

Operador Instanceof en Java

El operador **instanceof** nos permite comprobar si un objeto es de una **clase concreta**.

Por ejemplo, tenemos 3 clases: Empleado (clase padre), Comercial (clase hija de Empleado) y Repartidor (clase hija Empleado).

```
1 public class EmpleadoApp {
2     public static void main(String[] args) {
3
4         Empleado empleados[]=new Empleado[3];
5         empleados[0]=new Empleado();
6         empleados[1]=new Comercial();
7         empleados[2]=new Repartidor();
8
9         for(int i=0;i<empleados.length;i++){
10            if(empleados[i] instanceof Empleado){
11                System.out.println("El objeto en el indice "+i+" es de la clase Empleado");
12            }
13            if(empleados[i] instanceof Comercial){
14                System.out.println("El objeto en el indice "+i+" es de la clase Comercial");
15            }
16            if(empleados[i] instanceof Repartidor){
17                System.out.println("El objeto en el indice "+i+" es de la clase Repartidor");
18            }
19        }
20    }
21 }
22 }
23 }
24 }
25 }
```

Si ejecutas el ejemplo anterior, veras que cuando las **clases hijas** además de mostrar que es de su propia clase, también son de la clase Empleado, ya que, por ejemplo, un comercial “es un” empleado.

¿Pero que pasaría si quisiera ejecutar un **método** definido en la clase hija (que en la **clase padre** no esta)?

Debemos hacer un **casting de objetos** para poder ejecutarlo.

Veamos un ejemplo:

```
1 public class EmpleadoApp {
2
3     public static void main(String[] args) {
4
5         Empleado empleados[]=new Empleado[4];
6         empleados[0]=new Comercial("Fernando", "Ureña", 23, 700, 50);
7         empleados[1]=new Repartidor("Antonio", "Lopez", 30, 800, "Este");
8         empleados[2]=new Comercial("Laura", "Perez", 24, 1000, 30);
9         empleados[3]=new Repartidor("Rosa", "Serrano", 43, 1200, "Oeste");
10
11        for(int i=0;i<empleados.length;i++){
12            //El array es de objetos Empleados
13            if(empleados[i] instanceof Comercial){
14                //Si la posición del array es un comercial, hacemos un casting de objetos
15                Comercial ref=(Comercial)empleados[i];
16                //Ahora podemos invocar métodos del objeto
17                System.out.println("La comision es de "+ref.getComision());
18            }
19        }
20    }
21 }
```

```

17     }
18     if(empleados[i] instanceof Repartidor){
19         //Si la posición del array es un repartidor, hacemos un casting de objetos
20         Repartidor ref=(Repartidor)empleados[i];
21         //Ahora podemos invocar métodos del objeto
22         System.out.println("La zona es "+ref.getZona());
23     }
24 }
25 }
26 }
27 }
28

```

Como vemos, para hacer un **casting de objetos**, creamos un objeto aparte de la clase que queremos hacer el casting y convertimos la posición del array ya que el **array** considera que contiene objetos de la **clase Empleado**.

Recuerda que si ese método está tanto en la clase padre como en la hija, no es necesario hacer el casting de objetos.