

NOMBRES Y APELLIDOS: _____

IDENTIFICACIÓN: _____ CODIGO ESEIT: _____

REGISTRO MATERIA CODIGO: _____ FECHA: _____

Selecciona la opción correcta para cada pregunta. Cada pregunta tiene una única respuesta correcta.

1. ¿Qué característica de Java permite que el mismo código se ejecute en diferentes plataformas sin modificaciones?

- A) Compilación estática
 - B) Máquina Virtual de Java (JVM)
 - C) Herencia múltiple
 - D) Tipado dinámico
-

2. ¿Cuál de los siguientes patrones de diseño se utiliza comúnmente para garantizar que una clase tenga una única instancia y proporcionar un punto de acceso global a ella?

- A) Factory Method
 - B) Observer
 - C) Singleton
 - D) Strategy
-

3. En Java, ¿qué interfaz debe implementar una clase para que sus objetos puedan ser serializados?

- A) Cloneable
 - B) Serializable
 - C) Comparable
 - D) Runnable
-

4. ¿Qué mecanismo de Java se utiliza para gestionar la concurrencia y evitar condiciones de carrera en entornos multihilo?

- A) Sincronización con la palabra clave synchronized
 - B) Uso de colecciones inmutables
 - C) Herencia de la clase Thread
 - D) Todas las anteriores
-

5. ¿Qué es el "garbage collector" en Java?

- A) Un compilador de código fuente
 - B) Un depurador de errores en tiempo de ejecución
 - C) Un mecanismo automático que libera memoria de objetos no referenciados
 - D) Una herramienta para optimizar el rendimiento del CPU
-

6. ¿Cuál de las siguientes afirmaciones sobre los Streams en Java 8+ es correcta?

- A) Los Streams modifican la colección original
 - B) Los Streams siempre se ejecutan en paralelo
 - C) Los Streams permiten procesamiento funcional de colecciones sin modificarlas
 - D) Los Streams solo pueden usarse con arreglos
-

7. ¿Qué es la inyección de dependencias (Dependency Injection) en el contexto de la ingeniería del software con Java?

- A) Un método para inyectar código malicioso en una aplicación
 - B) Un patrón que permite que las dependencias de una clase sean proporcionadas externamente
 - C) Una técnica para compilar código Java más rápido
 - D) Un tipo de herencia en Java
-

8. ¿Cuál de las siguientes NO es una característica del paradigma orientado a objetos en Java?

- A) Encapsulamiento
 - B) Polimorfismo
 - C) Compilación anticipada (AOT)
 - D) Herencia
-

9. En Java, ¿qué ocurre si una clase implementa una interfaz pero no implementa todos sus métodos?

- A) El programa se ejecuta normalmente
 - B) Se genera un error en tiempo de ejecución
 - C) La clase debe declararse como abstracta
 - D) Java ignora los métodos no implementados
-

10. ¿Qué ventaja ofrece el uso de módulos en Java (introducidos en Java 9 con Project Jigsaw)?

- A) Mayor velocidad de compilación
 - B) Mejor gestión de dependencias y encapsulamiento a nivel de paquete
 - C) Eliminación total del uso de JARs
 - D) Soporte nativo para inteligencia artificial
-

11. ¿Qué es la inyección de dependencias (Dependency Injection) en Java y cómo se implementa comúnmente en frameworks como Spring?

12. Explica brevemente qué son los Streams en Java 8 y proporciona un ejemplo que filtre una lista de enteros mayores que 10 y los eleve al cuadrado.

13. ¿Cuál es la diferencia entre `synchronized` y `ReentrantLock` en Java para el control de concurrencia?

(1 punto)

Respuesta:

- `synchronized` es una palabra clave que proporciona un mecanismo de bloqueo implícito, más simple pero menos flexible.
 - `ReentrantLock` es una clase (`java.util.concurrent.locks.ReentrantLock`) que ofrece más control: permite intentar adquirir el bloqueo con `timeout`, interrumpir hilos, y usar múltiples condiciones. Además, debe liberarse explícitamente con `unlock()`.
-

14. ¿Qué es un patrón de diseño Singleton y cómo se implementa de forma segura en un entorno multihilo en Java?

15. ¿Qué es la serialización en Java y qué interfaz debe implementar una clase para ser serializable?

16. ¿Qué es un "memory leak" en Java y cómo puede ocurrir a pesar del recolector de basura (Garbage Collector)?

17. ¿Qué ventajas ofrece el uso de `Optional<T>` en Java 8 respecto al manejo tradicional de valores nulos?

18. Explica brevemente qué es el principio SOLID "Inversión de Dependencias" (Dependency Inversion Principle) y cómo se aplica en Java.

19. ¿Qué es un "thread pool" y cómo se crea uno en Java usando la clase `ExecutorService`?

20. ¿Qué es la anotación `@FunctionalInterface` en Java y cuál es su propósito?