

# MÉTODOS

UNIDAD 2: INTRODUCCIÓN A LA PROGRAMACIÓN ORIENTADA  
A OBJETOS

SEMANA 4



## Temario

---

- Declaración y llamado a métodos.
- Métodos y atributo de Clase.
- Control de acceso a miembros.
- Alcance de la declaración en una Clase.
- Métodos sobrecargados.



<https://youtu.be/N3uOi3qin8w>

# ChatGPT and software development

How can developers use generative AI to write better code, increase productivity, and meet high user expectations?

# Declaración y llamado a métodos

- Los métodos son funciones o comportamientos definidos dentro de una clase.
- Para **declarar un método**, se especifica su nombre, parámetros y tipo de retorno dentro de la clase.
- Para **utilizar un método**, primero se crea un objeto de esa clase y luego se llama al método en ese objeto utilizando la notación de punto.
- Esto permite encapsular datos y acciones relacionadas en objetos, promoviendo la modularidad y la reutilización de código en la POO.

# Métodos y atributo de Clase

- En programación orientada a objetos, se pueden crear métodos y atributos estáticos que no dependen de instancias específicas de objetos.
- Estos **métodos estáticos** se aplican a la clase en su conjunto y son útiles para realizar tareas comunes que no requieren objetos individuales.
  - Por ejemplo, una clase podría tener un método estático para realizar un cálculo matemático común, como la raíz cuadrada, sin necesidad de crear una instancia de la clase.
- Los **atributos estáticos**, conocidos como variables de clase, son compartidos por todas las instancias de la clase y pueden utilizarse para almacenar información que debe ser compartida entre objetos.

# Diferencia entre Métodos de instancia y métodos de clase

## Métodos de instancia

- Es un método que está asociado con un objeto específico de una clase. Este método puede acceder a los atributos específicos del objeto y modificarlos si es necesario. Para llamar a un método de instancia, se necesita una instancia u objeto de la clase y se utiliza la notación de punto para acceder al método.

## Métodos de clase

- Es un método que está asociado con la clase en sí, no con un objeto específico. Este método no puede acceder a los atributos específicos del objeto, sino que se utiliza para realizar operaciones que son comunes a todas las instancias u objetos de la clase. Para llamar a un método de clase, no se necesita una instancia de la clase y se utiliza la notación de punto para acceder al método.

# Control de acceso a miembros

- El control de acceso a miembros en programación orientada a objetos (POO) se refiere a cómo se restringe o permite el acceso a los atributos y métodos de una clase.
- Los modificadores más comunes son **public** (acceso desde cualquier lugar), **private** (acceso solo dentro de la misma clase), **protected** (acceso dentro de la misma clase y sus subclases) y el acceso predeterminado (acceso dentro del mismo paquete).
- Estos controles de acceso ayudan a establecer niveles de seguridad y **encapsulación** en el código, lo que facilita la modularidad y la protección de los datos.
- Paquetes:

# Alcance de la declaración en una Clase

- En la programación orientada a objetos (POO) se refiere a las reglas que determinan quién puede acceder y utilizar los atributos y métodos declarados dentro de esa clase.
- Estas reglas están controladas por modificadores de acceso como **private**, **public**, **protected** y el acceso predeterminado (package-private).



# Paquetes

- Un "paquete" es una herramienta de organización que agrupa clases y elementos relacionados en un proyecto de software.
- Los paquetes ayudan a organizar el código, controlar el acceso a clases y evitar conflictos de nombres, facilitando la reutilización y la gestión de componentes en programas complejos.

# Sobrecarga de métodos

- La sobrecarga (overload) se evidencia en una clase cuando se declaran varios métodos con el mismo nombre pero diferenciados por número o tipo de parámetros.



# Sobrecarga de métodos

- A veces, desea tener varios métodos con el mismo nombre para poder realizar diferentes operaciones en diferentes parámetros.
- Java permite esto a través de un proceso llamado sobrecarga.
  - La sobrecarga es tener múltiples métodos en la misma clase con el mismo nombre, pero aceptan diferentes tipos de parámetros.
  - Por ejemplo:

```
public double add(double num1, double num2) {  
    return num1 + num2;  
}  
  
public String add(String str1, String str2) {  
    return str1 + str2;  
}
```

- Aunque ambos métodos se denominan agregar, realizan diferentes operaciones en diferentes parámetros.



# Sobrecarga de métodos

- Cuando llamamos a un método, el compilador debe determinar cuál de los métodos usar a través de un proceso llamado vinculación.
  - Java vincula los métodos haciendo coincidir la firma de un método con la forma en que se llama.
    - La firma de un método consta de su nombre y los tipos de datos de sus parámetros.
    - Las firmas de los dos métodos anteriores son:
      - add (doble, doble)
      - add (String, String)
  - Entonces, el compilador de Java puede decir qué método usar en función de cómo se llamó.
  - Tenga en cuenta que no puede tener métodos con el mismo nombre y los mismos tipos de datos para los parámetros AUNQUE TENGAN UN TIPO DE DEVOLUCIÓN DIFERENTE.

# ¿Por qué la sobrecarga de métodos?

- Supongamos que tiene que realizar la suma de números dados, pero puede haber cualquier cantidad de argumentos (digamos 2 o 3 argumentos para simplificar).
- Para realizar la tarea, puede crear dos métodos `sum2num(int, int)` y `sum3num(int, int, int)` para dos y tres parámetros respectivamente. Sin embargo, otros programadores, así como usted en el futuro, pueden confundirse ya que el comportamiento de ambos métodos es el mismo pero difieren en el nombre.
- La mejor forma de realizar esta tarea es sobrecargando los métodos. Y, dependiendo del argumento pasado, se llama a uno de los métodos sobrecargados. Esto ayuda a aumentar la legibilidad del programa.

# Sobrecarga del método Java

- En Java, dos o más métodos pueden tener el mismo nombre si difieren en los parámetros (diferente número de parámetros, diferentes tipos de parámetros o ambos). Por ejemplo:
  - `void func() { ... }`
  - `void func(int a) { ... }`
  - `float func(double a) { ... }`
  - `float func(int a, float b) { ... }`
- La sobrecarga de métodos también se conoce como **polimorfismo en tiempo de compilación**, **polimorfismo estático** o **enlace anticipado** en Java. En la sobrecarga de métodos en comparación con el argumento principal, el argumento secundario tendrá la prioridad más alta.

# Referencia

- Deitel, H. M. (2016). Java: como programar.
- Arturo Rozas Huacho, Algoritmos y Estructuras de Datos, 2002, Grupo Liebre, Cusco, Perú.