



Programación Orientada a Objetos en Java: Clases y objetos

Elaboró:
José Antonio Vergara Camacho
Patricia Martínez Moreno



Elaborado: Enero 26, 2022

```
... object to mirror_
mirror_mod.mirror_object
operation == "MIRROR_X":
mirror_mod.use_x = True
mirror_mod.use_y = False
mirror_mod.use_z = False
operation == "MIRROR_Y":
mirror_mod.use_x = False
mirror_mod.use_y = True
mirror_mod.use_z = False
operation == "MIRROR_Z":
mirror_mod.use_x = False
mirror_mod.use_y = False
mirror_mod.use_z = True
```

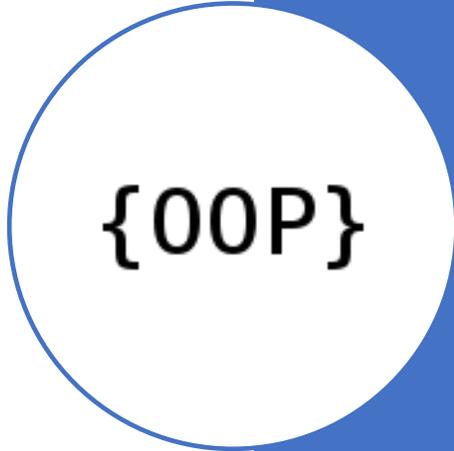
```
selection at the end -add
mirror_ob.select= 1
modifier_ob.select=1
context.scene.objects.active
("Selected" + str(modifier_ob
mirror_ob.select = 0
= bpy.context.selected_object
data.objects[one.name].select
print("please select exactly
-- OPERATOR CLASSES ----
```

```
types.Operator):
on X mirror to the selected
object.mirror_mirror_x"
mirror X"
```

```
context):
context.active_object is not
```

Intención educativa

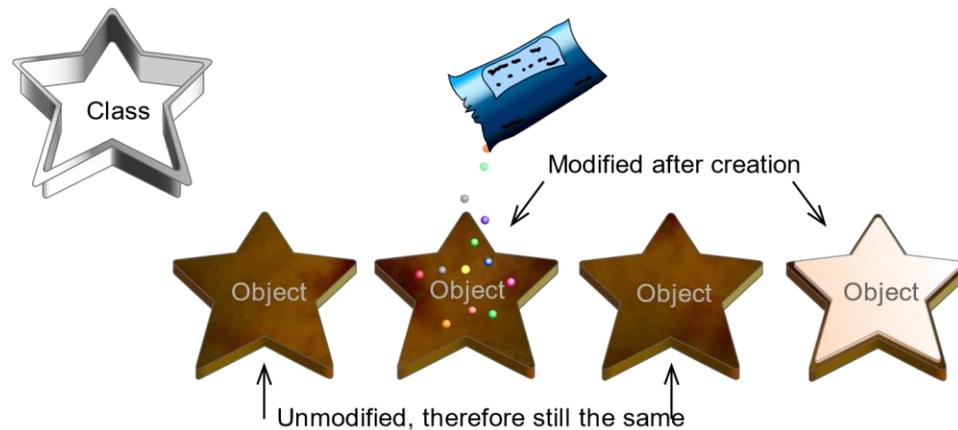
- El presente recurso educativo tiene el propósito de apoyar la comprensión de conceptos básicos de la programación orientada a objetos, con la finalidad de construir programas de cómputo que serán la base en la formación para el desarrollo de software.
- Se describen los conceptos de **clases**, **objetos**, **atributos** y **métodos** que representan los más importantes de la Programación Orientada por Objetos (POO) y están fuertemente relacionados. Los objetos se crean a partir de clases, y las clases sirven como plantillas para crear objetos. A los objetos también se les llama instancias de clase.



{OOP}

Clases y objetos

- En Orientación a Objetos hay dos conceptos que son el fundamento para desarrollar software con un lenguaje de programación como Java: la Clase y el Objeto. Una clase es como un molde o plantilla a partir de la cual se crean Objetos.



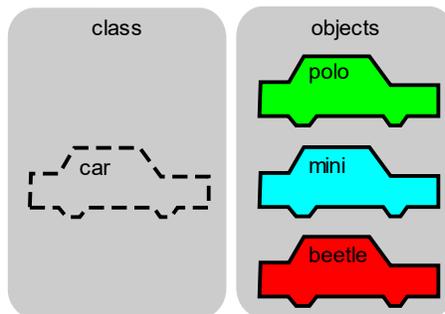
[Class-Object-Modification](#) por [Wikimedia](#) bajo licencia [CC BY-SA 4.0](#)

Con el fin de ejemplificar el concepto, imaginemos un molde de galletas. A partir de éste podremos generar todas las galletas que deseemos y como se ve en la ilustración podríamos obtener galletas de diferentes colores o con diferente peso...a estas características que nos permiten diferenciar un **Objeto** de otro se les conoce como **atributos**, o campos o variables de instancia. Los tres términos son equivalentes. A un **objeto** que se crea a partir de una clase también se le conoce como una **INSTANCIA** de la clase

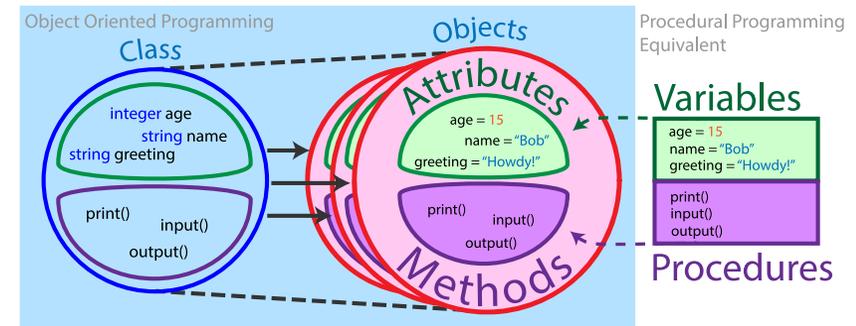
Clases y objetos

¿Qué es un objeto?

- El objeto es una entidad que se genera a partir de una clase y que “vive” en la memoria RAM del sistema.
 - Ejemplo; primero podemos ver la clase como un prototipo a partir del cual se construyen los objetos y que determina cuales serán las **características** del **objeto** y su **comportamiento**.
 - También podemos ver la clase como un Plano Maestro para crear objetos de un mismo tipo; por ejemplo, al construir varios automóviles del mismo modelo existe un plano o diagrama que contiene todas las especificaciones que deben llevar dichos automóviles para poderlos fabricar o, según la terminología de Java, poderlos instanciar.
-
- Ejemplos de clases y objetos:



OOP-objects and classes por [Wikimedia](#) bajo licencia [CC0 1.0](#)



Object-Var-Proc por [Wikimedia](#) bajo licencia [CC BY-SA 4.0](#)

Atributos

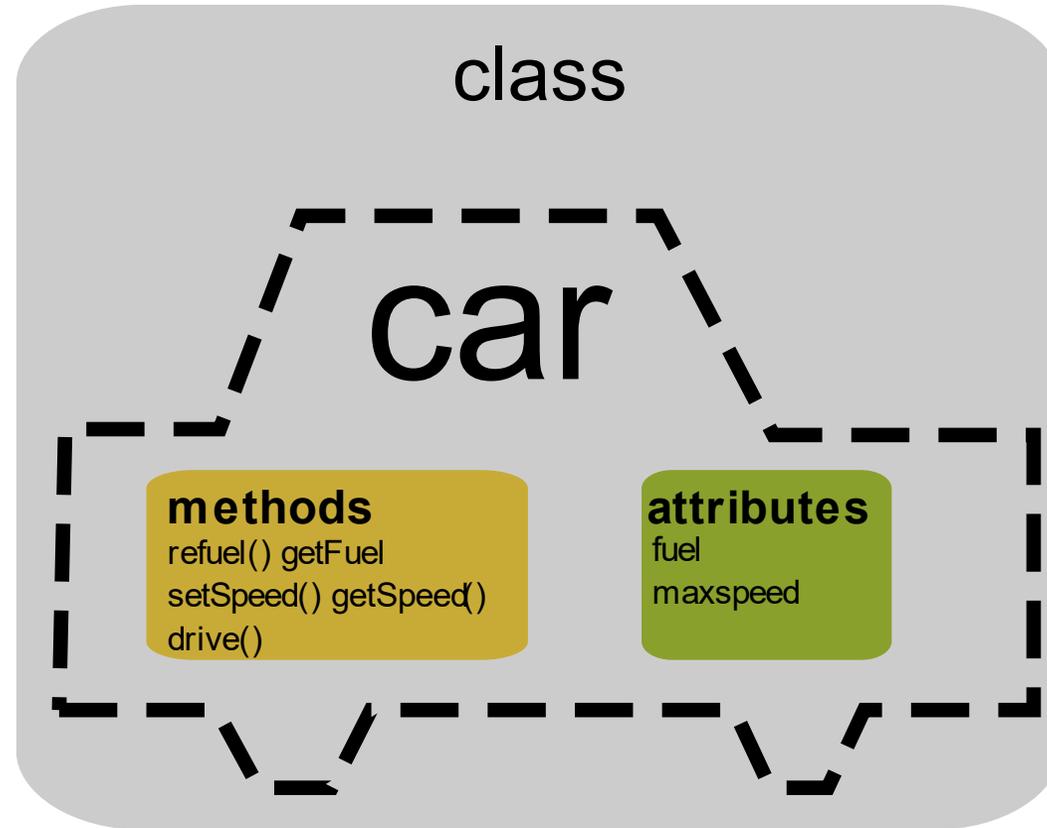
- Cada *Objeto* que se genera a partir de una *clase* tendrá los atributos definidos en dicha clase, pero con sus propios valores. El conjunto de los valores de sus atributos define el ESTADO del Objeto.
- Continuando con nuestro ejemplo, la *clase* Galleta tiene entonces los atributos color y peso. Los atributos son las variables que durante la ejecución del programa sufrirán cambios en su valor, así que, como toda variable, éstas deberán tener un tipo de dato asociado. En este caso el *atributo color* tiene un tipo de dato conocido como *String* o cadena de caracteres y el tipo de dato de la variable peso es un tipo primitivo llamado *double* y se trata de un número de punto decimal.
- Finalmente asignemos al atributo color el valor “naranja” y al atributo peso el valor 99.5 gramos. Toda esta descripción se resume en las siguientes declaraciones en lenguaje Java:

```
String color=" naranja";
```

```
Double peso= 99.5;
```

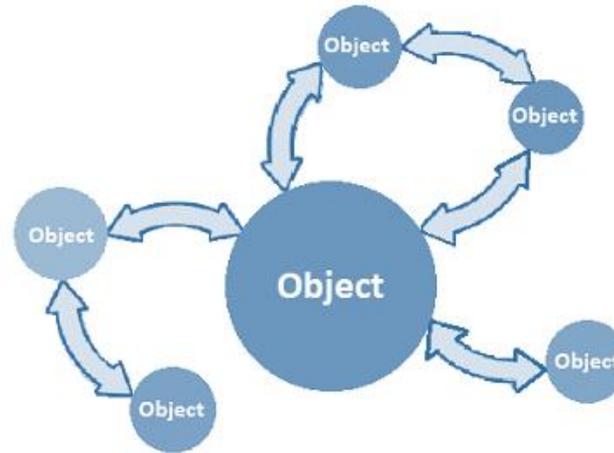
Atributos

- Ejemplo



Métodos

- Aparte de los *atributos*, en las *clases* también se define el comportamiento que tendrán los objetos generados a partir de la clase. Este comportamiento tiene como objetivo modificar el valor de los *atributos* del propio objeto en donde está definido dicho comportamiento o el valor de los atributos de otros objetos que tienen algún tipo de relación con éste objeto.
- La forma de definir el comportamiento es a través de los métodos de la clase.



- Por ejemplo, podríamos tener un comportamiento que asigne un valor al *atributo color* de la galleta y al método por medio del cual lograríamos esto, podríamos llamarle *asignarColor()*. Los paréntesis son un elemento sintáctico obligatorio al declarar un método en Java. También podemos tener un método que cambie el valor del peso del objeto galleta, podríamos llamarle *modificarPeso()*. Además, deseamos que, si este método cambia con éxito el valor, nos lo informe regresando un valor de tipo lógico (booleano), o sea que los métodos tienen la capacidad de regresar o no información a quien los invoca.

Declaración de atributos

- Ya hemos descrito lo que son los atributos de una clase, ahora vamos a especificar la forma como se declaran los atributos en Java. Importante: los paréntesis angulares [] indican que lo encerrado entre ellos puede o no aparecer en la declaración, ¡es opcional!

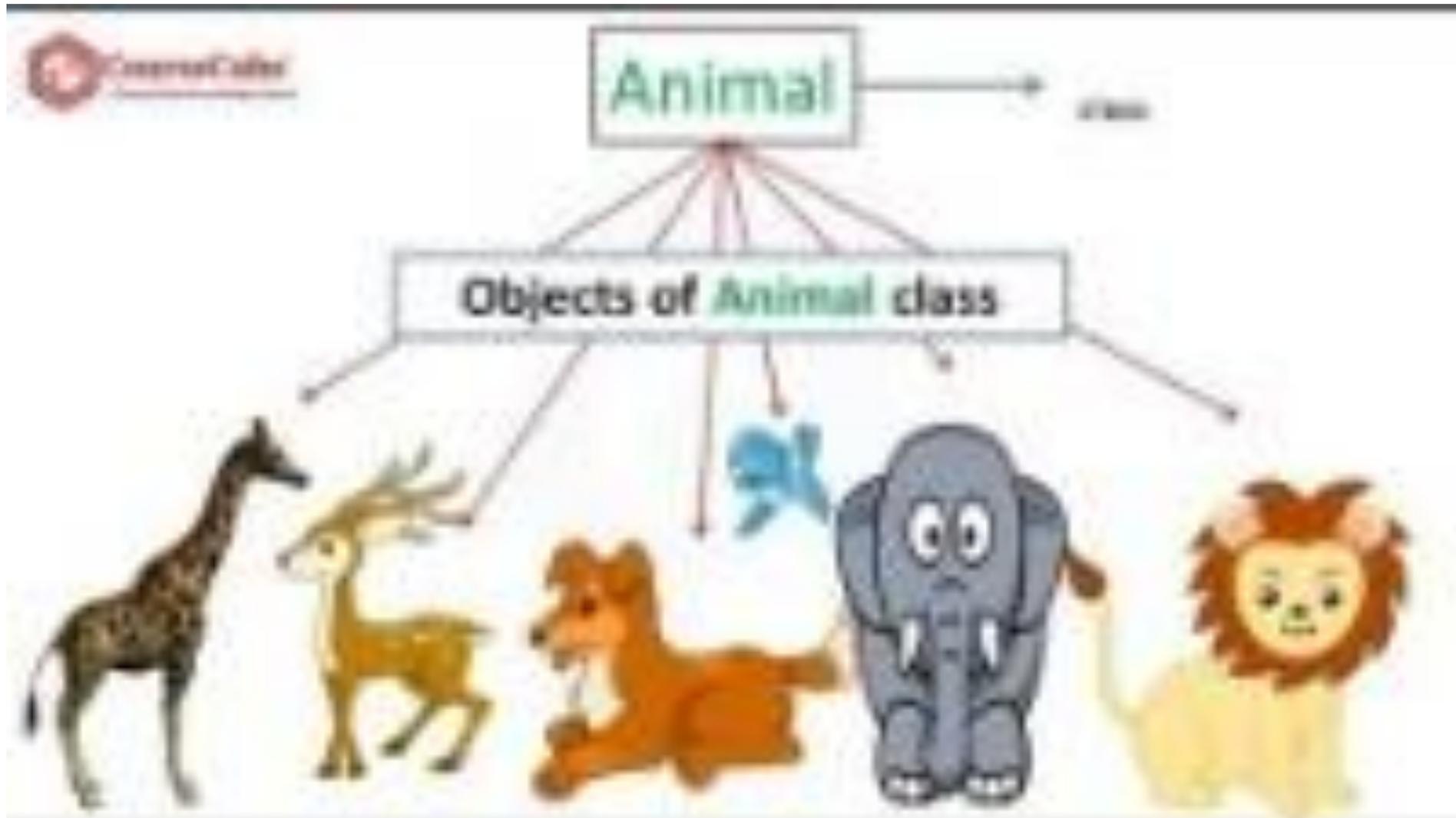
[modificadorAcceso] tipoDato nombreAtributo[=valor];

- Aquí tenemos 5 ejemplos de la declaración de atributos pertenecientes a un objeto Galleta.

```
public int idTipoGalleta=5;  
public String saborGalleta="naranja";  
public char codigoColor='N';  
public double precio=2.50;  
public boolean esIntegral=true;
```

Los nombres de los atributos únicamente pueden comenzar con letras, guion bajo (_) o signo de dólar (\$).

Ver video: [Class and object with example in real world](#)



Ejercicios

Identifica 4 características relevantes de los siguientes objetos dependiendo de su uso y de su comportamiento:

- Persona
- Vehículo
- Hospital
- Competencia deportiva



Problemas

1. Una empresa de venta de autos quiere llevar un control de los autos que tiene a disposición.

¿Qué características requiere conocer de los autos?

2. La Dirección de Tránsito Vehicular lleva una base de datos con la información más importantes de las personas para efectos de conceder licencias de manejo.

¿Qué atributos serían necesarios para resolver el problema?



Bibliografía consultada

1. Schachs, S. (2005). Análisis y Diseño Orientado a Objetos con UML y el Proceso Unificado. México: McGraw-Hill.
2. Horstmann, C. y Cornell, G. (2012). Core Java Volumen I Fundamentals 9th Edition. Prentice Hall - Sun Microsystem press



Icons8, MIT, via [Wikimedia Commons](#)