

Que es un paradigma de Programación?

Un paradigma de programación es una manera o estilo de programación de software.

Existen diferentes formas de diseñar un lenguaje de programación y varios modos de trabajar para obtener los resultados que necesitan los programadores.

En síntesis un paradigma son un conjunto de métodos sistemáticos aplicables en todos los niveles del diseño de programas para resolver problemas computacionales.



Programación Orientada a Objetos (POO)

Es una forma especial de programar, más cercana a como expresaríamos las cosas en la vida real que otros tipos de programación.

Se basa en la idea natural de un mundo lleno de objetos y que la resolución de problemas se realiza mediante el modelo de objetos.

Dicho de una forma mas simple, es una forma de descomponer un problema para solucionarlo de una modo distinto.

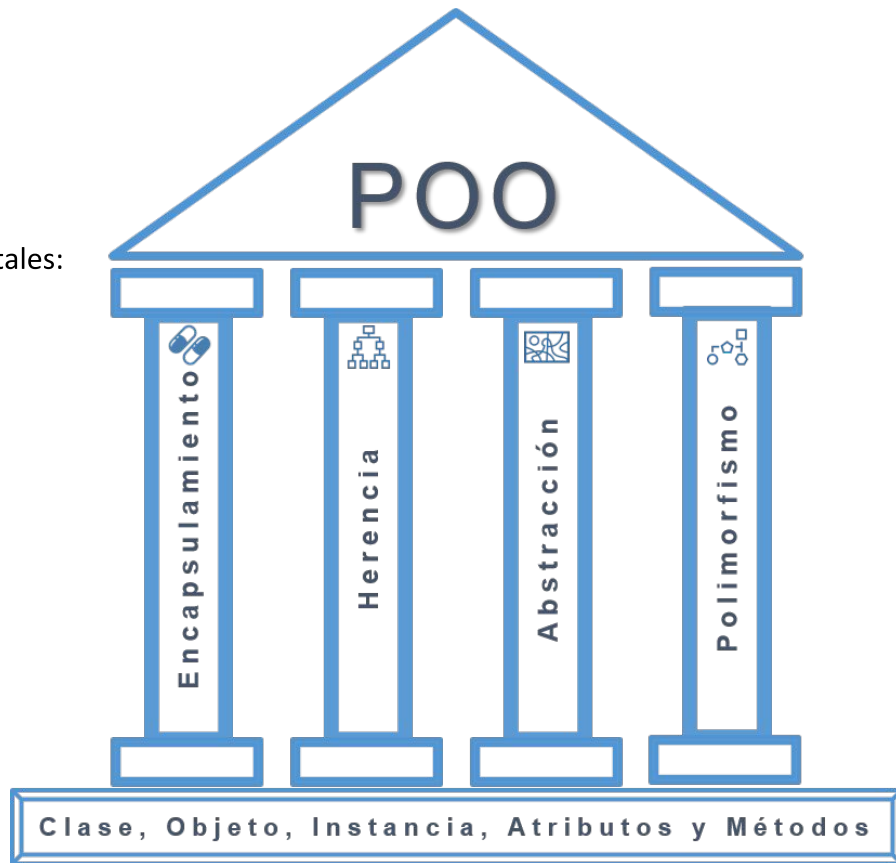


Pilares de la P00

La programación Orientada a Objetos se basa en 4 pilares fundamentales:
El encapsulamiento, la Herencia, la Abstracción y el Polimorfismo.

Pero como todo pilar debe tener unas bases que debemos aprender primeramente:

- ✓ Clase
- ✓ Objeto
- ✓ Instancia
- ✓ Atributos
- ✓ Métodos



Clase

Lo primero que debemos entender es que los tipos de datos que nos proporcionan los lenguajes de programación no son suficientes para representar algo en la vida real, como por ejemplo: un automóvil.

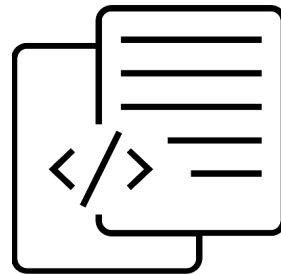
Un automóvil necesita mucho mas que una cadena de caracteres, un numero entero e incluso hasta un arreglo para que podamos diferenciarlo de otros automóviles.

Las clases nos ayudan a definir este nuevo “tipo de dato” con una estructura que le daremos para solucionar un problema planteado.

También lo podemos ver como una plantilla, un molde, en la que existen dos categorías de miembros a tener en cuenta:

Los Atributos (Datos) y los Métodos (Algoritmos)

En resumen una clase es un conjunto de objetos con características similares.



Atributos

Cuando debemos crear una clase lo primero que tenemos que hacer es pensar en que hace único a cada objeto, esos datos que diferencian una cosa de otra.

Si vemos un automóvil, ¿qué lo hace único entre tantos? Podemos pensar en su color, marca, patente etc.

Acá podremos utilizar los datos que ya conocemos (numéricos, caracteres y booleanos) y también podemos utilizar otros objetos.

Entonces, podemos resumir que los atributos son características individuales que diferencian un objeto de otro y determinan su apariencia, estado u otras cualidades.

Suelen ser llamadas como: Variable de Instancia, Miembro de Instancia o Campo de Instancia.



Métodos

Después debemos analizar que pueden hacer estos nuevos “tipos de datos”, por ejemplo: un automóvil puede encenderse, apagarse y además te puede mostrar su información.

A esto llamamos métodos, a un conjunto de instrucciones definidas dentro de una clase, que realizan una determinada tarea y a las que podemos invocar mediante un nombre.

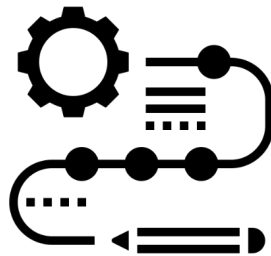
Cuando se llama a un método, la ejecución del programa pasa al método, ejecuta las instrucciones que se encuentren en él y la ejecución continúa a partir del punto donde se produjo el llamado, es decir, sigue su flujo natural.

La idea principal de los métodos es no escribir una misma instrucción varias veces, si la vamos a utilizar mas de una vez.

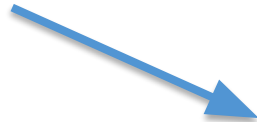
Mensajes: Los objetos interactúan enviándose mensajes unos a otros.

Tras la recepción de un mensaje el objeto actuará. La acción puede ser el envío de otros mensajes, el cambio de su estado, o la ejecución de cualquier otra tarea que se requiera que haga el objeto.

En resumen a un objeto se le envía un mensaje cuando una instancia utiliza un método 🗣️



Ejemplo



Atributos (Características)

- color
- marca
- patente

Métodos (Acciones)

- encender()
- apagar()
- mostrarAtributos()

Ejemplo

Clase

Métodos

Paquete

Identificador

Atributos

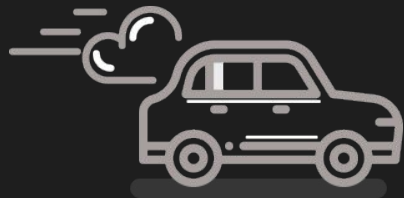
```
package com.curso.java.principal;

/**
 * @author EducacionIT
 * @class Auto
 * @date 10 jun. 2020
 */
class Auto {
    String color;
    String marca;
    String patente;
    boolean encendido;

    void encender() {
        encendido = true;
    }

    void apagar() {
        encendido = false;
    }

    void mostrarDatos() {
        String mensaje = "El Auto es de color " + color + ", marca " + marca + ", patente " + patente
            + " y se encuentra " + ((encendido) ? "encendido" : "apagado");
        System.out.println(mensaje);
    }
}
```



Instancia y Objeto

Recordemos un poco; cuando creamos una variable o constante debemos definir: el tipo de dato, su identificador y su valor inicial, con eso logramos poder acceder a ella mas adelante y poder cambiar su valor.

Lo mismo ocurre con un objeto, lo primero que debemos hacer es declararlo: tipo de dato, su identificador y asignarle un espacio de memoria.

Y que significa asignarle un espacio de memoria? Significa decirle al software que a partir de ese momento debe guardar bajo un mismo nombre una serie de atributos y comportamientos.

Pero como ahora el dato es mas complejo que una simple cadena de caracteres debemos también hacerla de forma distinta:

- ✓ Tipo de Dato: clase de donde tomara el molde o plantilla
- ✓ Identificador
- ✓ La sentencia "new"
- ✓ Volver a decir el tipo de dato, como en los arreglos
- ✓ Y cerrar con paréntesis "()"

```
Auto auto1 = new Auto();
```

Instancia y Objeto

Entonces ahora podemos decir que:

- ✓ Una Instancia es una ocurrencia de la clase, lo logramos cuando le decimos a Java que cree un **new Auto()**.
- ✓ Al momento de crear un objeto se produce la instanciación.
- ✓ Un Objeto es una instancia de una Clase específica.



Instancia y Objeto

Identificador

Declaración

```
// declaramos el objeto auto1 y lo instanciamos de la Clase Auto  
Auto auto1 = new Auto();
```

Instancia

Cambiamos los atributos
o asignamos valores

```
//le damos valores a los atributos del auto1  
auto1.color = "Rojo";  
auto1.marca = "Ferrari";  
auto1.patente = "ABC-188";
```

```
//les damos comportamiento  
auto1.encender();  
auto2.apagar();  
  
//comportamiento de mostrar datos  
auto1.mostrarDatos();
```

Usamos los métodos



Como se destruye un Objeto en Java

En Java quien se encarga de destruir o liberar la memoria de los objetos que ya no usamos es El Garbage Collector (GC) y lo hace de forma automática administrando la memoria.

