

1. Introducción a Java

Test de autoevaluación

- 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 era una notación simbólica que representaba las operaciones que podía realizar la computadora. Al principio la traducción del código de ensamblaje a código de máquina se hacía manualmente, pero al ver que de esta tarea podría encargarse también la computadora se creó un programa traductor llamado ensamblador.
- 2) La abstracción en informática implica ignorar selectivamente ciertos detalles o partes de un proceso para facilitar su comprensión y manejo. En el contexto del lenguaje ensamblador, esta abstracción se ejemplifica al utilizar una instrucción simple como "add destino, fuente" para realizar una operación de suma.
- 3) Los diferentes niveles de abstracción en lenguajes de programación son:
 - Bajo nivel: Lenguajes como ensamblador que están más cerca del lenguaje de máquina.
 - Medio nivel: Lenguajes como C que permiten un equilibrio entre control de hardware y abstracción.
 - Alto nivel: Lenguajes como Java, Python, y PHP que ofrecen una mayor abstracción y facilidad de programación.
- 4) Un intérprete es un programa que traduce y ejecuta el código fuente línea por línea, mientras que un compilador traduce todo el código fuente a código de máquina antes de la ejecución. La diferencia principal radica en el momento de la traducción y ejecución del código.
- 5) Un paradigma de programación es un enfoque o estilo para resolver problemas de programación. Algunos ejemplos de paradigmas son: imperativo, lógico, declarativo, orientado a eventos y orientado a objetos.

6) El paradigma orientado a objetos es un enfoque de programación que se basa en la creación de objetos que contienen datos y métodos para interactuar con esos datos. Permite la reutilización de código, la encapsulación y la herencia.

7) Java se ha vuelto popular en tecnologías web debido a su portabilidad, seguridad, y facilidad de uso. Aunque en sus inicios no estaba destinado específicamente para internet, su capacidad multiplataforma y su soporte para redes lo han convertido en una opción atractiva para el desarrollo web.

8) Algunas características relevantes de Java son:

- Facilidad de uso: Java combina poder y facilidad de uso, siendo familiar para programadores de C y C++.

- Independencia de plataforma: Los programas Java pueden ejecutarse en múltiples plataformas sin necesidad de recompilar.

- Soporte para trabajo en red: Java ofrece transparencia al trabajar en red mediante Sockets y URL.

- Seguridad en la ejecución de aplicaciones remotas: Java presenta un sistema de seguridad por capas basado en certificados.

9) Java es multiplataforma y eficiente en tiempo de ejecución gracias a su máquina virtual (JVM). La JVM interpreta el bytecode Java y lo convierte en código nativo de la plataforma en tiempo de ejecución, permitiendo la portabilidad y la eficiencia en la ejecución de programas.

10) La máquina virtual de Java (JVM) es un componente esencial que interpreta y ejecuta el bytecode Java en diferentes plataformas. La JVM convierte el bytecode en código nativo de la plataforma en tiempo de ejecución, lo que permite la portabilidad de los programas Java y una ejecución eficiente en diferentes sistemas operativos.

2. Elementos básicos del lenguaje

Test de autoevaluación

1) ¿Qué es un linker o cargador, y para qué sirve?

2) ¿Cuáles son las tres zonas de memoria utilizadas por Java?

3) Explique para qué sirve la zona de memoria de datos

4) Explique para qué sirve la zona de memoria conocida como pila o stack

5) Explique para qué sirve la zona de memoria conocida como montículo o heap

6) ¿Qué es un atributo y de qué tipos de datos puede ser definido?

7) ¿Cuál es la diferencia entre tipos de datos primitivos y de referencia? Explique cada una de estas categorías

8) El tipo de dato String, ¿Es un tipo de dato primitivo o de referencia? Explique las particularidades correspondientes al mismo

- 9) Qué es la conversión entre tipos de datos y en qué casos resultaría útil hacer este tipo de conversión. Detalle los problemas que pueden surgir al trabajar con conversiones
- 10) Liste y explique las convenciones de escritura adoptado por el común denominador de programadores.

Respuestas:

- 1) Un linker o cargador es una interfaz que se encarga de suministrar a un programa un bloque de memoria sobre el cual se va a ejecutar. Su función es esencial para que un programa pueda ejecutarse en un sistema operativo.
- 2) Las tres zonas de memoria utilizadas por Java son la zona de datos, la pila (stack) y el montículo (heap).
- 3) La zona de memoria de datos en Java se utiliza para almacenar las instrucciones del programa ejecutable en código de máquina, las constantes, el código correspondiente a los métodos y a las clases.
- 4) La zona de memoria conocida como pila o stack se utiliza para guardar variables locales, variables de referencia, parámetros y valores de retorno, resultados parciales, y el control de la invocación y retorno de métodos.
- 5) La zona de memoria conocida como montículo o heap es un espacio de memoria dinámico en donde se guardan variables de instancia y objetos. El Garbage Collector se encarga de administrar este espacio de memoria, liberando al programador de estas tareas.
- 6) Un atributo en programación es un espacio de memoria dentro de una clase donde se pueden almacenar datos. Estos atributos representan las características o propiedades de un objeto y permiten guardar información relevante para su funcionamiento.

Los atributos pueden ser definidos con diferentes tipos de datos, como por ejemplo:

- 1- Tipos de datos primitivos: Incluyen enteros (int), decimales (double, float), caracteres (char), booleanos (boolean), entre otros.
- 2- Tipos de datos de referencia: Incluyen objetos, interfaces y arreglos. Estos tipos de datos hacen referencia a posiciones de memoria que contienen los valores reales de los objetos, en lugar de contener directamente el valor como en los tipos primitivos.
- 7) Los tipos de datos primitivos son valores simples como enteros, booleanos, etc., mientras que los tipos de datos de referencia son referencias a objetos, interfaces y arreglos. La diferencia radica en que los tipos primitivos contienen directamente el valor, mientras que los tipos de referencia contienen una referencia a un objeto en memoria.
- 8) El tipo de dato String en Java es un tipo de dato de referencia. Aunque se utiliza de manera similar a un tipo primitivo, en realidad es un objeto que contiene métodos y propiedades específicas para manipular cadenas de caracteres.

9) La conversión entre tipos de datos consiste en cambiar un tipo de dato a otro compatible. Es útil para adaptar datos a diferentes operaciones o formatos. Los problemas que pueden surgir al trabajar con conversiones incluyen la pérdida de precisión en algunos casos o posibles errores si la conversión no es válida.

10) Las convenciones de escritura adoptadas por el común denominador de programadores incluyen:

1- Clases: Deben ser escritas en minúsculas, con la primera letra de cada palabra en mayúscula. Por ejemplo: `MiClase`, `ClasePrincipal`, `Fisica`, `ManejadorDeGraficos`.

2- Métodos: Deben ser verbos escritos en minúsculas, con la primera letra de cada palabra en mayúscula, excepto la primera palabra del método. Por ejemplo: `dibujarRectangulo()`, `animar()`, `sumatoriaDeSaldos()`, `importarBaseDeDatos()`.

3- Atributos: Deben ser escritos en minúsculas, con la primera letra de cada palabra en mayúscula, excepto la primera palabra del identificador del atributo. Por ejemplo: `double` `perimetro`, `boolean` `datoBandera`, `int` `saldosEnteros`.

Estas convenciones son importantes para mantener un código limpio, legible y consistente, facilitando la comprensión y colaboración entre programadores.