



P00

Instanciación - constructores - referencias

INSTANCIACIÓN

- Se realiza enviando un **mensaje** de **creación** a la **clase**.
 - ⚡ Reserva de espacio para el objeto.
 - ⚡ Ejecuta el código inicializador/**constructor**.
- Devuelve la referencia al objeto.
- **Asociar** la referencia a una **variable**(a través de ella podemos enviarle mensajes al objeto).

CONSTRUCTOR

- **Inicializa** el objeto con valores recibidos como parámetro.

⚡ Usuario jeziel; ⚡ jeziel = new Usuario("Jezieel7", 123456, "contrasena123");

⚡ Usuario jeziel = **new Usuario("Jezieel7", 123456, "contrasena123")**.

- Si la **clase no** declara ningún constructor, Java **incluye uno sin parámetros y sin código** (constructor nulo).

⚡ Si **declaramos un constructor**, Java ya **no** nos proporcionará el constructor nulo y en caso de quererlo debemos **declararlo**.

DECLARACIÓN DE CONSTRUCTORES

- Se pueden declarar **múltiples** constructores y Java entenderá que constructor queremos usar **dependiendo de los parámetros que reciba**.
- ⚡ Si **declaramos un constructor**, Java ya **no** nos proporcionará el constructor nulo y en caso de quererlo debemos **declararlo**.

EJEMPLO

CONSTRUCTOR DE

USUARIO

```
■ Usuario(String nombre, int dni, int contrena){  
    nombre = nombre;  
    dni = dni;  
    contrasena = contrasena;  
}
```

EJEMPLO CONSTRUCTOR DE USUARIO

```
■ Usuario(String nombre, int dni, int contrena){  
    this.nombre = nombre;  
    this.dni = dni;  
    this.constrasena = contrasena;  
}
```

REFERENCIA THIS

- Poniendo `this.nombreMétodo` (parámetros), el objeto que está ejecutando (`this`) se enviará un **mensaje a sí mismo**.

REFERENCIA THIS

Cliente

-username: String
-age: int
-phone: int
-password: String
-pet: Mascota

+getter y setters...

Mascota

-name: String
-age: int
-libreta:
LibretaSanitaria

+getter y setters...

REFERENCIA THIS

■ Estando en la clase **Cliente**

⚡ `public setAge(int age){
 this.age = age;
}`

REFERENCIAS

■ Asignación:

- ⚡ Usuario jeziel = new Usuario("Jezieel7", 123456, "contrasena123");
- ⚡ Usuario pepe= new Usuario("elPEPE", 364789, "eteeSech");
- ⚡ Usuario pepe2= new Usuario("Jezieel7", 123456, "contrasena123");
- ⚡ jeziel = pepe;

■ Comparación:

- ⚡ System.out.println(jeziel == pepe); ⚡ System.out.println(jeziel == pepe);
- ⚡ System.out.println(jeziel != pepe); ⚡ System.out.println(jeziel != pepe);
- ⚡ System.out.println(String1.equals(String2));

RECOLECTOR DE BASURA

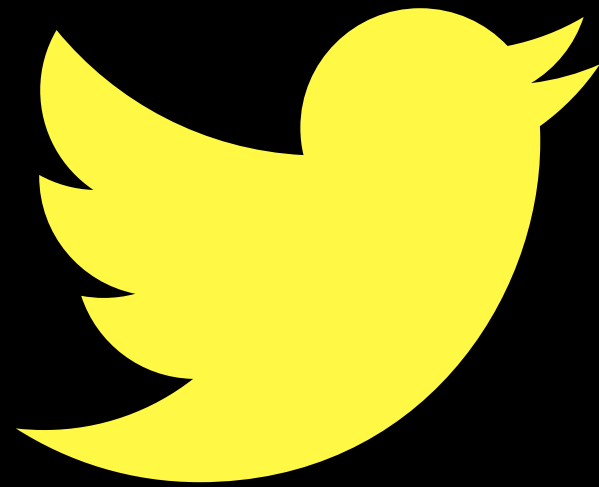
- Libera memoria de objetos no referenciados.

OBJETOS EN JAVA

- Java incluye bibliotecas de **clases** que permiten crear objetos de **uso común**.
 - ⚡ **String**, **Date**, **Math**, **Point2D.Double**, etc.

ACLARACIONES

- Los programas se organizan como una **colección de objetos** que cooperan entre sí enviándose mensajes.
- Los objetos se **crean** a medida que se **necesitan**.
- Cuando los objetos ya **no son necesarios** se **borran** de la memoria.
- El usuario le envía un mensaje a un objeto, en caso de que un objeto **requiera de otro**, puede enviarle un mensaje.



@Jezieel7



@Jezieel7