

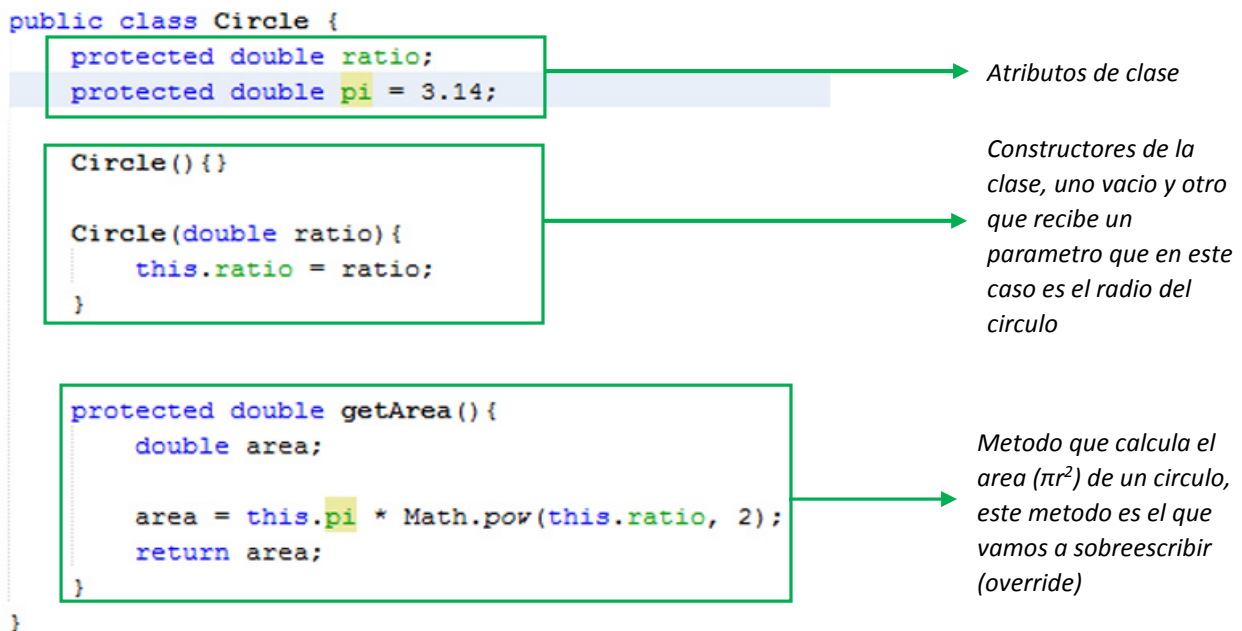
¿Cuándo utilizarlos?

Override.

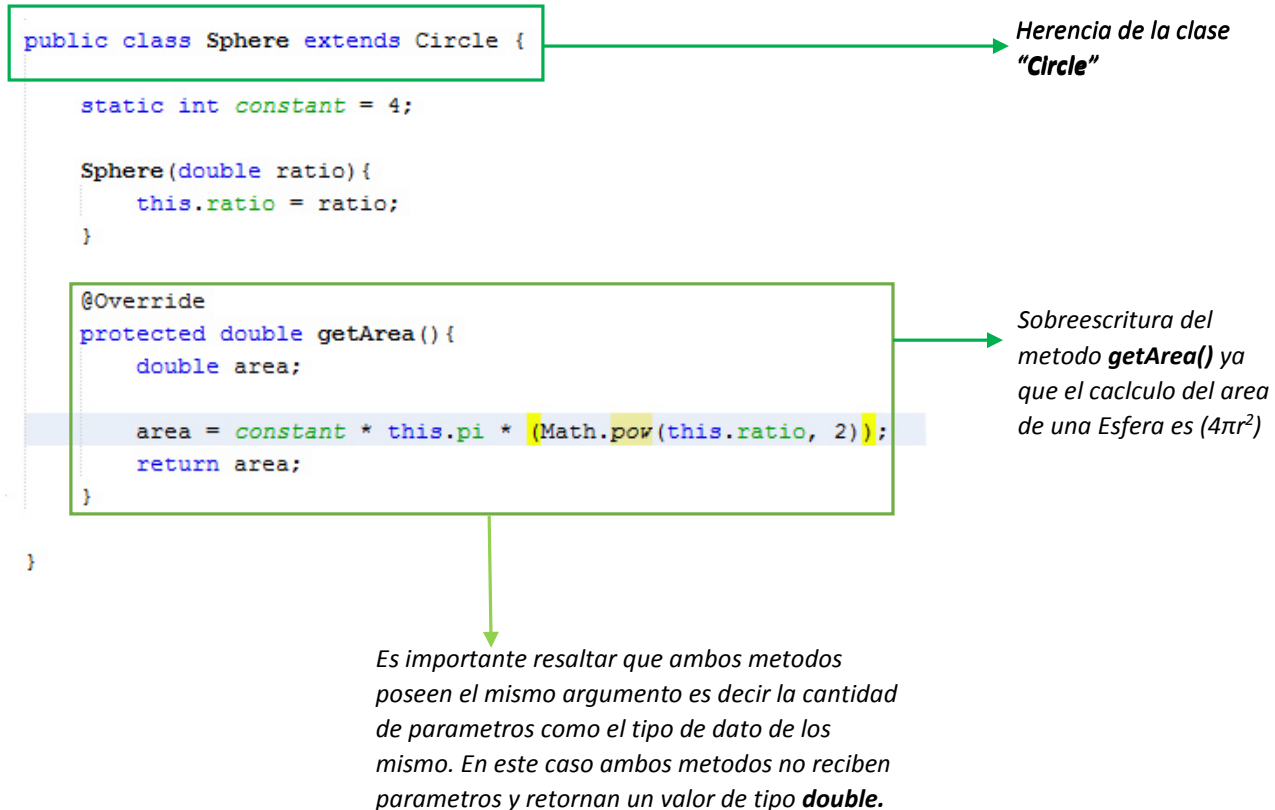
La sobrescritura (override) se utiliza a través de la herencia, ya que este se da únicamente cuando un método heredado dentro de una clase hija, posee las mismas características que el método de la clase padre; (cantidad de parámetros, tipos de datos, y tipo de dato de retorno del método) y lo único que cambia es la definición de dicho método.

Ejemplo:

Para este ejemplo supongamos que tenemos una clase llamada “**Circle**” en la cual existe un método llamado “**getArea()**” el cual calcula el área de un círculo.



Ahora bien supongamos que tenemos otra clase que se llama “**Sphere**” la cual define una esfera que vendria siendo un circulo en 3 dimensiones, por lo que hereda de la clase “**Circle**”, en dicha herencia todos los atributos y metodos de la clase “**Circle**” seran heredados a la clase “**Sphere**”, incluyendo el metodo “**getArea()**” el cual calcula el area del ciculo, pero para el caso de la esfera el calculo se realiza de forma distinta, por que convendria sobrescribir el metodo y quedaria de la siguiente manera.



Overload.

La sobrecarga (overload), a diferencia de la sobreescritura, se utiliza cuando necesitamos que un metodo dentro de una misma clase, realice la misma funcionalidad pero recibiendo parametros distintos (cantidad y tipos de datos) y asi como tambien retorne un tipo de dato distinto, en sus diferentes definiciones.

Ejemplo:

Para este caso supongamos que tenemos una clase que se llama **“Figure”** la cual posee un metodo **“getArea()”** el cual calcula el area de un cuadrado o de un rectangulo dependiendo de la cantidad y el tipo de parametros que tenga el argumento de dicho metodo.

```
public class Figure {
```

```
    Figure() {}
```

```
    public double getArea(double length, double width){
        double area;

        area = length * width;
        return area;
    }
```

Metodo para calcular el area de un rectangulo, el cual recibe 2 parametros en su argumento el lado **“length”** y el ancho **“width”** para poder realizar el calculo.

```
    public double getArea(double length){
        double area;
        area = Math.pow(length, 2);

        return area;
    }
```

Metodo para calcular el area de un cuadrado, el cual solo recibe un parametro en su argumento en este caso es el lado **“length”** para poder realizar el calculo.

```
}
```

Es importante resaltar que ambos metodos se encuentran dentro de la misma clase, y los que los diferencia es su argumento en este caso, ya que dependiendo de los parametros que reciba, es donde se define que metodo se ejecuta

Diferencias entre Override y Overload

Override (SobreEscritura)	Overload (SobreCarga)
La firma del metodo(numero de parametros, tipos de datos etc.) y tipo de dato de retorno del mismo tienen que ser iguales (Lo cual impulsa al concepto de Polimorfismo).	La firma del metodo (numero de parametros, tipos de datos etc.) tienen que ser distintos para poder sobrecargar un metodo.
Solo ocurre si un metodo heredado en la clase hija necesita una definicion distinta que la que se encuentra especificada en la clase padre.	Solo ocurre si un metodo dentro de una misma clase necesita una definicion diferente.
Ocorre en tiempo de ejecucion del Sistema	Ocorre o se ejecuta al en tiempo de compilacion de la aplicacion.
El constructor de una clase no puede ser sobreescrito	Se puede sobrecargar un constructor
Un metodo estatico (static) no puede ser sobreescrito	Cualquier metodo puede ser sobrecargado