**Introducción a Java**

1) ¿Qué es el código de ensamblaje y el programa traductor ensamblador?

2) ¿Qué es la abstracción en informática?

3) Explique los diferentes niveles de abstracción en lenguajes de programación y de algunos

ejemplos de lenguajes por nivel de abstracción

4) Defina interprete, compilador y explique la diferencia entre estos dos conceptos

5) ¿Qué es un paradigma de programación? Nombre algunos de ellos

6) Explique que es el paradigma orientado a objetos

7) ¿Cómo es que Java es tan popular en tecnologías webs, si cuando se creó el lenguaje no

existía Internet tal y como se la conoce en la actualidad?

8) Liste las características más relevantes del lenguaje y desarrolle brevemente cada una de

ellas

9) ¿Cómo es que Java es multiplataforma y al mismo tiempo sus programas se ejecutan de

una manera eficiente respecto al tiempo de ejecución?

10) ¿Qué es y cómo funciona la máquina virtual de Java?

**RESPUESTAS:**

1) El código de ensamblaje es un conjunto de instrucciones de bajo nivel que representan operaciones directas para la arquitectura de un procesador. Por ejemplo, en ensamblaje x86, MOV se utiliza para mover datos entre registros o memoria. El programa traductor ensamblador es un software que convierte este código de ensamblaje a lenguaje máquina, entendible por la CPU.

2) La abstracción en informática implica simplificar un sistema al ocultar detalles innecesarios y mostrar solo las funciones esenciales. Por ejemplo, al usar una biblioteca en un lenguaje de programación, como la función printf() en C, los programadores no necesitan conocer los detalles internos de cómo funciona esa función, solo necesitan saber cómo utilizarla para imprimir texto en la consola.

3) Los niveles de abstracción en lenguajes de programación van desde el bajo nivel, como el lenguaje ensamblador que se acerca más al hardware, hasta el alto nivel, como Python, que se enfoca en la lógica del problema sin preocuparse por los detalles de implementación. Por ejemplo, en C, para acceder a la memoria se utilizan punteros, mientras que en Python, esto se maneja automáticamente y los programadores no tienen que preocuparse por la gestión de memoria.

4) Un intérprete ejecuta código línea por línea, traduciéndolo a lenguaje máquina sobre la marcha. Por ejemplo, en Python, el intérprete ejecuta cada instrucción del código fuente de Python a medida que se encuentra. Por otro lado, un compilador traduce todo el código a lenguaje máquina antes de la ejecución, generando un archivo ejecutable. Un ejemplo es el compilador de C, que traduce el código C a lenguaje máquina y produce un archivo ejecutable que puede ejecutarse sin necesidad de un intérprete.

5) Un paradigma de programación es un enfoque o estilo para resolver problemas de programación. Por ejemplo, el paradigma imperativo se centra en cómo se deben ejecutar las instrucciones para cambiar el estado del programa, mientras que el paradigma funcional se enfoca en aplicar funciones matemáticas puras para resolver problemas, evitando el cambio de estado.

6) El paradigma orientado a objetos organiza el software en objetos que tienen datos y métodos que operan en esos datos. Por ejemplo, en Java, una clase Persona podría tener datos como nombre y edad, y métodos como obtenerNombre() y obtenerEdad(), que operan en esos datos.

7) Java es popular en tecnologías web debido a su portabilidad, seguridad y facilidad de uso. Aunque Java fue creado antes de la explosión de Internet tal como la conocemos hoy, su diseño robusto y su capacidad para ejecutarse en diferentes plataformas lo han hecho ideal para el desarrollo web.

8) Java tiene características como portabilidad (puede ejecutarse en diferentes plataformas), orientación a objetos (organiza el código en objetos), robustez (manejo de excepciones), seguridad (entorno seguro para ejecutar código) y facilidad de uso (sintaxis clara y bibliotecas extensas).

9) Java es multiplataforma porque su código fuente se compila en bytecode, que luego se ejecuta en la Máquina Virtual de Java (JVM), disponible en diferentes plataformas. La eficiencia en el tiempo de ejecución se logra gracias a la optimización realizada por la JVM durante la ejecución del bytecode.

10) La Máquina Virtual de Java (JVM) es un componente esencial que ejecuta el bytecode de Java. La JVM se encarga de tareas como la gestión de memoria, la ejecución del código y la recolección de basura, lo que permite que el mismo bytecode se ejecute en diferentes plataformas sin necesidad de modificar el código fuente.