**Elementos básicos del lenguaje**

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 herramienta de software que se encarga de vincular diferentes módulos de un programa durante la fase de enlazado. Por ejemplo, cuando compilas un programa en Java, el linker se asegura de combinar todas las clases y bibliotecas necesarias para crear un archivo ejecutable o un archivo JAR.

2) Las tres zonas de memoria en Java son el heap, la stack y el área de métodos. Por ejemplo, creamos un objeto llamado "persona" en Java. La información relacionada con este objeto, como su nombre y edad, se almacenaría en el heap. Mientras que las variables locales, como un contador utilizado en un método, se almacenarían en la stack. Por último, el área de métodos contiene el bytecode del método en sí.

3) La zona de memoria de datos almacena variables estáticas y constantes durante toda la ejecución del programa. Por ejemplo, si tenemos una clase en Java que contiene una variable estática llamada "numeroDePersonas", esta variable se almacenaría en la zona de memoria de datos y podría ser accesible desde cualquier parte del programa.

4) La zona de memoria conocida como pila o stack se utiliza para almacenar variables locales y referencias a objetos. Por ejemplo, si tenemos un método en Java que declara una variable entera llamada "edad", esta variable se almacenaría en la stack. Además, las referencias a objetos, como un objeto "persona", también se almacenarían en la stack.

5) La zona de memoria conocida como montículo o heap se utiliza para almacenar objetos dinámicos creados durante la ejecución del programa. Por ejemplo, si creamos un nuevo objeto "persona" en Java utilizando la palabra clave "new", este objeto se asignaría en el heap y su referencia se almacenaría en la stack.

6) Un atributo es una característica de un objeto que define su estado. Por ejemplo, si tenemos una clase en Java llamada "Persona" con atributos como "nombre" y "edad", podemos definir estos atributos de la siguiente manera:

public class Persona {

String nombre;

int edad;

}

7) Los tipos de datos primitivos son datos simples como enteros, flotantes y caracteres, que se almacenan directamente en la memoria. Por ejemplo, los tipos de datos primitivos en Java incluyen int, float, char, etc. Por otro lado, los tipos de datos de referencia son objetos complejos que se almacenan en el heap y se accede a ellos a través de referencias. Por ejemplo, si creamos un objeto "persona" en Java, estamos utilizando un tipo de dato de referencia.

8) El tipo de dato String en Java es un tipo de dato de referencia. Aunque se utiliza comúnmente como un tipo de dato primitivo debido a su simplicidad, en realidad es un objeto que se almacena en el heap y su referencia se almacena en la stack. Por ejemplo, si declaramos una variable String en Java, como "String nombre = "moya"", estamos creando un objeto String en el heap y almacenando su referencia en la stack.

9) La conversión entre tipos de datos es el proceso de cambiar un tipo de dato a otro. Por ejemplo, si tenemos un número entero y queremos convertirlo a un número flotante en Java, podemos usar una conversión explícita como sigue:

int entero = 5;

float flotante = (float) entero;

10) Las convenciones de escritura en Java incluyen usar nombres descriptivos para variables y métodos, escribir en camelCase, indentar el código de manera consistente y usar comentarios para explicar el propósito del código. Por ejemplo:

int numeroDeEstudiantes = 10;

public void imprimirMensaje() {

System.out.println("Hola mundo!");

}