

# JAVA FUNDAMENTALS



## LO QUE APRENDERÁ

El curso Conceptos fundamentales de Java SE 7 se ha diseñado para permitir a los estudiantes con poca o ninguna experiencia en programación empezar a conocer la programación mediante el lenguaje de programación Java. El curso enseña la importancia de la programación orientada a objetos, las palabras clave y las construcciones del lenguaje de programación Java, así como los pasos necesarios para crear programas simples de tecnología Java. Los estudiantes que realicen este curso pueden recibir una sólida base en lenguaje de programación Java en la que basar su trabajo y formación futura.

- Utilizar distintas construcciones de lenguaje de programación Java para crear varias aplicaciones de tecnología Java
- Utilizar construcciones de decisión y bucle y métodos para establecer el flujo de programa
- Realizar el manejo de errores básico para programas de tecnología Java
- Implantar conceptos intermedios de programación Java y orientados a objetos en programas de tecnología Java
- Demostrar los conocimientos sobre la tecnología Java y el lenguaje de programación Java

## REQUISITOS PREVIOS

- Conocimientos básicos de al menos un lenguaje de programación
- La capacidad de seguir las instrucciones de instalación de software e instalar Alice, Greenfoot y Eclipse en un ordenador

## OBJETIVOS DEL CURSO

- Analizar un problema de negocio para reconocer los objetos y las operaciones que forman los bloques integrantes del diseño de programas Java
- Declarar un método con argumentos y valores de retorno
- Declarar e inicializar variables
- Declarar e instanciar matrices y ArrayLists y poder iterar con ellas
- Definir el término
- Mostrar la sintaxis de programación Java
- Describir ejemplos de cómo se utiliza Java en aplicaciones, así como en productos de consumo
- Describir cómo se manejan los errores en un programa Java
- Describir las ventajas de utilizar un entorno de desarrollo integrado (IDE)
- Desarrollar clases y describir cómo declarar una clase
- Instanciar un objeto y utilizar de forma eficaz variables de referencia de objetos
- Enumerar y describir varias características clave de la tecnología Java
- Enumerar varios tipos de datos primitivos
- Utilizar la herencia para declarar y definir una subclase de una superclase existente
- Utilizar operadores, bucles y construcciones de decisión
- Escribir un programa Java simple que se compile y ejecute correctamente

## ◆ Introducción a la tecnología Java

- Relación de Java con otros lenguajes
- Demostración de cómo descargar, instalar y configurar el entorno Java en un sistema Windows
- Descripción de las distintas tecnologías Java como Java EE, JavaME, Embedded Java SE
- Descripción de las funciones clave de la tecnología y las ventajas de utilizar Java
- Uso de un entorno de desarrollo integrado (IDE)

## ◆ Consideraciones sobre los objetos

- Definición del dominio de problemas
- Identificación de objetos y reconocimiento de los criterios para definir objetos

## ◆ Introducción al lenguaje Java

- Definición de clases
- Identificación de los componentes de una clase
- Creación y uso de una clase de prueba
- Compilación y ejecución de un programa de prueba

## ◆ Trabajo con variables primitivas

- Declaración e inicialización de variables de campo
- Descripción de tipos de dato primitivos como integral, punto flotante, textual y lógico
- Declaración de variables y asignación de valores
- Uso de constantes
- Uso de operadores aritméticos para modificar valores

## ◆ Trabajo con objetos

- Declaración e inicialización de objetos
- Almacenamiento de objetos en memoria
- Uso de referencias de objetos para manipular datos
- Uso de documentación Java de JSE para buscar los métodos de una clase
- Trabajo con objetos String y StringBuilder



## ◆ Uso de operadores y construcciones de decisión

- Uso de operadores relacionales y condicionales
- Prueba de igualdad entre cadenas
- Evaluación de distintas condiciones en un programa y determinación del algoritmo
- Creación de construcciones if e if/else
- Anidamiento y encadenamiento de sentencias condicionales
- Uso de una sentencia switch

## ◆ Creación y uso de matrices

- Declaración, instanciación e inicialización de una matriz unidimensional
- Declaración, instanciación e inicialización de una matriz bidimensional
- Uso de un bucle for para procesar una matriz
- Creación e inicialización de una ArrayList
- Uso de la sentencia import para trabajar con API de Java existentes
- Acceso a un valor de una matriz o ArrayList
- Uso de la matriz args

## ◆ Uso de construcciones de bucle

- Creación de bucles while y bucles while anidados
- Desarrollo de un bucle for
- Uso de ArrayLists con bucles for
- Desarrollo de un bucle do while
- Descripción del ámbito de variable

## ◆ Trabajo con métodos y sobrecarga de métodos

- Creación y llamada de un método
- Transferencia de argumentos y devolución de valores
- Creación de métodos estáticos y variables estáticas
- Uso de modificadores
- Sobrecarga de un método



## ◆ Uso de encapsulación y constructores

- Creación de constructores
- Implantación de encapsulación

## ◆ Introducción a conceptos orientados a objetos avanzados

- Uso de herencia
- Uso de tipos de polimorfismo como sobrecarga, sustitución y enlace dinámico
- Trabajo con superclases y subclasses
- Adición de abstracción al análisis y diseño
- Descripción del objetivo de las interfaces Java
- Creación e implantación de una interfaz Java

## ◆ Manejo de errores

- Descripción de los diferentes tipos de errores que se pueden producir y cómo se manejan en Java
- Descripción de los diferentes tipos de excepciones en Java
- Uso de documentación Java para buscar las excepciones devueltas por los métodos de las clases base
- Escritura de código para manejar excepciones

## ◆ Estructura general

- Creación de paquetes y archivos JAR para el despliegue mediante Java
- Arquitecturas de dos y tres niveles
- Algunos ejemplos de aplicaciones Java

