



UNIVERSIDAD TECNOLÓGICA DE PANAMÁ  
FACULTAD DE INGENIERÍA DE SISTEMAS COMPUTACIONALES  
LICENCIATURA EN CIBERSEGURIDAD



PROGRAMACION 1

INVESTIGACION 1

PREPARADO POR  
ADRIAN JIMENEZ 4-839-2413

A CONSIDERACIÓN DE:  
NAPOLEON IBARRA

GRUPO: 2S3111

FECHA: 20-8-25

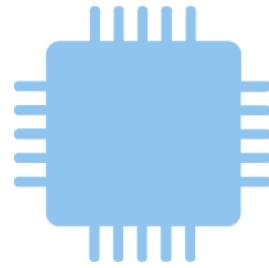
# CONSTRUCCION DE CLASE EN JAVA

Una clase en Java se puede entender como un prototipo que define las variables y los métodos comunes a un cierto tipo de instancias, una clase define todo lo que caracteriza y pueden hacer una o varias instancias.





# Miembros de una clase



## Concepto

Los miembros de una clase son los componentes heredados del cuerpo de la clase, incluyendo campos, métodos, clases anidadas, interfaces y tipos enumerados.

## Importancia

Los miembros de una clase en Java son la base para construir objetos con datos y comportamientos definidos, permitiendo la creación de sistemas robustos, modulares y mantenibles.

## Ventajas



Reutilización de código



Modularidad



Encapsulación

## Desventajas



Sobrecarga de memoria



Possible complejidad en el manejo de acceso



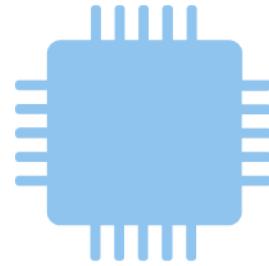
Segmentación excesiva de la lógica

## Ejemplo

```
public class Bicycle{  
    private int cadence;  
    private int gear;  
    private int speed;  
  
    // añade una variable de instancia para el ID del objeto  
    private int id;  
  
    // añade una variable de clase para la cantidad de objetos Bicycle que se hayan instanciado  
    private static int numberofBicycles = 0;  
    ....  
}
```



# Modificadores de acceso



## Concepto

Los modificadores de acceso nos introducen al concepto de encapsulamiento. El encapsulamiento busca de alguna forma controlar el acceso a los datos que conforman un objeto o instancia, de este modo podríamos decir que una clase y por ende sus objetos que hacen uso de modificadores de acceso (especialmente privados) son objetos encapsulados.

## Importancia

Son cruciales para implementar el encapsulamiento, permitiendo controlar la visibilidad y el acceso a los miembros de una clase (atributos, métodos, constructores) desde otras partes del código. Esto ayuda a proteger los datos de la clase, prevenir accesos no deseados y facilitar el mantenimiento del código..

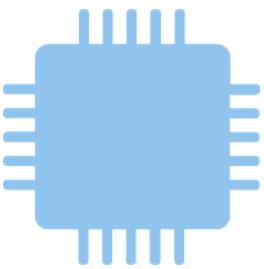
### Ventajas

- Encapsulamiento
- Seguridad
- Flexibilidad

### Desventajas

- Sintaxis más verbosa
- Mayor complejidad en proyectos grandes
- Posibles errores de acceso

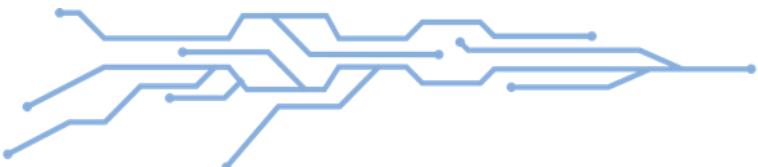
## EJEMPLO



### Comando **Public**

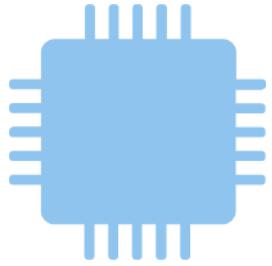
```
package com.javamagician.modificadoresacceso;

public class SuperPublic {
    public static void metodoPublic() {
        // ...
    }
}
```





# Miembros estáticos



## Concepto

Un miembro estático, ya sea una variable estática o un método estático, se asocia directamente con la clase y se inicializa al cargar la clase en la memoria.

## Importancia

La importancia de los miembros estáticos radica en su capacidad para modelar propiedades y comportamientos que son comunes a todos los objetos de una clase, como constantes, contadores o métodos de utilidad que no requieren el estado de un objeto.

## Ventajas



Eficiencia de memoria



Acceso directo



Compartición de datos

## Desventajas



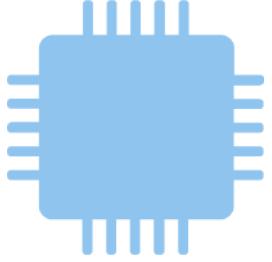
Rompe la programación orientada a objetos



Dificultad en las pruebas unitarias



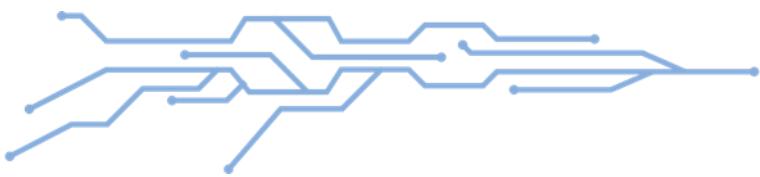
Riesgo de cambios no deseados



## EJEMPLO

La clase SumaSerie define tres variables static, y un método static que calcula la suma cada vez que se llama.

```
public class SumaSerie {  
    private static long n;  
    private static long m;  
  
    static{  
        n = 0;  
        m = 1;  
    }  
    public static long suma(){  
        m = m+n;  
        n = m-n;  
    }  
}
```





Gracias!