

Lenguaje de programación Java™

Duration: 5 Days

What you will learn

El curso Java Programming Language (Programación Java) ofrece a los estudiantes las bases necesarias para programar en Java. Estos son algunos de los temas que se incluyen: información sobre la sintaxis del lenguaje de programación Java; programación orientada a objetos con el lenguaje de programación Java; creación de interfaces gráficas de usuario (GUI), excepciones, subprocessos y entrada y salida (E/S) de archivos, e información sobre redes. Quienes estén familiarizados con los conceptos de la programación orientada a objetos podrán aprender a desarrollar aplicaciones de tecnología Java. El curso se basa en la plataforma Java Platform, Standard Edition 6 (Java SE 6) y requiere el uso del producto Java SE Development Kit 6 (JDK 6). Los estudiantes llevan a cabo los ejercicios de laboratorio del curso con el entorno de desarrollo integrado (IDE) de NetBeans. Estudiantes para los que se ha diseñado el curso: Programadores interesados en sumar el lenguaje de programación Java a su lista de cualificaciones.

* Estudiantes que se están preparando para el examen Oracle Certified Professional, Java SE 6 Programmer (Profesional certificado por Oracle, programador de Java SE 6).

Related Training

Required Prerequisites

Saber crear programas con algún lenguaje de programación o haber asistido al curso SL-110-SE6.

Saber crear y editar archivos de texto con un editor de texto.

Curso Fundamentals of the Java Programming Language (Programación Java para no programadores).

Comprender los principios de la programación orientada a objetos.

Suggested Prerequisites

Fundamentals of the Java Programming Language, Java SE 6

Fundamentos del lenguaje de programación Java™

Course Objectives

Crear aplicaciones de tecnología Java que hagan uso de las características orientadas a objetos del lenguaje Java, como la encapsulación, la herencia y el polimorfismo.

Crear un cliente de red simple que use el protocolo de control de transmisión/protocolo de Internet (TCP/IP) para comunicarse con un servidor mediante sockets.

Crear un cliente de red simple que use el protocolo de control de transmisión/protocolo de Internet (TCP/IP) para comunicarse con un servidor mediante sockets.

Crear una interfaz gráfica de usuario (GUI) controlada por eventos con componentes Swing: paneles, botones, etiquetas, campos de texto y áreas de texto.

Crear programas multiproceso.

Ejecutar una aplicación de tecnología Java desde la línea de comandos.

Implementar técnicas de control de errores mediante el control de excepciones.

Implementar funciones de entrada/salida (E/S) para leer y escribir archivos de texto y datos y comprender los flujos de E/S avanzados.

Usar expresiones y tipos de datos de tecnología Java.

Usar estructuras de control de flujo de tecnología Java.

Usar arreglos y otras colecciones de datos.

Course Topics

Introducción

Examinar la tecnología Java

Analizar una aplicación simple de tecnología Java

Ejecutar una aplicación de tecnología Java

Programación orientada a objetos

Definir los conceptos de modelado: abstracción, encapsulación y paquetes

Examinar la reutilización del código de las aplicaciones de tecnología Java

Definir clase, miembro, atributo, método, constructor y paquete

Usar los modificadores de acceso private y public según las pautas de encapsulación

Invocar un método de un objeto específico

Usar la documentación en línea de la API de la tecnología Java

Identificadores, palabras clave y tipos

Usar comentarios en un programa de origen

Distinguir entre identificadores válidos y no válidos

Usar los ocho tipos primitivos

Definir valores literales para tipos textuales y numéricos

Definir los términos variable primitiva y variable de referencia

Declarar variables de tipo de clase

Construir un objeto mediante el uso de new y describir la inicialización predeterminada

Describir la importancia de una variable de referencia

Expresiones y control de flujo

Distinguir entre variables de instancia y locales

Describir la forma de inicializar variables de instancia

Reconocer, describir y usar operadores de software de Java

Distinguir entre asignaciones válidas y no válidas de tipos primitivos

Identificar expresiones booleanas y sus requisitos en construcciones de control

Reconocer la compatibilidad de asignaciones y los moldeos necesarios en tipos fundamentales

Usar las construcciones if, switch, for, while y do, y las formas con etiqueta de las instrucciones break y continue, como €

Arreglos

Declarar y crear arreglos de tipos primitivos, de clase o de array
Explicar por qué se inicializan los elementos de un arreglo
Explicar cómo se inicializan los elementos de un arreglo
Determinar la cantidad de elementos de un arreglo
Crear un arreglo multidimensional o matriz
Escribir un código para copiar valores de un arreglo a otro

Diseño de clases

Definir herencia, polimorfismo, sobrecarga, anulación e invocación de métodos virtuales
Usar el modificador de acceso predeterminado (package-friendly) y el modificador de acceso protected
Describir los conceptos de sobrecarga de métodos y constructores
Describir por completo la operación de construcción e inicialización de objetos

Características avanzadas de clases

Crear variables, métodos e inicializadores estáticos
Crear clases, métodos y variables finales
Crear y usar tipos enumerados
Usar la instrucción de importación estática
Crear clases y métodos abstractos
Crear y usar una interfaz

Excepciones y aserciones

Definir el concepto de excepciones
Usar las instrucciones try, catch y finally
Describir las categorías de excepciones
Identificar las excepciones comunes
Desarrollar programas para controlar las excepciones
Usar aserciones
Distinguir el uso correcto o incorrecto de las aserciones
Habilitar aserciones en tiempo de ejecución

Marcos genéricos y de colecciones

Describir las implementaciones de objetivos generales de las interfaces centrales del marco de las colecciones
Examinar la interfaz Map
Examinar las clases de colecciones antiguas
Crear ordenaciones personalizadas y naturales mediante la implementación de las interfaces Comparable y Comparator
Usar colecciones genéricas y parámetros de tipo en clases genéricas
Refactorizar el código no genérico existente
Escribir un programa para iterar sobre una colección
Examinar el bucle for mejorado

Fundamentos de E/S

Escribir un programa que use propiedades del sistema y argumentos de línea de comandos
Examinar la clase Properties
Construir secuencias de procesamiento y nodos y usarlas de forma adecuada
Serializar y deserializar objetos
Distinguir flujos de lectores y escritores y seleccionar uno de ellos de forma adecuada

Entrada y salida (E/S) de consola y de archivo

- Leer los datos de la consola
- Escribir datos en la consola
- Describir archivos y E/S de archivos

Creación de una GUI en Java mediante la API Swing

- Describir la tecnología Swing de JFC
- Identificar los paquetes Swing
- Describir los bloques de creación de GUI: administradores de diseño, componentes y contenedores
- Examinar las propiedades de nivel superior, de propósito general y de propósito especial de los contenedores
- Examinar los componentes
- Examinar los administradores de diseño
- Describir el modelo de subproceso único de Swing
- Crear una GUI utilizando los componentes Swing

Control de eventos generados por la GUI

- Definir eventos y control de eventos
- Analizar el modelo de eventos de Java SE
- Describir el comportamiento de la GUI
- Determinar la acción del usuario que originó un evento
- Desarrollar receptores de eventos
- Describir la simultaneidad de las GUI basadas en Swing y describir las características de la clase SwingWorker

Aplicaciones basadas en GUI

- Describir el modo de crear una barra de menús, un menú y los elementos del menú en una GUI de Java
- Aprender a cambiar el color y la fuente de un componente

Subprocesos

- Definir el concepto de subproceso
- Crear subprocesos separados en un programa de tecnología Java y controlar el código y los datos que usan los subprocesos
- Controlar la ejecución de un subproceso y escribir un código multiplataforma con subprocesos
- Describir las dificultades que pueden surgir cuando varios subprocesos comparten datos
- Usar wait y notify para que haya comunicación entre los subprocesos
- Usar synchronized para evitar la corrupción de datos

Redes

- Desarrollar código para configurar una conexión de red
- Comprender TCP/IP
- Usar las clases ServerSocket y Socket para implementar clientes y servidores TCP/IP