecha / hora de Java

12. Crear y administrar eventos basados en fechas

- Crear y administrar eventos basados en el tiempo
- Combinar fecha y hora en un solo objeto
- Trabajar con fechas y horas en distintas zonas horarias
- Gestión de cambios resultantes del horario de verano
- Definición y creación de marcas de tiempo, períodos y duraciones
- Aplicar formato a fechas y horas locales y por zonas

13. I / O Fundamentals

- Describir los conceptos básicos de entrada y salida en Java
- Leer y escribir datos desde la consola
- Usar secuencias para leer y escribir archivos
- Escribir y leer objetos mediante serialización
- E / S de archivos (NIO.2)

14. Usar la interfaz de ruta para operar en rutas de archivos y directorios

- Usar la clase Archivos para verificar, eliminar, copiar o mover un archivo o directorio
- Uso de Stream API con NIO2
- Concurrencia

15. Describir la programación de tareas del sistema operativo

- Crear subprocesos de trabajo usando Runnable y Callable
- Usar un ExecutorService para ejecutar tareas al mismo tiempo
- Identificación de posibles problemas de subprocesos
- Uso de atómico sincronizado y concurrente para administrar la atomicidad
- Usar bloqueos de monitor para controlar el orden de ejecución del hilo
- Usando las colecciones java.util.concurrent
- El marco de unión en bifurcación

16. Paralelismo

- La necesidad de Fork-Join
- Robo de trabajo
- RecursiveTask

17. Secuencias paralelas

- Revisión de las características clave de las corrientes
- Describir cómo hacer que una canalización de flujo se ejecute en paralelo
- Enumere las suposiciones clave necesarias para utilizar una canalización paralela
- Definición de reducción
- Describir por qué la reducción requiere una función asociativa
- Calcular un valor usando reducir
- Describir el proceso para descomponer y luego fusionar el trabajo.
- Listado de las consideraciones clave de rendimiento para transmisiones paralelas
- Aplicaciones de base de datos con JDBC

18. Definición del diseño de la API de JDBC

- Conexión a una base de datos mediante un controlador JDBC
- Enviar consultas y obtener resultados de la base de datos
- Especificación externa de la información del controlador JDBC
- Realización de operaciones CRUD utilizando la API de JDBC
- Localización

19. Describir las ventajas de localizar una aplicación

- Definiendo lo que representa una localidad
- Leer y establecer la configuración regional mediante el objeto Locale
- Creación de un paquete de recursos para cada configuración regional
- Llamar a un paquete de recursos desde una aplicación
- Cambiar la configuración regional de un paquete de recursos

Acreditado:

Microsoft **Imagine Academy**
Program Member

Fundación Estatal
PARA LA FORMACIÓN EN EL EMPLEO 

LABORA
Servei Valencià d'Ocupació i Formació

CertiProf® | Partner

 **Pearson**
VUE
Authorised
Test Centre


A PEARSON VUE BUSINESS
AUTHORIZED TESTING CENTER