

PROMETEO

PROGRAMACIÓN

Uso de Objetos

Índice

- 01** Características
- 02** Constructores
- 03** Instancias
- 04** Métodos
- 05** Atributos o Propiedades
- 06** Métodos Estáticos

01

Características

Los objetos son la unidad básica de la programación orientada a objetos (POO). Representan entidades del mundo real, y cada objeto tiene un estado único, un comportamiento definido y una identidad propia.

❖ Estado

El estado de un objeto se define por sus **atributos** o **propiedades**. Los atributos son variables que almacenan información sobre el objeto.

Un objeto **Coche** puede tener diferentes estados, dependiendo de los valores de **color** y **velocidad**.

01

Características

❖ Comportamiento

El comportamiento de un objeto se define mediante sus **métodos**. Los métodos son funciones asociadas al objeto que pueden manipular su estado o realizar acciones.

❖ Identidad

Cada objeto tiene una **identidad única** que lo diferencia de otros objetos, incluso si tienen el mismo estado y comportamiento. Esto se debe a que los objetos se almacenan en ubicaciones distintas de memoria.

02

Constructores

Un **constructor** es un método especial que se utiliza para inicializar objetos de una clase. Se ejecuta automáticamente al crear una instancia de la clase y permite establecer valores iniciales para sus atributos.

Características de los Constructores

1. **Mismo nombre que la clase:**
 - Un constructor siempre tiene el mismo nombre que la clase.
 - No tiene tipo de retorno, ni siquiera **void**.
2. **Automático al instanciar un objeto:**
 - Se ejecuta automáticamente cuando se crea un objeto con la palabra clave **new**.
3. **Sobrecarga:**
 - Puedes definir múltiples constructores con diferentes parámetros (sobrecarga de constructores).
4. **Por defecto:**
 - Si no defines un constructor, Java proporciona uno por defecto sin parámetros.

02

Constructores

Constructores con Parámetros

Puedes definir constructores con parámetros para inicializar los atributos con valores específicos al crear un objeto.

Sobrecarga de Constructores

Puedes definir múltiples constructores en la misma clase, siempre y cuando tengan diferentes listas de parámetros.

Uso de **this** en Constructores

La palabra clave **this** se utiliza para referirse a los atributos de la clase actual, especialmente cuando los nombres de los parámetros son iguales a los nombres de los atributos.

03

Instancias

La instanciación de objetos es el proceso de crear un objeto en Java a partir de una clase. Este proceso consta de dos pasos principales: declaración y creación.

★ Declaración de un Objeto

La **declaración** consiste en definir una variable que contendrá la referencia al objeto. La sintaxis es similar a la declaración de una variable normal, pero se especifica el tipo como el nombre de la clase.

★ Creación de un Objeto

La **creación** consiste en asignar un nuevo objeto a la referencia declarada. Esto se realiza con la palabra clave **new**, que reserva espacio en memoria y ejecuta el constructor de la clase.

03

Instancias

★ Proceso Interno de Instanciación

Cuando se crea un objeto:

1. **La JVM reserva memoria** para almacenar el objeto.
2. **Se ejecuta el constructor** de la clase.
3. **Se inicializan los atributos** con valores predeterminados o definidos en el constructor.
4. La referencia al objeto se almacena en la variable declarada.

★ Asignación de Referencias

Una variable puede almacenar la referencia a un objeto. Si dos variables apuntan al mismo objeto, los cambios realizados a través de una variable afectarán al objeto referenciado por la otra.

★ Instanciar Múltiples Objetos

Cada objeto creado con `new` es independiente, aunque sea de la misma clase.

04

Métodos

Los métodos son bloques de código que realizan una tarea específica y pueden ser reutilizados en diferentes partes de un programa. Entender cómo funcionan los parámetros y los valores de retorno permite modularizar y estructurar mejor el código.

¿Qué es un Método?

Un método es una función dentro de una clase que puede realizar acciones sobre los datos del objeto o devolver información.

- **Modificador:** Define el nivel de acceso (`public`, `private`, etc.).
- **Tipo de Retorno:** El tipo de dato que devuelve el método (o `void` si no devuelve nada).
- **Nombre del Método:** Identificador único del método.
- **Parámetros:** Lista opcional de variables que recibe el método como entrada.

04

Métodos

1. Métodos sin Parámetros ni Retorno

Estos métodos ejecutan una acción sin requerir datos de entrada ni devolver un resultado.

2. Métodos con Parámetros

Los métodos con parámetros reciben datos de entrada que afectan su comportamiento.

3. Métodos con Valores de Retorno

Un método puede devolver un resultado utilizando la palabra clave `return`.

4. Métodos con Parámetros y Retorno

Los métodos pueden recibir datos de entrada y devolver un resultado.

5. Sobrecarga de Métodos

La sobrecarga permite definir varios métodos con el mismo nombre pero con diferentes parámetros.

Atributos o Propiedades

Las propiedades de un objeto, también conocidas como atributos, son variables asociadas a una clase que representan las características del objeto. Estas propiedades se utilizan para definir el estado del objeto y se pueden acceder y modificar a través de métodos, como getters y setters.

Declaración de Propiedades

En Java, las propiedades de una clase se declaran como variables dentro de la clase. Por convención, estas suelen ser privadas para proteger su acceso directo desde fuera de la clase.

Acceso a las Propiedades

El acceso a las propiedades se realiza generalmente mediante métodos especiales llamados **getters** (para obtener el valor) y **setters** (para modificar el valor).

Getters y Setters

1. **Getter:** Devuelve el valor de una propiedad.
2. **Setter:** Asigna un nuevo valor a la propiedad.

Atributos o Propiedades

Encapsulación de Propiedades

La **encapsulación** es una práctica fundamental en la programación orientada a objetos que protege las propiedades de acceso no autorizado o modificación indebida.

Beneficios de la Encapsulación

1. **Protección:** Evita modificaciones directas a las propiedades.
2. **Flexibilidad:** Permite validar valores antes de asignarlos.
3. **Control:** Facilita el mantenimiento del código al centralizar el acceso.

06

Métodos estáticos

Los métodos estáticos en Java son aquellos que pertenecen a la clase en lugar de a las instancias de la clase. Esto significa que se pueden llamar sin crear un objeto de la clase.

¿Qué es un Método Estático?

Un método se declara como **estático** utilizando la palabra clave `static`. Estos métodos:

1. No dependen de una instancia de la clase.
2. Solo pueden acceder a otros miembros estáticos de la clase.
3. Son ideales para operaciones generales que no necesitan modificar el estado del objeto.



PROMETEO