



POO (Programación Orientada a Objetos)

Paradigma de Programación orientado Objetos.

Class of Poo by Raúl Gómez & Itai Bejar

Esquema a desarrollar



1 – ¿Qué es Poo?



2- ¿Por dónde empiezo?



3- ¿Qué es un objeto?, ¿Cómo lo creo?



4- ¿Cuáles son los Atributos de una Clase?



5- ¿Qué son los constructores , Getter y Setter ?

- Es un paradigma de programación que busca reestructurar un programa en piezas simples y reutilizables de código.
- Se enfoca en las Clases ,los objetos, los atributos y sus interacciones para diseñar un programa.
- Representa Objetos de una forma más cercana a cómo los expresaríamos en la vida real.
- Los programas se definen en términos de objetos, propiedades, métodos, y la interacción (comunicación) entre objetos.

¿Qué es Poo?

¿Por dónde empiezo?

- Primero Dividir el problema en subproblemas mas simples. Pero para eso ahora utilizaremos 3 clases.
- Crear las clases necesarias para resolver el problema.
 - 1- La Clase Objeto (Atributos).
 - 2- La Clase ObjetoServicio (Lógica de Objetos).
 - 3- La Clase Main (Lógica Principal).
- 1- Crear los Constructores en la clase Objeto.
- 1- Crear los métodos para utilizar los atributos (Getter y Setter) en la clase Objeto.
- 2- Crear los Métodos y funciones que se encargan de la lógica de los Objetos en la clase ObjetosServicio.
- 3- Desde el Main controlar el programa en si.

Objeto:

Un objeto en la programación es la forma que tenemos de llevar a código, la representación más fidedigna posible de un objeto de la vida real. Un objeto tendrá características (atributos). Los Atributos entre los objetos pueden ser diferentes.

Ejemplo:

Una persona seria un objeto cuyos atributos son: brazos, piernas, ojos, pelo. Pero de una persona a otra pueden variar.

Creación:

El Objeto se crea a partir de una Clase cuando definimos los atributos que componen a ese objeto y le dan su identidad.

Recuerda: El nombre de la clase debe comenzar en mayúscula.

Y los atributos siempre en minúsculas.

¿Qué es un
Objeto? ¿Cómo
lo creo?

Clases y Atributos

Clase Jugador		
Atributos	nombre	
	edad	
	colorRemera	
	colorPantalon	
	colorBotines	
	colorPelo	
	colorPiel	

Objeto jugador1		
Atributos	nombre	Raul
	edad	44
	colorRemera	Roja
	colorPantalon	Rojo
	colorBotines	Negro
	ColorPelo	Castaño
	colorPiel	Mestizo

Objeto jugador2		
Atributos	nombre	Itai
	edad	42
	colorRemera	Azul
	colorPantalon	Blanco
	colorBotines	Azul
	ColorPelo	Negro
	colorPiel	Clara



Cuando creamos una persona (Objeto/Instancia de una clase), con sus diferentes características (ATRIBUTOS), que podrían ser:

- nombre
- edad
- color remera
- etc.

vamos a requerir un molde o plantilla (**CLASE**).

La persona (OBJETO) puede tener diferentes acciones (METODOS), que podrían ser:

- correr(),
- patear(),
- saltar()
- cabecear(),
- entre otros.

Constructores

- Constructor:

Son métodos propios del objeto que nos permiten Crearlo. (hacerlo nacer, instanciarlo).

- Se puede utilizar en NetBeans: alt+insert o botón derecho insertar código y constructor (vacío y/o con parámetros).
- Por defecto Java provee un constructor vacío.
- Una vez que declaramos una clase, podemos crear o instanciar objetos de esa clase.
- **NombreClase nombreObjeto = new NombreClase();**

The diagram illustrates the creation of objects in Java. It shows two ways to declare and instantiate a `Jugador` object:

- Uppercase Declaration:** A blue arrow points down to `Jugador`, which then points down to `Jugador`. This represents the declaration and instantiation of an object using the class name.
- Lowercase Declaration:** A blue arrow points down to `jugador`, which then points down to `jugador`. This represents the declaration and instantiation of an object using a lowercase variable name.
- Object Creation:** The text `= new Jugador(); //constructor vacío` shows the creation of an object using the `new` keyword and the default constructor.
- Object Creation with Parameters:** The text `= new Jugador(nombre, edad, colorRemera, colorPantalon, colorBotines, colorPelo, colorPiel); //constructor por parámetros (sobrecargado).` shows the creation of an object using the `new` keyword and a parameterized constructor.

Getter Setter

- Los atributos se declaran **private** (por encapsulamiento).
- para acceder a los atributos de un objeto desde otra Clase se utilizan:
 - Set: modifica el atributo.
`Jugador1.setnombre("Raul"); //ingresamos el nombre`
`Jugador2.setnombre("Itai");`
 - Get: trae el contenido del atributo
`System.out.print("el nombre es: " + Jugador1.getnombre());`
`//mostramos el nombre`

Otro ejemplo de Clase y Objetos

