

Curso Java JSE

```
/*
 * www.DCINTERNET.com.mx
 */
package mx.com.dcinternet.com.dao;

import java.util.Map;

/**
 *
 * @author DCINTERNET
 */
public interface IDAO<I, T> {

    int inserta(T t);

    Map<I, T> consultaTodos();

    static void insertaS() {
```

Descripción: Nuestro curso *Programación Java JSE* ofrece el curso de este nivel más completo, sin duda, en donde aprenderás el Java, JSE desde cero pero sin que te quedes en cero. De hecho aprenderás temas que otros pueden considerar intermedios o avanzados como JDBC, colecciones y *threads*

Abarca desde qué es Java, cómo instalar lo que necesitas para empezar a codificar en Java, cómo se utiliza la sintaxis básica, cómo se crean clases, cómo usamos las clases ya existentes, la definición y aplicación en casos reales de la programación orientada a

objetos, colecciones, manejo de excepciones, acceso a archivos del sistema, es decir, el lenguaje Java como tal, pero vamos más allá.

Algo muy característico de nuestro curso es que no sólo aprenderás a programar en Java sino a programar “correctamente” con el uso de patrones, buenas prácticas y siempre bajo las recomendaciones y estándares de Java.

Al final del curso abordamos un tema esencial en cualquier desarrollo Java, JDBC, la API que te permitirá comunicarte con una base de datos relacional usando el lenguaje Java, aquí aprenderás a obtener conexiones y hacer operaciones CRUD ejecutando sentencias SQL desde Java a la base de datos. La base de datos que usamos en nuestros cursos es Oracle pero para grupos se puede usar SQL Server.

El curso cubre algunos temas como el manejo de fechas y tiempos con las clases `LocalDate` y `LocalTime`, así como introducción a la API Stream y a las expresiones Lambda.

Objetivo:

El propósito primordial de este entrenamiento es “aprender Java”, conocer las reglas de sintaxis del lenguaje y, sobre todo conocer, y aplicar las características de los lenguajes orientado a objetos como lo es la abstracción, encapsulación, herencia y polimorfismo. Llevamos más de 11 años formando profesionistas con este perfil.

Aprenderás a crear tus propias clases y también a usar las clases ya existentes de Java SE.

Conocerás las nuevas características de Java que hace al lenguaje más fácil de usar pues tiene mejoras en la sintaxis y también más ágil en términos de rendimiento por las mejoras que trae en la máquina virtual.

Aprenderás a utilizar hilos o *threads*, una de las características que de distinguen a Java de otros lenguajes.

Duración: 32 horas (en línea)

Requisitos: Ser del área de sistemas. Conocimientos de principios de programación, saber que es una variable, un ciclo o conocer otro lenguaje de programación

Contenido

Introducción a Java

- .¿Qué es Java?
 - . ¿Qué tipo de aplicaciones hacemos con Java?
 - . JSE vs JEE
- .JSE vs JEE
- . JDK vs JRE
- . Máquina Virtual
- .Entornos de desarrollo
 - .Netbeans o Eclipse
- .Un primer ejercicio
 - . static public void para tu primer ejercicio
 - .Compilación tu primer ejercicio

Fundamentos de Java

- .Estructura de un programa Java
 - .Conceptos
 - .El estilo de programación Java
 - .Comentarios
 - .Variables y Tipos de Datos
- .Variables
 - .Tipos de Dato
 - .Declaraciones de Variables
 - .Inicialización de variables
- .Conversiones entre tipos
 - .Casting
- .Operadores
- .Estructuras de Control de Flujo
 - .Ejecución condicional
 - .La estructura if y la estructura switch
 - .Ejecución iterativa
 - .La estructura for y la estructura while

Programación Orientada a Objetos

- .Programación Orientada a Objetos
 - .¿Qué es la programación?
 - .¿Qué es la POO?
 - .¿Cuáles son los beneficios de la POO?
- .Características de la POO
 - .Abstracción
 - .Encapsulación
 - .Herencia
 - .Polimorfismo
- .Objetos
 - .Definición
- .Relaciones entre Objetos
 - .Relaciones "Es un": Herencia
 - .Relaciones "Tiene un": Composición
 - .Relaciones "Usa un": Colaboración

Manejo de Objetos

- .Variables y Tipos de Dato Objeto
 - .Variables Objeto
 - .Tipos de Dato Objeto
 - .Declaraciones de Variables Objeto
 - .Inicialización de variables
 - .Accesando los elementos del objeto
- .Creación de Objetos
 - .El operador new
 - .Constructores
- .Manipulación de Variables Objeto
 - .Asignación de Variables Objetos
 - .Comparación entre Variables

Paquetes

- ¿Qué son los paquetes?
- Uso de paquetes
- Organización de Paquetes

Arreglos

- Arreglos
- Arreglos Multidimensionales
- Creación de Arreglos a partir de Literales
- Argumentos de línea de comando
- Ciclo for-each

Clases LocalDate y LocalTime

- Paquetes
- Uso de paquetes
- La clase LocalDate
- La clase LocalTime
- Arreglos
- Arreglos Multidimensionales
- Creación de Arreglos a partir de Literales
- Argumentos de línea de comando
- Ciclo for-each

Creación de nuevas clases

- .Creación de nuevas clases
 - .Las clases como aplicaciones ejecutables
 - .Reglas Generales
 - .Nomenclatura de Clases
 - .Estructura Básica
 - .Declaración de Variables
 - .Constantes
 - .Nomenclatura de variables y constantes
 - .Declaración de Métodos
 - .Nomenclatura de métodos
 - .Declaración de Constructores
 - .El constructor por defecto
 - .Constructores Especiales
- .Sobrecarga de métodos
- .Destrucción de Objetos
- .El Garbage Collector

Elementos Instancia y Elementos Clase

- .Elementos Instancia y Elementos Clase
 - .Elementos Instancia
 - .Referencia genérica: `this`
 - .Elementos Clase
 - .El modificador *static*
 - .Inicializador estático
 - .Interacción entre elementos de clase y de instancia

record
o clases

- .El nuevo tipo record
- .Instancia de record
- .Constructor de record

Herencia

- .Implementando Herencia
 - .La terminología de Herencia
 - .Herencia Múltiple
 - .La Clase Object
 - .La instrucción extends ...
 - .Herencia y Variables Objeto
 - .Herencia y asignación de variables
 - .Argumentos de los métodos
 - .Referencias genéricas: *super*
 - .Llamando a un método sobre escrito
 - .Llamando a un constructor del antecesor
- .Modificadores de Acceso
- .Encapsulación en Java
- .Sobre Escritura de Métodos
 - .Sobre escribiendo métodos de Object

Clases Abstractas e Interfaces

- Clases abstractas
 - .El modificador *abstract*
 - .Modificadores de Herencia
 - .Métodos abstractos
 - .El Modificador final
- .Interfaces
 - .Herencia de Interfaces
 - .Práctica Uso de la Herencia

Polimorfismo

- .Formas de Polimorfismo
 - .Polimorfismo por sobrecarga
 - .Polimorfismo por sobreescritura
- .Ligado Temprano y Ligado Tardío
- .Práctica de Polimorfismo

Colecciones

- .Framework de colecciones
 - .Introducción
 - .Tipos de colecciones
 - .Autoboxing
- .Interfaces y clases coleccion
 - .Interfaz List
 - .La Clase ArrayList
 - .Interfaz Set
 - .Clase HashSet
 - .Interfaz Map
 - .La clase HashTable
- .Genéricos y Colecciones
- .El operador diamante
- .Recorriendo elementos de una colección
 - .for-each

Excepciones

- .Clasificación de errores
 - .Excepciones y Errores como objetos
- .Manejo de Excepciones
 - .El bloque try-catch
 - .Múltiples Atrapadas (catches)
 - .El bloque finally
 - .El flujo de las Excepciones
 - .Re-lanzando una Excepción
- .Excepciones Personalizadas
- .Sentencias Multicatch
- .Gestión automática de recursos
- .Métodos que lanzan Excepciones
 - .Invocación de métodos que lanzan excepciones

Streams Entrada/Salida

- .Definición de streams
 - .El paquete javaio
- .Streams nodos y filtros
 - .El Enfoque Básico
 - .El Enfoque Avanzado
- .Accesando Archivos
- .Lectura del teclado y Escritura en pantalla
- .Clases stream avanzadas
 - .Práctica: Uso de Filtros

Threads

- .Introducción
 - .MultiProcesamiento, Multitarea y Multihilo
 - .Beneficios de los Threads
 - .La interfaz Runnable
- .Sincronización y Prioritización
 - .Problemas de acceso concurrente
 - .Sincronizacion de Threads
 - .Prioritizacion de Threads
- .Calendarizacion
 - .Calendarizacion básica
 - .Métodos wait y notify

DCInternet

JDBC

- .Introducción al JDBC
- .Instalación driver JDBC
- .Estructura básica de invocación JDBC
- .Importación del paquete de las interfaces básicas JDBC
- .Registro del driver
- .Creación y apertura de la conexión
- .Creación y ejecución de una sentencia
- .Prepared Statements
- .Recuperación de datos
- .Cerrar los recursos
- .Manejo de excepciones

Expresiones Lambda y Streams

- .Introducción a expresiones lambda
- .Ejemplo con expresiones lambda