



Métodos estáticos

▼ Declaración y uso de métodos estáticos

Según los conceptos de POO (Programación Orientada a Objetos) los atributos que se definen en una clase reservan espacio de forma independiente para cada instancia de la misma.

A diferencia de los anteriores, los atributos estáticos tienen un comportamiento muy distinto a los atributos vistos hasta el momento. Un atributo estático reserva espacio para el mismo indistintamente que definamos un objeto de dicha clase. En caso de crear varios objetos de dicha clase, todas las instancias acceden al mismo atributo estático.

Para declarar un atributo de tipo estático agregamos el modificador 'static' en su definición:

```
public class Matematica {  
    public static float PI=3.1416f;  
}  
public class Prueba {  
    public static void main(String[] args) {  
        System.out.println(Matematica.PI);  
    }  
}
```

La clase Matematica define un atributo estático:

```
public static float PI=3.1416f;
```

Luego podemos acceder a dicho atributo directamente a través del nombre de la clase sin tener que crear un objeto de la clase 'Matematica':

```
System.out.println(Matematica.PI);
```

▼ Ejemplo

Definir un atributo estático que almacene la cantidad de objetos creados de dicha clase.

```
public class Persona {
    private String nombre;
    private int edad;
    public static int cantidad;
    public Persona(String nombre,int edad) {
        this.nombre=nombre;
        this.edad=edad;
        cantidad++;
    }
    public void imprimir() {
        System.out.println(nombre+" "+edad);
    }
}

public class PruebaPersona {
    public static void main(String[] args) {
        System.out.println("Valor del atributo estático cantidad:"+Persona.cantidad);
        Persona per1=new Persona("Juan",30);
        per1.imprimir();
        System.out.println("Valor del atributo estático cantidad:"+Persona.cantidad);
        Persona per2=new Persona("Ana",20);
        per2.imprimir();
        System.out.println("Valor del atributo estático cantidad:"+Persona.cantidad);
        Persona per3=new Persona("Luis",10);
        per3.imprimir();
        System.out.println("Valor del atributo estático cantidad:"+Persona.cantidad);
    }
}
```

Un atributo estático de tipo entero se inicializa en cero cuando lo definimos (lo definimos de tipo public para poder acceder a su valor desde fuera de la clase).

Luego podemos consultarlo inclusive antes de crear un objeto de dicha clase:

```
System.out.println("Valor del atributo estático cantidad:"+Persona.cantidad);
```

Cada vez que creamos un objeto de la clase Persona:

```
Persona per1=new Persona("Juan",30);
```

Se incrementa en uno el atributo estático 'cantidad':

```
public Persona(String nombre,int edad) {
    this.nombre=nombre;
    this.edad=edad;
    cantidad++;
}
```

No importa cuantos objetos de la clase Persona se creen luego, existe un solo atributo cantidad:

```
Persona per1=new Persona("Juan",30);  
Persona per2=new Persona("Ana",20);  
Persona per3=new Persona("Luis",10);
```

Creamos 3 objetos de la clase Persona y los tres comparten el mismo atributo 'cantidad'.

Un atributo estático se lo puede acceder tanto a través del nombre de la clase como a través del nombre de un objeto:

```
Persona per1=new Persona("Juan",30);  
System.out.println("Valor del atributo estático cantidad:"+Persona.cantidad);  
System.out.println("Valor del atributo estático cantidad:"+per1.cantidad);
```

▼ Métodos estáticos de una clase

Así como una clase puede tener atributos estáticos, Java también permite definir métodos estáticos que se crean independientemente a la definición de objetos. Un método estático puede llamarse sin tener que crear un objeto de dicha clase. Igual que los atributos estáticos, un método estático tiene ciertas restricciones:

- No puede acceder a los atributos de la clase (salvo que sean estáticos)
- No puede utilizar el operador this, ya que este método se puede llamar sin tener que crear un objeto de la clase.
- Puede llamar a otro método siempre y cuando sea estático.

Si recordamos cada vez que creamos un programa en Java debemos especificar el método main:

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

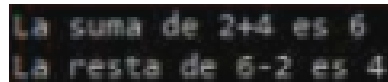
El método main es estático para que la máquina virtual de Java pueda llamarlo directamente sin tener que crear un objeto de la clase que lo contiene.

▼ Ejemplo

Ejemplar: Implementar una clase llamada Operacion. Definir dos métodos estáticos que permitan sumar y restar dos valores enteros.

```
public class Operacion {  
    public static int sumar(int x1,int x2) {  
        int s=x1+x2;  
        return s;  
    }  
    public static int restar(int x1,int x2) {  
        int r=x1-x2;  
        return r;  
    }  
}  
public class PruebaOperacion {  
    public static void main(String[] args) {  
        System.out.print("La suma de 2+4 es ");  
        System.out.println(Operacion.sumar(2,4));  
        System.out.print("La resta de 6-2 es ");  
        System.out.println(Operacion.restar(6,2));  
    }  
}
```

La ejecución del proyecto genera una salida:

A screenshot of a terminal window showing the output of the Java program. The text is displayed on two lines: "La suma de 2+4 es 6" and "La resta de 6-2 es 4". The background is dark, and the text is light-colored.

```
La suma de 2+4 es 6  
La resta de 6-2 es 4
```

No es obligatorio que una clase defina todos sus métodos estáticos, veremos con un ejemplo como podemos tener métodos estáticos y no estáticos.