

DENOMINACIÓN: **Certificación oficial de Programador Java**

OBJETIVOS: **Obtener** la Certificación Oficial de Programador Java por Sun Microsystems

DIRIGIDO A: Curso dirigido a programadores, analistas, desarrolladores de software interesados en convertirse en programadores certificados por Sun Microsystem.

REQUISITOS: Tener conocimientos de programación

HORAS: 40 Horas

METODOLOGÍA: Presencial

COSTE: 4200€

CONTENIDOS

Modulo 1: Declaraciones y control de accesos

- Escribir código que declare, construya e inicialice matrices (arrays) de cualquier tipo básico utilizando alguna de las formas permitidas de declaración e inicialización.
- Declarar clases, clases anidadas, métodos, variables de instancia, variables estáticas y variables automáticas (método local) utilizando adecuadamente todos los modificadores permitidos (como public, final, static, abstract, etc.). Especificar la importancia de cada uno de estos modificadores por separado y en combinación, e indicar el efecto que producen las relaciones de paquetes sobre los elementos declarados calificados por esos modificadores.
- Dada una clase, determinar si se creará un constructor predeterminado y, en caso afirmativo, indicar el prototipo de ese constructor.
- Especificar los tipos admitidos que puede devolver un método dadas las declaraciones de todos los métodos relacionados en su clase o en las clases de nivel superior.

Modulo 2 : Control del flujo, afirmaciones y manejo de excepciones

- Escribir código utilizando las instrucciones if y switch, e identificar los tipos de argumentos válidos para estas instrucciones.
- Escribir código utilizando todas las formas de bucles, incluidos los de instrucciones break y continue utilizadas con y sin etiqueta, e indicar los valores que adoptan las variables de control del bucle durante y después de la ejecución del bucle.
- Escribir código que utilice adecuadamente excepciones y cláusulas de manejo de excepciones (try, catch, finally), y que declare métodos y métodos sustitutivos (overrides) que generen excepciones.
- Reconocer el efecto que produce una excepción que se genera en un punto dado de un fragmento de código. Nota: Puede ser una excepción runtime, una excepción comprobada (checked) o un error (el programa debe incluir las cláusulas try, catch o finally en cualquier combinación válida).
- Escribir código que utilice adecuadamente afirmaciones y distinga entre el uso apropiado e inapropiado de las afirmaciones.
- Identificar las instrucciones correctas en el mecanismo de afirmación.

Modulo 3 : Liberación de recursos

- Explicar el comportamiento que garantiza el sistema de liberación de recursos (garbage collection).
- Escribir código que explícitamente cree objetos aptos para reciclaje.
- Identificar el punto de un fragmento de código dado en el que un objeto se puede elegir para reciclaje.

Modulo 4: Fundamentos del lenguaje

- Identificar correctamente declaraciones package, instrucciones import, declaraciones de clases (en todas sus formas, incluidas las internas), declaraciones de interfaces, declaraciones de métodos (incluido el método main que se utiliza para iniciar la ejecución de una clase), declaraciones de variables e identificadores.
- Identificar clases que implementan correctamente una interfaz java.lang.Runnable u otra interfaz completamente especificada en la pregunta.
- Establecer la correspondencia entre los valores de índice en la matriz (array) de argumentos pasados a un método main y los argumentos de la línea de comandos.
- Identificar todas las palabras reservadas de Java. Nota: No habrá preguntas que obliguen a hacer extrañas distinciones entre palabras reservadas y constantes manifiestas.
- Indicar el efecto de usar una variable o un elemento de matriz de cualquier tipo cuando no ha sido asignado explícitamente.
- Establecer el rango de valores de todos los tipos de datos primitivos y declarar valores literales para String y todos los tipos primitivos utilizando todos los formatos, bases y representaciones permitidos.

Modulo 5: Operadores y asignaciones

- Determinar el resultado de aplicar cualquier operador (incluidos los operadores de asignación e instanceof) a operandos de cualquier tipo, clase, ámbito o accesibilidad, o cualquier combinación de estos.
- Determinar el resultado de aplicar el método boolean equals(Object) a objetos de cualquier combinación de las clases java.lang.String, java.lang.Boolean y java.lang.Object.
- Dada una expresión que incluye los operadores &, |, &&, || y variables de valores conocidos, indicar qué operandos se evalúan y el valor de la expresión.
- Indicar cómo afecta a los objetos y valores primitivos el pasar variables a métodos y realizar asignaciones y otras operaciones de modificación dentro de ese método.

Modulo 6: Sobrecarga (overloading), sobrescritura (overriding), tipo Runtime y programación orientada a objetos

- Indicar las ventajas de la encapsulación en un diseño orientado a objetos (OO) y escribir código que implemente clases bien encapsuladas y las relaciones "is a" y "has a".
- Escribir código que llame a métodos sobrescritos o sobrecargados y constructores sobrecargados o de jerarquía superior, y describir el efecto de llamar a esos métodos.
- Escribir código para construir instancias de cualquier clase, incluidas clases normales de nivel superior y clases anidadas.

Modulo 7: Threads

- Escribir código para definir, instanciar e iniciar nuevos threads utilizando java.lang.Thread y java.lang.Runnable.
- Reconocer las condiciones que pueden impedir la ejecución de un thread.
- Escribir código utilizando synchronized, wait, notify y notifyAll para evitar problemas de accesos concurrentes y permitir la comunicación entre threads.
- Definir la interacción entre threads y bloqueos de objetos al ejecutar los métodos synchronized, wait, notify o notifyAll.

Modulo 8: Clases fundamentales del paquete java.lang

- Escribir código utilizando los siguientes métodos de la clase java.lang.Math: abs, ceil, floor, max, min, random, round, sin, cos, tan y sqrt.
- Explicar el significado de la inmutabilidad de los objetos String.
- Explicar la importancia de las clases Wrapper, lo que incluye seleccionar aquellas que mejor cumplen los requisitos de comportamiento dados, indicar el resultado de ejecutar un fragmento de código que incluya una instancia de una de las clases Wrapper y escribir código que utilice los siguientes métodos de las clases Wrapper (es decir, Integer, Double, etc.):
 - o doubleValue
 - o floatValue
 - o intValue
 - o longValue
 - o parseXxx
 - o getXxx
 - o toString
 - o toHexString

Modulo 9: Arquitectura de colecciones

- Hacer la selección apropiada de clases/interfaces que respondan a unos requisitos de comportamiento dados.
- Distinguir entre las implementaciones correctas e incorrectas de los métodos hashCode.

DATOS DEL PROVEEDOR

Denominación Social: Glokal Consulting S.L.

C.I.F.: B- 83853705

DOMICILIO SOCIAL: Paseo de la Castellana, 139, 8º Dcha

PERSONA DE CONTACTO: Paula García Marina

TELÉFONOS: 672 034 689 / +34 911 852 476

FAX:+34 911 852 479

E-MAIL: paula.garcia@glokal.com