

Programación Java

Tutorial Java. Aprende a programar con Java desde cero.



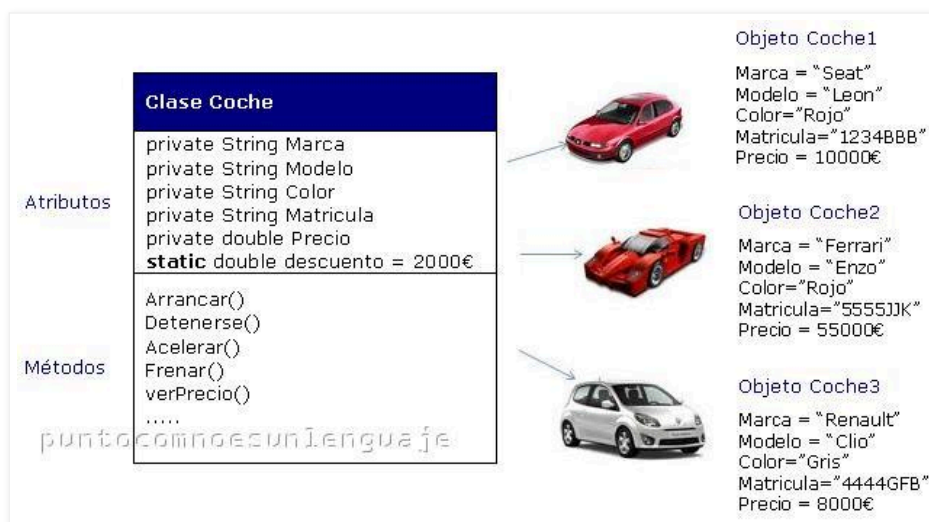
Última entrada Teoría Ejercicios Ejercicios POO C++

Java static. Atributos y métodos estáticos o de clase

Los atributos y métodos estáticos también se llaman **atributos de clase** y **métodos de clase**.

Se declaran como **static**.

Supongamos una clase Coche sencilla que se utiliza en un programa de compra-venta de coches usados. De esta clase se han creado tres objetos de tipo Coche:



La clase contiene 6 atributos: marca, modelo, color, matrícula, precio y descuento. Supongamos que el descuento es una cantidad que se aplica a todos los coches sobre el precio de venta. Como este dato es el mismo para todos los coches y es un valor que se puede modificar en cualquier momento no debe formar parte de cada coche sino que es un dato que deben compartir todos. Esto se consigue declarándolo como atributo **static**.

Un Atributo static:

- No es específico de cada objeto. Solo hay una copia del mismo y su valor es compartido por todos los objetos de la clase.
- Podemos considerarlo como una *variable global* a la que tienen acceso todos los objetos de la clase.
- Existe y puede utilizarse aunque no existan objetos de la clase.

Para acceder a un atributo de clase se escribe:

NombreClase.atributo

Un Método static:

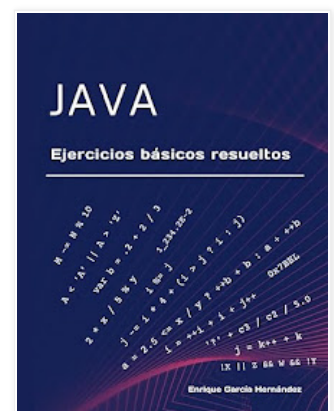
- Tiene acceso solo a los atributos estáticos de la clase.
- No es necesario instanciar un objeto para poder utilizarlo.

Para acceder a un método de clase se escribe:

NombreClase.método()

Ejemplo:

PUBLICACIONES DEL BLOG



JAVA - Ejercicios básicos resueltos



ENTRADAS POPULARES

Vamos a escribir una clase Persona que contendrá un atributo contadorPersonas que indique cuantos objetos de la clase se han creado.

contadorPersonas debe ser un atributo de clase ya que no es un valor que se deba guardar en cada objeto Persona que se crea, por lo tanto se debe declarar static:

```
public class Persona {

    private String nombre;
    private int edad;
    static int contadorPersonas;
    .....

}
```

Un ejemplo de uso desde fuera de la clase Persona podría ser mostrar su valor en la clase principal:

```
System.out.println(Persona.contadorPersonas);
```

También podríamos modificar su valor en la clase principal de esta forma:

```
Persona.contadorPersonas++;
```

Si contadorPersonas lo declaramos como atributo **private** solo podremos acceder al atributo desde fuera de la clase a través de métodos static. En ese caso podemos tener métodos get y set para acceder a contadorPersonas que sean static además de un método incrementarContador también static:

```
public class Persona {

    private String nombre;
    private int edad;
    private static int contadorPersonas;

    .....

    public static int getContadorPersonas() {
        return contadorPersonas;
    }

    public static void setContadorPersonas(int contadorPersonas) {
        Persona.contadorPersonas = contadorPersonas;
    }

    public static void incrementarContador(){
        contadorPersonas++;
    }

    . . . . .

}
```

En este caso, para acceder desde fuera de la clase al contador de personas debemos usar:

Persona.metodoStatic

Por ejemplo:

```
System.out.println(Persona.getContadorPersonas());
```

La clase persona completa sería esta:



Java Ejercicios Básicos Resueltos 1
Relación Nº 1:
Ejercicios 1, 2 y 3
Empezaremos por unos ejercicios básicos de programas Java con estructura secuencial, es decir, en es...

[Calcular el factorial de un número en Java](#)

Factorial en java Programa que calcule el factorial de un número entero que se introduce por teclado. El factorial de un número se expresa m...

[Leer un char por teclado en Java](#)

Cómo leer un carácter desde teclado en Java La clase Scanner NO CONTIENE un método nextChar() para leer un dato de tipo char desde tecl...



Java Quicksort
El método de ordenación QuickSort fue desarrollado por Hoare en el año 1960. Es el algoritmo de ordenación más rápido. Se basa en la té...



Java printf para dar formato a los datos de salida
Vamos a ver como utilizar printf para dar formato a los datos que se imprimen por pantalla en Java. Este problema se nos plantea por ejempl...

[Torres de Hanoi en Java](#)

La leyenda. En una antigua ciudad en la India, los monjes en un templo tienen que mover una pila de 64 discos sagrados de un lugar a ot...



Programación Java
Enrique García Hernández

SEGUIDORES

TRANSLATE

LENGUAJE C++

Programacion C++
Números amigos en C++

```
//Clase Persona
public class Persona {

    private String nombre;
    private int edad;
    private static int contadorPersonas;

    public Persona() {
    }

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

    public void setNombre(String nom) {
        nombre = nom;
    }

    public String getNombre() {
        return nombre;
    }

    public void setEdad(int ed) {
        edad = ed;
    }

    public int getEdad() {
        return edad;
    }

    public static int getContadorPersonas() {
        return contadorPersonas;
    }

    public static void setContadorPersonas(int contadorPersonas) {
        Persona.contadorPersonas = contadorPersonas;
    }

    public static void incrementarContador() {
        contadorPersonas++;
    }
}
```

Ejemplo de uso de la clase Persona:

```
//Clase Principal
public class Estatico1 {
    public static void main(String[] args) {
        Persona p1 = new Persona("Tomás Navarra", 22);
        Persona.incrementarContador();
        Persona p3 = new Persona("Jonás Estacio", 23);
        Persona.incrementarContador();
        System.out.println("Se han creado: " + Persona.getContadorPersonas() + " personas");
    }
}
```

Vemos en el ejemplo que cada vez que se crea un nuevo objeto se incrementa el contador invocando al método incrementarContador().

En lugar de invocar al método incrementarContador() cada vez que se crea un objeto, es más correcto hacer el incremento de la variable estática directamente en el constructor.

El código de la clase Persona y de la clase principal quedaría ahora así:

```
//Clase Persona
public class Persona {

    private String nombre;
    private int edad;
    private static int contadorPersonas;

    public Persona() {
        contadorPersonas++;
    }

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

    public void setNombre(String nom) {
        nombre = nom;
    }

    public String getNombre() {
        return nombre;
    }

    public void setEdad(int ed) {
        edad = ed;
    }

    public int getEdad() {
        return edad;
    }

    public static int getContadorPersonas() {
        return contadorPersonas;
    }

    public static void setContadorPersonas(int contadorPersonas) {
        Persona.contadorPersonas = contadorPersonas;
    }

}

//Clase Principal
public class Estatico1 {
    public static void main(String[] args) {
        Persona p1 = new Persona("Tomás Navarra", 22);
        Persona p3 = new Persona("Jonás Estacio", 23);
        System.out.println("Se han creado: " + Persona.getContadorPersonas() + " personas");
    }
}
```



Si te ha sido útil compártelo
Post

19 comentarios:



Unknown 11 de marzo de 2022, 16:00

Muchas gracias, excelente servicio.

[Responder](#)



Pepe Argento 10 de mayo de 2019, 15:04

Excelente ejemplo. Me aclaró algunas dudas que tenía.

[Responder](#)



EXCEL_2012 11 de noviembre de 2017, 17:09

como puedo contar los objetos pero sin pasar parametros al objeto..?

[Responder](#)



Unknown 23 de octubre de 2017, 1:58

Que significa si se hace la clase del tipo static

Es decir;

```
public static class ClaseEstatica{
```

```
// Código a realizar
```

```
// Me llamo la atención que la clase base no puede ser estática pero las interna si!!
```

```
}
```

[Responder](#)



jacv 16 de diciembre de 2016, 1:52

Amigo una pregunta: ¿Para que crear todos esos setter y getter si no se van a usar?

[Responder](#)

[Respuestas](#)

[Responder](#)



Enrique 16 de diciembre de 2016, 8:10

Para este ejemplo en concreto no haría falta crearlos ya que no se utilizan cuando se usa la clase.



Anónimo 2 de septiembre de 2016, 1:10

no me ha quedado claro por que a veces se puede acceder a un metodo estatico desde otro metodo estatico como clase.metodo().metodo() alguien me lo puede explicar por favor se los agradeceria mucho

[Responder](#)



Unknown 31 de mayo de 2016, 16:43

Como que no lo he digerido bien...

[Responder](#)

[Respuestas](#)

[Responder](#)



Anónimo 16 de septiembre de 2016, 7:58

<http://www.discoduroderoer.es/metodos-y-atributos-estaticos/>



Anónimo 8 de noviembre de 2015, 20:07

buen ejemplo

[Responder](#)



Anónimo 12 de octubre de 2015, 4:57

Confuso, confuso. hay que decodificarlo y entenderlo. Gracias.

[Responder](#)



Anónimo 31 de agosto de 2015, 20:04

buen ejemplo gracias

[Responder](#)



Anónimo 15 de julio de 2015, 16:43

Gracias ! :D

[Responder](#)

Anónimo 8 de febrero de 2015, 5:49

Excelente :D

[Responder](#)



juanitocaspi 8 de febrero de 2015, 5:48

Este comentario ha sido eliminado por el autor.

[Responder](#)

Anónimo 5 de febrero de 2015, 13:57

muy buen ejemplo me sirvió mucho.. :)

[Responder](#)

Anónimo 17 de enero de 2015, 18:06

Buen ejemplo :D

[Responder](#)

Anónimo 5 de diciembre de 2013, 20:44

muy buen ejemplo. gracias

[Responder](#)

Anónimo 3 de diciembre de 2013, 17:23

GRACIAS

[Responder](#)



Escribe tu comentario

[Entrada más reciente](#)

[Inicio](#)

[Entrada antigua](#)

Suscribirse a: [Enviar comentarios \(Atom\)](#)

LICENCIA



Programación Java by [Enrique García Hernández](#)

Esta obra está bajo una licencia [Creative Commons Reconocimiento-NoComercial-CompartirIgual 3.0 España License](#).

Para reconocer la autoría debes poner el enlace <http://puntocomnoesunlenguaje.blogspot.com.es>

Con la tecnología de [Blogger](#).

[Configuración de la privacidad y las cookies](#)

Gestionado por Google Cumple el TCF de IAB. ID de CMP: 300