

ÉRICA PALOMINO

TEMA 4

PROGRAMACIÓN ORIENTADA A OBJETOS

erica.palomino@escuelaartegranada.com

ESCUELAARTEGRANADA

INTRODUCCIÓN

La **Programación Orientada a Objetos (POO)** es un paradigma de diseño cuyo objetivo es “traducir” los elementos de la vida real a una estructura utilizable dentro de un lenguaje de programación.

Esta “traducción” se realiza mediante un proceso llamado **Abstracción**. Este proceso trata de obtener los **atributos y las acciones** básicas que representan la entidad que se quiere abstraer.

INTRODUCCIÓN

Los **Atributos** obtenidos mediante abstracción, al traducirlos a código, forman un **conjunto de variables** que otorgan las características a la clase.

Por otro lado, las **acciones** estarán representadas por los **métodos**; estructuras de código a las que se pueden referenciar en cualquier parte del código.



DECLARAR UNA CLASE

DECLARAR UNA CLASE

Para declarar una clase, crearemos un fichero java nuevo.

OJO. Para cada clase, se utilizará un fichero ÚNICO para esa clase.

La cabecera se construye de la siguiente forma:

`public class nombre {}`

Visibilidad palabra reservada Nombre de la clase



DECLARAR UNA CLASE - ESTRUCTURA

La estructura que vamos a seguir para construir una clase es la misma siempre.

```
public class nombre {  
    //Bloque de atributos  
  
    //Bloque de métodos  
}
```



DECLARAR ATRIBUTOS

DECLARAR ATRIBUTOS

Sigue una estructura similar a la cabecera de la clase:

```
public int nombre;  
private String nombre2;
```


DECLARAR ATRIBUTOS - VISIBILIDAD

Establece el acceso que tendrá el dato concreto, hay tres tipos distintos.

- **public.** lo ve todo el mundo.
- **private.** solo lo ve la propia clase.
- **protected.** visibilidad pública sólo entre los elementos del mismo paquete.



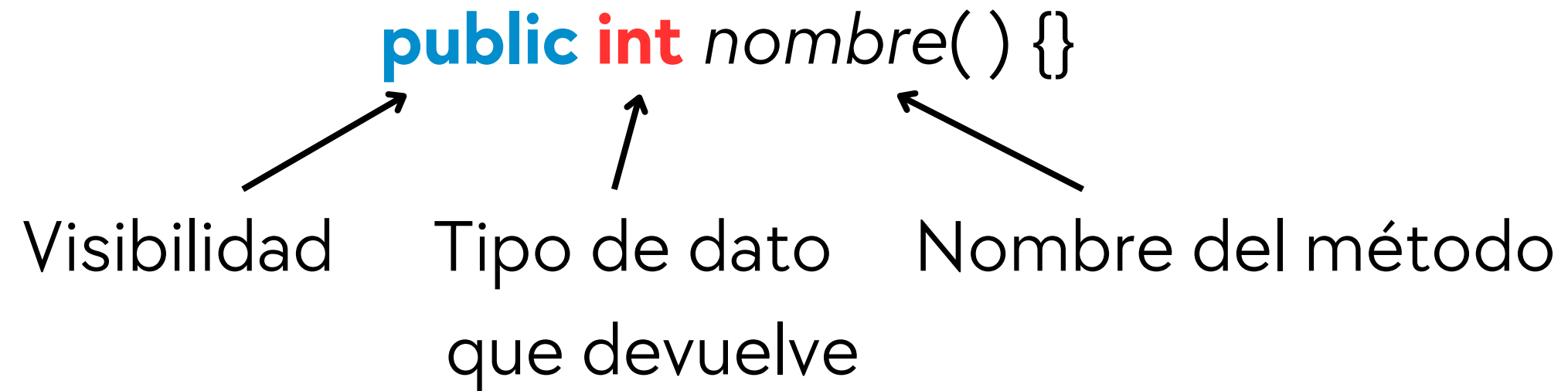
DECLARAR MÉTODOS

DECLARAR MÉTODOS

Al igual que el resto de elementos, sigue una estructura similar a la cabecera de la clase.

```
public int nombre() {}
```

Visibilidad Tipo de dato que devuelve Nombre del método



DECLARAR MÉTODOS

A la hora de declarar los métodos tenemos que tener en cuenta qué valor es el que va a devolver. Podemos hacernos una idea haciéndonos la pregunta **¿qué tipo de dato es el resultado de la ejecución del código?**

Para devolver el valor del resultado se utiliza la palabra reservada **return** seguida de la **variable que almacena el valor resultado. Solo se indica un return por método.**

Si el método no devuelve nada, se utiliza como tipo de dato **void**.

DECLARAR MÉTODOS

A veces, los métodos requieren de datos que se están utilizando en otra parte del código. Un bloque de código, no tiene acceso a las variables que se están utilizando en otro, por lo que se utiliza el **paso de mensajes**.

Esto no es más que indicarle al método correspondiente la **lista de variables** que se le tienen que proporcionar para poder funcionar.

DECLARAR MÉTODOS

Para indicarle esta lista de **parámetros** a los métodos, solo hay que introducir la lista entre los paréntesis de la cabecera, separados por comas.

```
public int nombre( tipo var1, tipo var2, ... ) {}
```

NOTA. El nombre de los parámetros y el de las variables que utilizemos a la hora de llamarlo **NO** tiene que tener el mismo nombre.

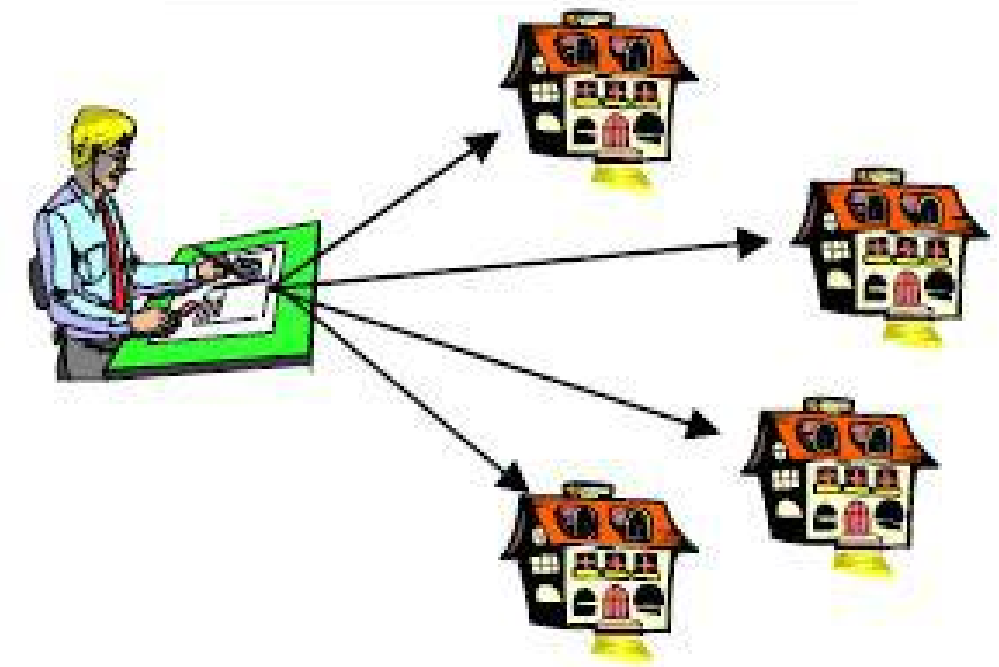




DIFERENCIA ENTRE CLASE Y OBJETO

DIFERENCIA ENTRE CLASE Y OBJETO

- **Clase:** Es la **plantilla** que utilizamos para construir los objetos. Indica los atributos que deben poseer los objetos y las acciones que pueden realizar, pero éstos **no tienen valores concretos**. Tampoco puede ejecutar los métodos.
- **Objeto:** Es un elemento creado a partir de la plantilla (clase) a la que pertenece. Los atributos que componen el objeto **tienen valores específicos** y puede llamar a los métodos.

OBJETOS Y CLASES





CONSTRUCTOR

CONSTRUCTOR

Para poder construir o **instanciar** un objeto de una clase concreta, la propia clase tiene que tener declarado un método llamado **constructor**. Este método se declara dándole visibilidad **pública** y utilizando como **nombre** de este, **el de la clase**. **No se le indica tipo de dato del return.**

```
public class Perro{  
    // Atributos  
  
    public Perro( ) { }  
}
```

CONSTRUCTOR

Dentro de los constructores, tenemos **dos** tipos principales, el **constructor por defecto** y el **constructor parametrizado**.

- **C. por defecto:** No admite parámetros. Todos los objetos que se creen utilizando este método **tendrán los mismo valores para los atributos**.
- **C. parametrizado:** Requiere que se le pasen una serie de parámetros para instanciar el objeto. Permite **inicializar** objetos con distintos valores para sus atributos.



INSTANCIAR UN OBJETO

INSTANCIAR UN OBJETO

Para instanciar un objeto vamos a seguir **los mismos pasos** que a la hora de declarar variables como la de **Scanner**.

Declaramos una variable, utilizando como **tipo de dato el nombre de la clase** y utilizando el **operador asignación =**, llamamos al constructor utilizando la palabra reservada **new**.

```
Perro toby = new Perro( );
```

```
Perro laika = new Perro( "Laika", 5 );
```



**ACCEDER A LOS
ELEMENTOS DEL
OBJETO**

ACCEDER A LOS ELEMENTOS DEL OBJETO

Para poder obtener los valores de los atributos del objeto, o ejecutar sus métodos, tendremos que referenciar a la variable que contiene el objeto y, **seguido de un punto**, indicar el elemento al que se quiere llamar (**como hemos hecho con Scanner**).

```
laika.nombre; //Obtengo el valor del atributo  
laika.ladRAR(); //Ejecuto el método ladRAR
```