

UNIVERSIDAD TECNOLÓGICA DE PANAMÁ
FACULTAD DE INGENIERÍA EN SISTEMAS COMPUTACIONALES
ASIGNATURA: Programación

INVESTIGACIÓN 1

Tema: Construcción de Clase en Java

Estudiante: Justing He

Profesor: Napoleón Ibarra

Fecha: 20/08/2025

Introducción

El lenguaje de programación Java se caracteriza por ser orientado a objetos, lo que significa que gran parte de su estructura se basa en clases. La construcción de clases es un tema fundamental, ya que permite organizar el código, favorecer la reutilización y mejorar la comprensión de los programas. En este trabajo se abordarán los miembros de una clase, los modificadores de acceso y otras opciones relacionadas con su construcción, destacando su concepto, importancia, ventajas y desventajas, además de ejemplos prácticos.

Desarrollo

1.3.1 Miembros de una clase

- Concepto: Son los elementos que componen una clase en Java, tales como atributos (variables), métodos (funciones), constructores y bloques de inicialización.
- Importancia: Permiten definir el comportamiento y las características de los objetos, siendo la base de la programación orientada a objetos.
- Ventajas y Desventajas:
 - Ventajas: Organización clara del código, reutilización de métodos, encapsulación.
 - Desventajas: Puede aumentar la complejidad si no se estructura bien.
- Ejemplo:

```
class Persona {  
    String nombre;  
    int edad;  
  
    void saludar() {  
        System.out.println("Hola, mi nombre es " + nombre);  
    }  
}
```

1.3.2 Modificadores de acceso

- Concepto: Son palabras clave que controlan el nivel de visibilidad de clases, métodos y atributos en Java. Los más comunes son: public, private, protected y default.
- Importancia: Garantizan la seguridad y control del acceso a los datos dentro de una clase, promoviendo la encapsulación.
- Ventajas y Desventajas:
 - Ventajas: Protección de datos, control de acceso, claridad en el diseño.
 - Desventajas: Un uso incorrecto puede limitar la reutilización o generar errores de acceso.

- Ejemplo:

```
class CuentaBancaria {
    private double saldo;

    public void depositar(double cantidad) {
        saldo += cantidad;
    }
    public double getSaldo() {
        return saldo;
    }
}
```

1.3.3 Otras opciones

- Concepto: Se refiere a elementos adicionales en la definición de clases como clases abstractas, interfaces, clases internas o el uso de final y static.
- Importancia: Estas opciones permiten mayor flexibilidad y diseño avanzado en la construcción de aplicaciones.
- Ventajas y Desventajas:

- Ventajas: Fomentan la reutilización de código, permiten la implementación de herencia múltiple mediante interfaces, mayor organización.
- Desventajas: Pueden aumentar la dificultad de comprensión para principiantes, complejidad en grandes sistemas.

- Ejemplo:

```
interface Animal {
    void hacerSonido();
}

class Perro implements Animal {
    public void hacerSonido() {
        System.out.println("Guau Guau");
    }
}
```

Conclusión

La construcción de clases en Java constituye la base de la programación orientada a objetos. Los miembros de una clase definen sus atributos y comportamientos, mientras que los modificadores de acceso aseguran el control y la encapsulación. Además, las opciones adicionales permiten ampliar las capacidades del diseño. Comprender estos conceptos es esencial para el desarrollo de programas robustos, seguros y escalables.

Referencias

Wikipedia. (2025). Java (lenguaje de programación).

[https://es.wikipedia.org/wiki/Java_\(lenguaje_de_programaci%C3%B3n\)](https://es.wikipedia.org/wiki/Java_(lenguaje_de_programaci%C3%B3n))

Wikipedia. (2025). Programación orientada a objetos.

https://es.wikipedia.org/wiki/Programaci%C3%B3n_orientada_a_objetos

Wikipedia. (2025). Modificadores de acceso.

https://es.wikipedia.org/wiki/Modificador_de_acceso