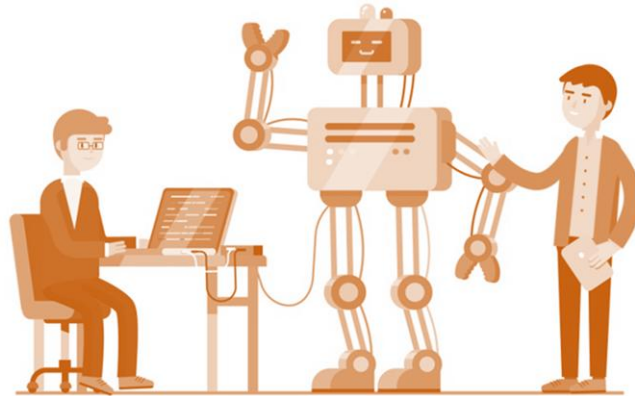


Unidad 3

Utilización de objetos y clases



Módulo Programación
modalidad @vanza

Unidad 03 Utilización de objetos y clases

1. Introducción
2. Fundamentos de la POO.
3. Clases y Objetos. Características de los objetos
4. Utilización de objetos
5. Utilización de métodos
6. Desarrollo de Clases
7. Librerías de objetos
8. Programación de la consola: E/S de la información

Unidad 03 Utilización de objetos y clases

1.- Introducción.

2.- Fundamentos de la POO.

2.1.- Conceptos.

La POO se centra en la relación que existe entre los datos y las acciones a realizar con ellos, y los encierra dentro del concepto de objeto, tratando de realizar una abstracción lo más cercana al mundo real.

2.2.- Beneficios

2.3.- Características.

- Abstracción
- Modularidad.
- Encapsulación.
- Jerarquía.
- Polimorfismo.

Unidad 03 Utilización de objetos y clases

3.- Clases y Objetos.

Un objeto es un conjunto de datos con las operaciones definidas para ellos. Los objetos tienen un estado y un comportamiento.

3.1.- Propiedades y métodos de los objetos.

- **Campos, Atributos o Propiedades:** Parte del objeto que almacena los datos. También llamadas Variables Miembro. Pueden ser de cualquier tipo primitivo (boolean, char, int, double, etc) o asu vez ser otro objeto.
Ejemplo, un objeto Coche puede tener un objeto Ruedas.
- **Métodos o Funciones Miembro:** Parte del objeto que lleva a cabo operaciones sobre los atributos definidos para ese objeto.

Unidad 03 Utilización de objetos y clases

3.- Clases y Objetos.

3.2.- Interacción entre objetos.

- Un mensaje es la acción que realiza un objeto.
- Un método es la función o procedimiento al que se llama para actuar sobre un objeto.
- Los objetos se pueden comunicar entre ellos invocando a los métodos de los otros objetos.

Unidad 03 Utilización de objetos y clases

3.- Clases y Objetos.

3.3.- Clases.

Una clase es una plantilla o prototipo donde se especifican:

- Los atributos (datos) comunes a todos los objetos de la clase.
- Los métodos que pueden utilizarse para manejar esos objetos.

Para declarar una clase se utiliza la palabra reservada `class`.

El método `main()` se utiliza para indicar que se trata de una clase principal, a partir de la cual va a empezar la ejecución del programa.

```
1  /*
2   * Estructura de una clase en Java
3   */
4
5  Cabecera de la clase
6  public class NombreClase {      Cuerpo de la clase
7      // Declaración de los atributos
8
9      // Declaración de los métodos
10
11     public static void main (String[] args) {
12         // Declaración de variables y/o constantes
13
14         // Instrucciones del método
15     }
16
17
18 }
19
```

Unidad 03 Utilización de objetos y clases

4.- Utilización de objetos.

Los objetos se crean a partir de las clases, y representan casos individuales de éstas.

4.1.- Ciclo de vida de los objetos.

4.2.- Declaración.

4.3.- Instanciación.

4.4.- Manipulación.

4.5.- Destrucción de objetos y liberación de memoria.

Unidad 03 Utilización de objetos y clases

5.- Utilización de métodos.

5.1.- Parámetros y valores devueltos. Cabecera de un método:

```
public tipo_dato_devuelto nombre_metodo (lista_param)
```

5.2.- Constructores. Método especial con el mismo nombre de la clase y que no devuelve ningún valor tras su ejecución.

5.3.- El operador this. Para referirse a los atributos de un objeto cuando estamos dentro de él.

5.4.- Métodos estáticos. Para llamar a un método estático usamos:

- El nombre del método, si lo llamamos desde la misma clase en la que está definido.
- El nombre de la clase, seguido de punto (.) más el nombre del método estático, si lo llamamos desde una clase distinta a la que se encuentra definido.
- El nombre del objeto, seguido de punto (.) más el nombre del método estático. Cuando tengamos un objeto instanciado de la clase en la que se encuentra definido el método estático, y no podamos utilizar la anterior.

Unidad 03 Utilización de objetos y clases

6.- Desarrollo de clases

6.1.- Estructura y miembros de una clase.

6.2.- Atributos

6.3.- Métodos

6.4.- Encapsulación, control de acceso y
visibilidad

6.5.- Constructores

Unidad 03 Utilización de objetos y clases

6.1.- Estructura y miembros de una clase.

6.1.1.- Declaración de una clase.

6.1.2.- Cabecera de una clase.

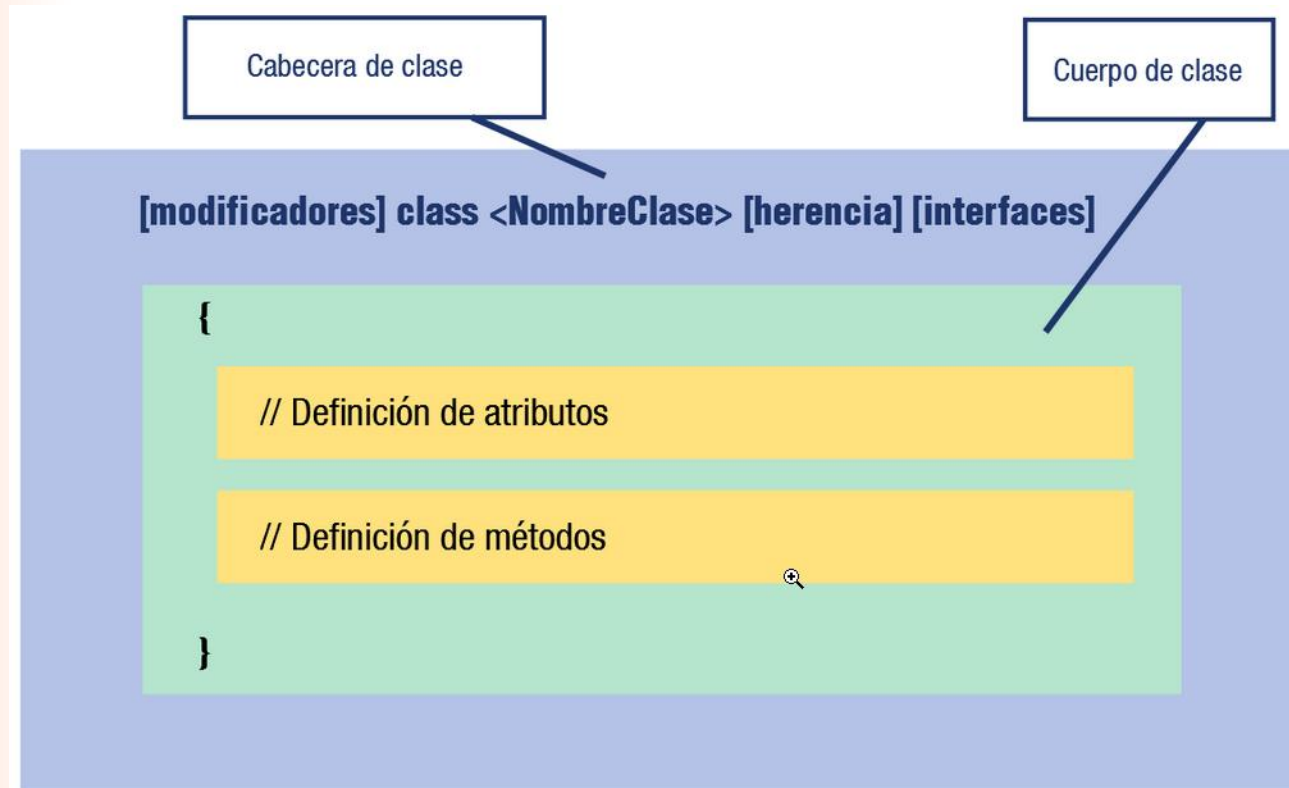
- public, abstract o final.
- implements y extends

6.1.3.- Cuerpo de una clase.

6.1.4.- Miembros estáticos o de clase. (static)

Unidad 03 Utilización de objetos y clases

6.1.- Estructura y miembros de una clase.



Unidad 03 Utilización de objetos y clases

6.2.- Atributos

Encapsulación, lo normal es declarar todos los atributos (o al menos la mayoría) como **privados (private)** de manera que **si se desea acceder o manipular algún atributo se tenga que hacer a través de los métodos proporcionados por la clase.**

6.2.1.- Declaración de atributos.

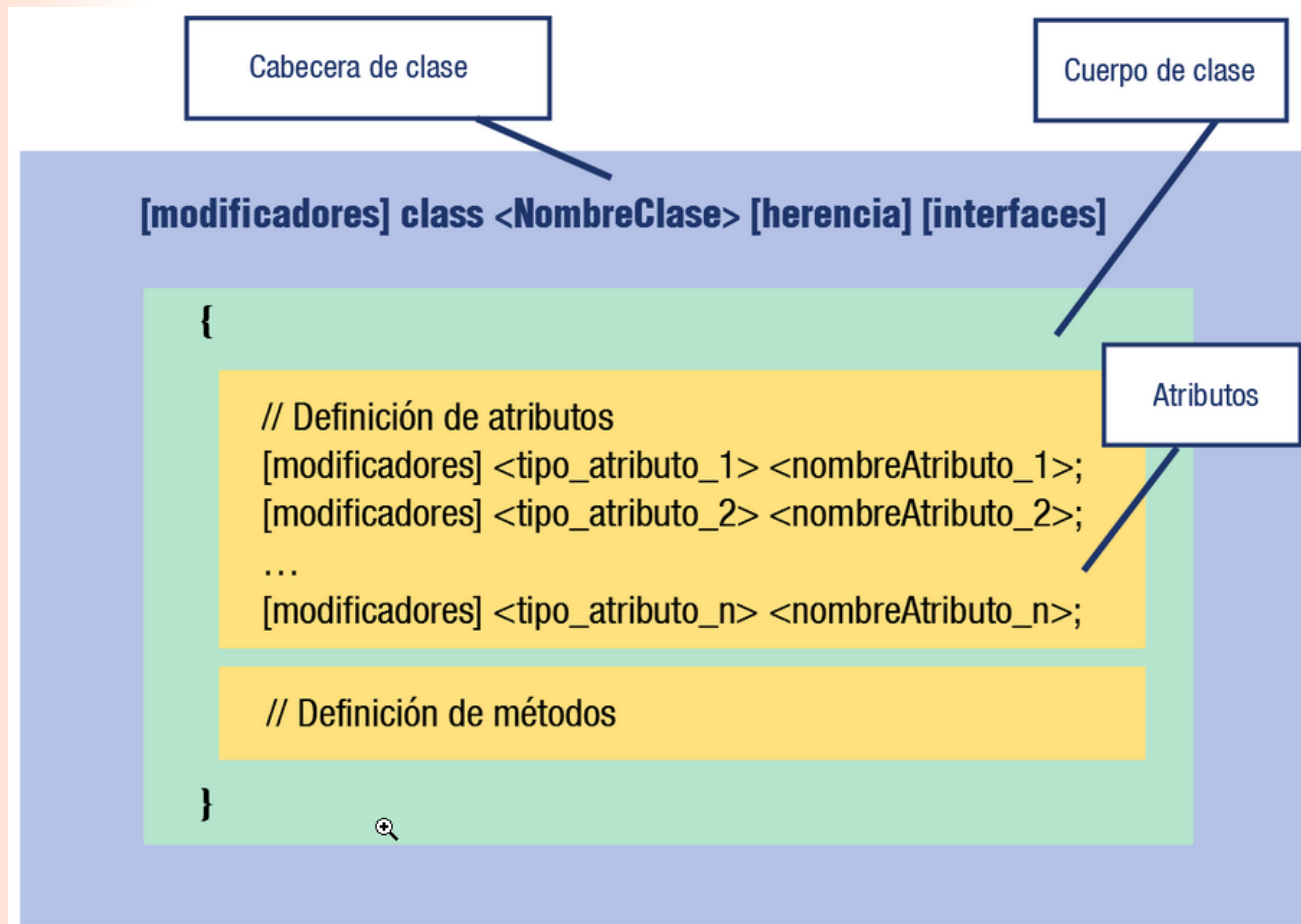
6.2.2.- Modificadores de acceso. (**public, private, protected**)

6.2.3.- Modificadores de contenido. (**static, final**)

6.2.4.- Atributos estáticos.

Unidad 03 Utilización de objetos y clases

Atributos



Unidad 03 Utilización de objetos y clases

6.3.- Métodos

6.3.1.- Declaración de un método

6.3.2.- Cabecera

6.3.3.- Modificadores

de acceso `public`, `private` y `protected`

de contenido `static` y `final`

de métodos `abstract`, `native`, `synchronized`

6.3.4.- Parámetros

6.3.5.- Cuerpo

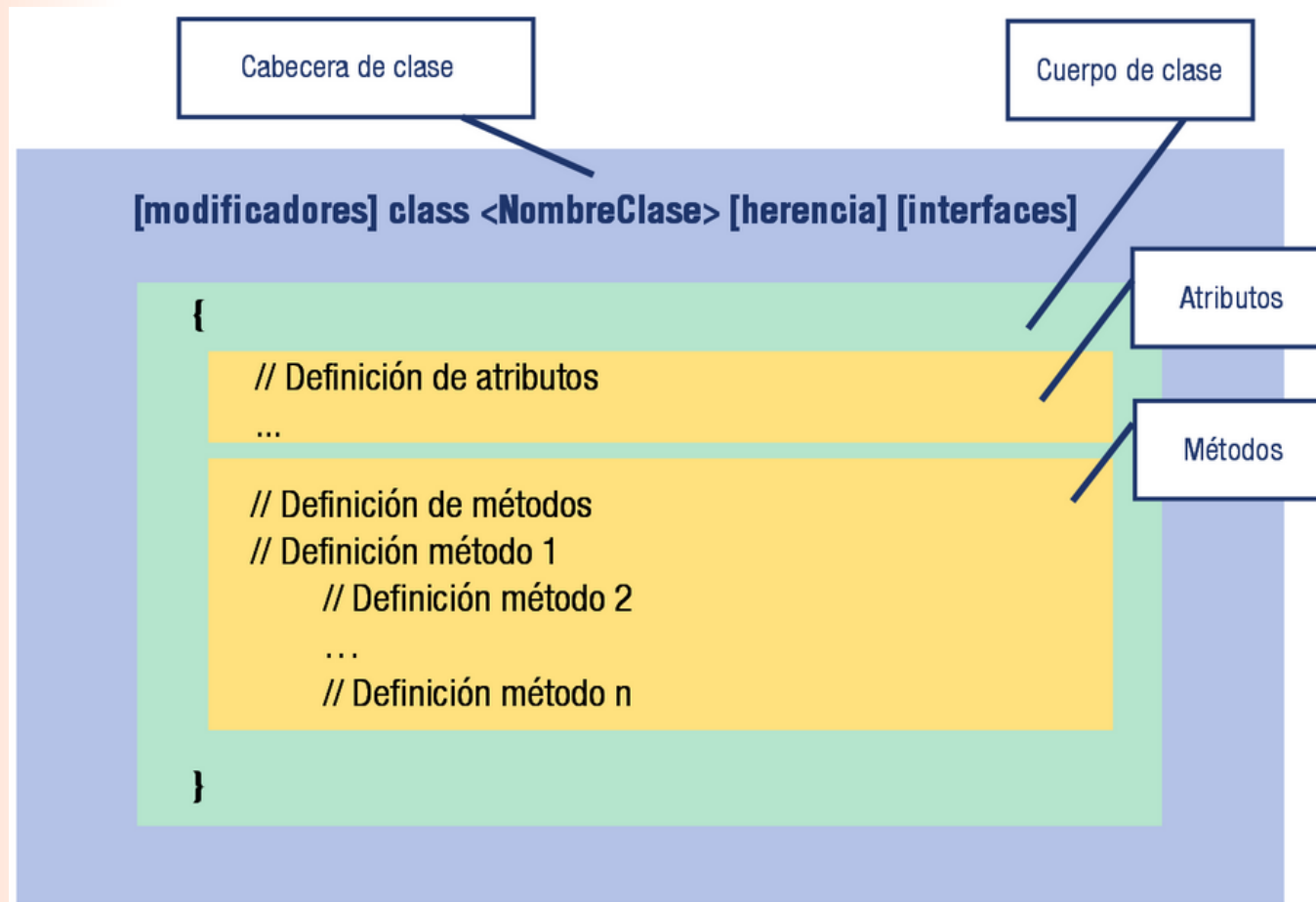
6.3.6.- Sobrecarga de métodos

6.3.7.- La referencia `this`.

6.3.8.- Métodos estáticos.

Unidad 03 Utilización de objetos y clases

Método



Unidad 03 Utilización de objetos y clases

6.4.- Encapsulación, control de acceso y visibilidad

- 6.4.1.- Ocultación de atributos

- 6.4.3.- Ocultación de métodos

6.5.- Constructores

- 6.5.1- Concepto

- 6.5.2.- Creación

- 6.5.3.- Utilización

- 6.5.4.- Constructores de copia

- 6.5.5.- Destrucción

Unidad 03 Utilización de objetos y clases

7.- Librerías de objetos (paquetes).

7.1.- Sentencia import.

Un **paquete** de clases es una agrupación de clases que consideramos que están relacionadas entre sí o tratan de un tema común.

Para usar una clase que está en un paquete distinto a la clase que estamos utilizando, se suele emplear la sentencia import.

7.2.- Compilar y ejecutar clases con paquetes.

7.3.- Jerarquía de paquetes.

7.4.- Librerías Java.

Unidad 03 Utilización de objetos y clases

8.- Entrada y salida de la información.

8.1.-Conceptos sobre la clase System.

```
InputStreamReader isr = new InputStreamReader(System.in);  
BufferedReader br = new BufferedReader (isr);
```

8.2.- Entrada por teclado. Clase System.

```
String cadena = br.readLine();
```

8.3.- Entrada por teclado. Clase Scanner.

```
Scanner teclado = new Scanner (System.in);  
int i = teclado.nextInt ();
```

8.4.- Salida por pantalla.

```
System.out.println("Bienvenido, " + nombre);
```

8.5.- Salida de error.

```
System.err.println("Salida")
```