

### 3.D. Métodos.

✓ Hecho

## 1. Utilización de métodos.

### 1.3. El operador this.

Los constructores y métodos de un objeto suelen utilizar el operador `this`. Este operador sirve para referirse a los atributos de un objeto cuando estamos dentro de él. Sobre todo se utiliza cuando existe ambigüedad entre el nombre de un parámetro y el nombre de un atributo, entonces en lugar del nombre del atributo solamente escribiremos `this.nombre_atributo`, y así no habrá duda de a qué elemento nos estamos refiriendo.

Vamos a ilustrar mediante un ejemplo la utilización de objetos y métodos, así como el uso de parámetros y el operador `this`. Aunque la creación de clases la veremos en las siguientes unidades, en este ejercicio creamos una pequeña clase para que podamos instanciar el objeto con el que vamos a trabajar.

Las clases se suelen representar como un rectángulo, y dentro de él se sitúan los atributos y los métodos de dicha clase.

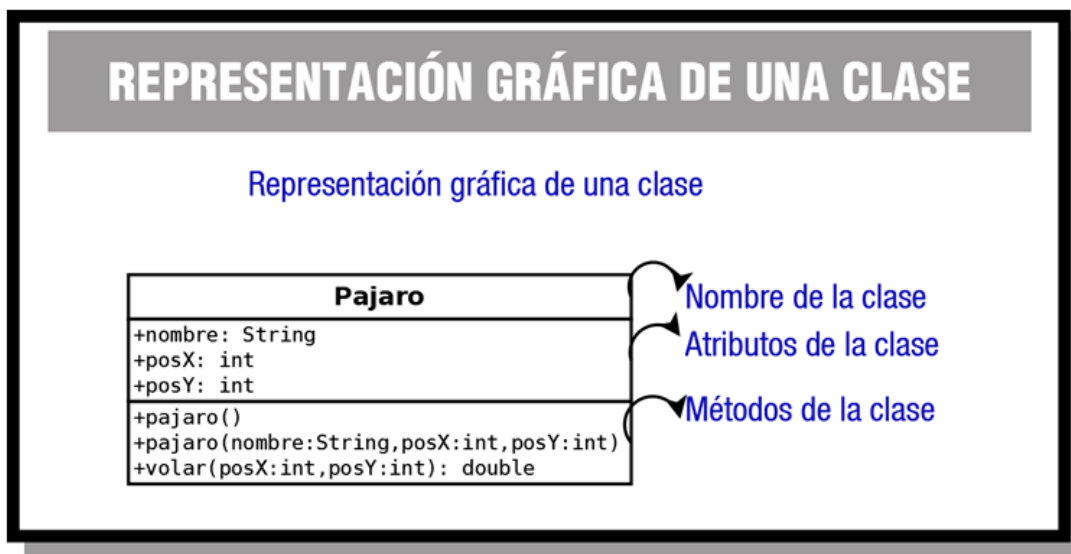


Imagen extraída de curso Programación del MECD.

En la imagen, la clase `Pajaro` está compuesta por tres atributos, uno de ellos el nombre y otros dos que indican la posición del ave, `posX` y `posY`. Tiene dos métodos constructores y un método `volar()`. Como sabemos, los métodos constructores reciben el mismo nombre de la clase, y puede haber varios para una misma clase, dentro de ella se diferencian unos de otros por los parámetros que utilizan.

#### Ejercicio resuelto

Dada una clase principal llamada `Pajaro`, se definen los atributos y métodos que aparecen en la imagen. Los métodos realizan las siguientes acciones:

- `pajaro()`. Constructor por defecto. En este caso, el constructor por defecto no contiene ninguna instrucción, ya que Java inicializa de forma automática las variables miembro, si no le damos ningún valor.
- `pajaro(String nombre, int posX, int posY)`. Constructor que recibe como argumentos una cadena de texto y dos enteros para inicializar el valor de los atributos.
- `volar(int posX, int posY)`. Método que recibe como argumentos dos enteros: `posX` y `posY`, y devuelve un valor de tipo `double` como resultado,

usando la palabra clave return. El valor devuelto es el resultado de aplicar un desplazamiento de acuerdo con la siguiente fórmula:  
$$\text{desplazamiento} = \sqrt{\text{posX} \cdot \text{posX} + \text{posY} \cdot \text{posY}}$$
 Diseña un programa que utilice la clase **Pajaro**, cree una instancia de dicha clase y ejecute sus métodos.

#### Solución:

Lo primero que debemos hacer es crear la clase Pajaro, con sus métodos y atributos. De acuerdo con los datos que tenemos, el código de la clase sería el siguiente:

```
public class Pajaro {  
  
    String nombre;  
    int posX, posY;  
  
    public Pajaro() {  
  
    }  
  
    public Pajaro(String nombre, int posX, int posY) {  
        this.nombre=nombre;  
        this.posX=posX;  
        this.posY=posY;  
    }  
    double volar (int posX, int posY) {  
  
        double desplazamiento = Math.sqrt( posX*posX + posY*posY );  
        this.posX = posX;  
        this.posY = posY;  
  
        return desplazamiento;  
    }  
}
```

Debemos tener en cuenta que se trata de una clase principal, lo cual quiere decir que debe contener un método main() dentro de ella. En el método main() vamos a situar el código de nuestro programa. El ejercicio dice que tenemos que crear una instancia de la clase y ejecutar sus métodos, entre los que están el constructor y el método volar(). También es conveniente imprimir el resultado de ejecutar el método volar(). Por tanto, lo que haría el programa sería:

- **Crear** un objeto de la clase e inicializarlo.
- **Invocar** al método volar.
- **Imprimir** por pantalla la distancia recorrida.

Para inicializar el objeto utilizaremos el constructor con parámetros, después ejecutaremos el método volar() del objeto creado y finalmente imprimiremos el valor que nos devuelve el método. El código de la clase main() quedaría como sigue:

```
public static void main(String[] args) {  
  
    Pajaro loro = new Pajaro("Lucy",50,50) ;  
    double d = loro.volar(50,50);  
    System.out.println("El desplazamiento ha sido " + d);  
}
```

Si ejecutamos nuestro programa el resultado sería el siguiente:

```
Output - UtilizarObjetos (run)  
run:  
El desplazamiento ha sido 70.71067811865476  
BUILD SUCCESSFUL (total time: 0 seconds)
```

El contenido completo del fichero será:

```
/*
```

```
* Programa que simula el comportamiento de un Pajaro
```

```
*/
```

```
/**
```

```
*
```

```
* @author FMA
```

```
*/
```

```
public class Pajaro {
```

```
    String nombre;
```

```
    int posX, posY;
```

```
    public Pajaro() {
```

```
    }
```

```
    public Pajaro(String nombre, int posX, int posY) {
```

```
        this.nombre=nombre;
```

```
        this.posX=posX;
```

```
        this.posY=posY;
```

```
    }
```

```
    double volar (int posX, int posY) {
```

```
        double desplazamiento = Math.sqrt( posX*posX + posY*posY );
```

```
        this.posX = posX;
```

```
        this.posY = posY;
```

```
        return desplazamiento;
```

```
    }
```

```
    public static void main(String[] args) {
```

```
        Pajaro loro = new Pajaro("Lucy",50,50) ;
```

```
        double d = loro.volar(50,50);
```

```
        System.out.println("El desplazamiento ha sido " + d);
```

```
    }
```

◀ 3.C. Trabajando con objetos.

Ir a...

3.E. Librerías y paquetes. ▶