

Curso preparación para la Certificación Java Programmer

Capítulo 1

- Identificadores
 - Convenciones del código Java
- JavaBeans
 - Estándares JavaBeans
- Declarar clases
 - Declaraciones y modificadores de Clases
- Declarar Interfaces
 - Declaración de una interfaz
 - Declarando constantes de interfaz
- Declarar miembros de una clase
 - Modificadores de acceso
 - Miembros Public
 - Miembros Privados
 - Miembros Protected o Default
 - Variables locales y Modificadores de acceso
 - Modificadores de no acceso en miembros
 - Métodos finales
 - Métodos abstract
 - Métodos synchronized
 - Métodos Native
 - Métodos Strictf
 - Métodos con lista de argumentos variable (var-args)
 - Declaración de constructores
 - Declaraciones de Variable
 - Declarando primitivos y Rangos de primitivos
 - Declaración de Variables de Referencia
 - Variables de instancia
 - Variables locales (Automática/Pila/Método)
 - Declaración de Array
 - Variables Finales
 - Variables Transient

- Variables Volatile
- Variables y Métodos estáticos
- Declarando enumerados (enum)

Capítulo 2

- Encapsulación
- Herencia, "Es-Un" (Is-A) y "Tiene-Un" (Has-A)
 - Relaciones Is-A y Has-A
- Polimorfismo
- Sobrescritura/Sobrecarga
 - Métodos Sobrescritos
 - Métodos Sobrecargados
 - Polimorfismo en Métodos Sobrescritos y Sobrecargados
- Casting a Variables de Referencia
- Implementando una Interfaz
- Tipos de Retorno Legales
 - Declaraciones de Tipos de Retorno
 - Devolver un Valor
- Constructores e Instanciación
 - Conceptos Básicos de los Constructores
 - Encadenado Constructores
 - Reglas para los Constructores
 - Determinar si será Creado un Constructor por Defecto
 - ¿Cómo Podemos estar Seguros de si se Creará un Constructor por Defecto?
 - ¿Cómo Podemos Saber el Aspecto del Constructor por Defecto?
 - ¿Qué Ocurre si el Super Constructor tiene Argumentos?
 - Constructores Sobrecargados
- Estáticos
 - Métodos y Variables Estáticos
 - Accediendo a Métodos y Variables Estáticos
- Acoplamiento y Cohesión
 - Acopilamiento
 - Cohesión

Capítulo 3

- Rápido repaso

- Literales, Asignaciones y Variables
 - Valores literales para todos los tipos Primitivos
 - Asignación de operadores
 - Ámbito de Variable
 - Uso de una variable o elemento de Array que ni está inicializado ni asignado
 - Variables de instancia de tipo primitivo y Object
 - Instanciar Variables Primitivas
 - Instanciar Variables de referencia a Objetos
 - Variables de instancia Array
 - Variables locales (Stack, automatic) Primitivas y
 - Asignando una variable referencia a otra
- Pasando variables a los métodos
 - Pasando referencias a objeto
 - Pasando variables primitivas
- Array declaración, construcción e inicialización
 - Declarando un array
 - Construyendo un array
 - Inicializando un array
 - Declarando, construyendo e inicializando en una línea
 - Construyendo e inicializando un array anónimo
 - Arrays de primitivos
 - Arrays de referencias a objetos
 - Asignaciones de referencias de array para Arrays Unidimensionales
 - Asignaciones de referencias de array para Arrays Multidimensionales
 - Bloques de inicialización
- Usando clases de envoltura y Boxing
 - Descripción general de clases de envoltura
 - Creando objetos de envoltura
 - Los constructores de envoltura
 - Los métodos valueOf()
 - Usando utilidades de conversión en clases de envoltura
 - Donde puede ser usado Boxing
- Sobrecarga
 - Sobrecarga - Método concordancia
 - Sobrecarga - Boxing y Var-args
 - Ampliación (widening) de variables de referencia
 - Sobrecarga combinando Widening y Boxing
 - Sobrecarga en combinación con var-args
- Garbage Collection

- Descripción de la gestión de memoria y garbage collection
- Descripción del garbage collector de Java
- Cuando se ejecuta el garbage collector
- Como trabaja el garbage collector
- Escribir código que marque explícitamente objetos elegibles para Collection
- Limpiando antes de Garbage Collector (método finalize())

Capítulo 4

- Operadores Java
 - Operadores de Asignación
 - Operadores Relacionales
 - Comparación instanceof
 - Operadores Aritméticos
 - Operador Condicional
 - Operadores Lógicos

Capítulo 5

- Sentencias if y switch
- Bucles e iteradores
 - While
 - Do
 - For
 - Break y Continue
- Manejo de excepciones
 - Try y Catch
 - Finally
 - Programando excepciones
 - Definiendo excepciones
 - Jerarquía Exception
 - Manejo de toda una jerarquía de clases de excepciones
 - Excepciones concordantes (Exception Matching)
 - Declaración Excepción y la interfaz pública.
- Excepciones y errores comunes.
 - De donde vienen las excepciones.
 - JVM lanzando Excepciones
 - Lanzando excepciones programáticamente.

- Resumen de las excepciones y errores.
- Trabajando con Assertion
 - Assertions
 - Reglas de expresión Assertion
 - Activando assertions
 - Usando assertions apropiadamente

Capítulo 6

- String, StringBuilder y StringBuffer
 - La Clase String
 - Cósas Importantes sobre los String y la Memoria
 - Métodos Importantes en la Clase String
 - Las Clases StringBuffer y StringBuilder
 - Métodos Importantes en las Clases StringBuffer y StringBuilder
- Manejo de Ficheros y I/O
 - Creando Ficheros Mediante la Clase File
 - Usando FileWriter y FileReader
 - Combinando Clases I/O
 - Trabajando con Ficheros y Directorios
- Serialización
 - Trabajando con ObjectOutputStream y ObjectInputStream
 - Representación de Objetos
 - Usar WriteObject y ReadObject
 - ¿Cómo afecta la Herencia a la Serialización?
 - La Serialización no vale para los Estáticos
- Fechas, Números y Divisas
 - Trabajando con Fechas, Números y Divisas
- Parseo, Tokens y Formateo
 - Tutorial de Búsqueda
 - Localizando Datos a través de Patrones
 - Tokenizando
 - Formateo con printf() y format()

Capítulo 7

- Sobrescribiendo hashCode() y equals()
 - Método toString()
 - Método equal()

- Significado de no sobrescribir equals()
- Implementando un método equals()
- Sobrescribiendo hashCode
- Entendiendo los HashCodes
- Implementando hashCode()
- El contrato hashCode()
- Colecciones
 - Que hacer con una colección
 - Interfaces y clases del framework Collection
- Usando el framework Collections
 - ArrayList Básicos
 - Autoboxing con Collections
 - Clasificando (sorting) Collections y Arrays
 - Sorting Collections
 - Interfaz Comparable
 - Sorting con Comparator
 - Sorting con la clase Arrays
 - Buscando en Arrays y Collections
 - Convirtiendo Arrays a Listas (y viceversa)
 - Usando Lists
 - Usando Sets
 - Usando Maps
 - Usando la clase PriorityQueue
 - Descripción Método para Arrays y Colecciones
 - Descripción Método para List, Set, Map, y Queue
- Tipos genéricos
 - Manera legal de hacer colecciones
 - Genéricos y el legado de código
 - Mezclando colecciones genéricas y no genéricas
 - Polimorfismo y Genéricos
 - Métodos genéricos
 - Declaraciones genérico
 - Creando tu propia clase genérica
 - Creación de métodos genéricos

Capítulo 8

- Clases Internas
- Clases Internas Locales a Métodos
- Clases Internas Anónimas

- Clases Anidadas Estáticas

Capítulo 9

- Definir, Instanciar e iniciar Hilos
- Crear un Hilo
 - - Extender la clase Thread
 - - Implementando `java.lang.Runnable`
- Instanciando un Hilo
- Iniciando un Hilo
 - - Iniciando y Ejecutando Varios Hilos
 - - El Scheduler (Planificador) de Hilos
- Estados de Hilos y Transiciones
- Estados de Hilos
- Prevenir la Ejecución de un Hilo
 - - Sleeping
- Prioridades de Hilos y `yield()`
 - - Modificando la Prioridad de un Hilo
 - - El método `yield()`
 - - El Método `join()`
 - - Resumen
- Código de Sincronización
- Problema de Concurrencia
 - - Previniendo el Problema
- Sincronización y Locks
 - - ¿Pueden Sincronizarse los Métodos Estáticos?
 - - ¿Qué ocurre si un Hilo no puede obtener el Lock?
 - - ¿Cuándo Necesito Sincronizar?
 - - Clases Seguras en Hilos
- Abrazo Mortal de Hilos
- Interacción entre Hilos