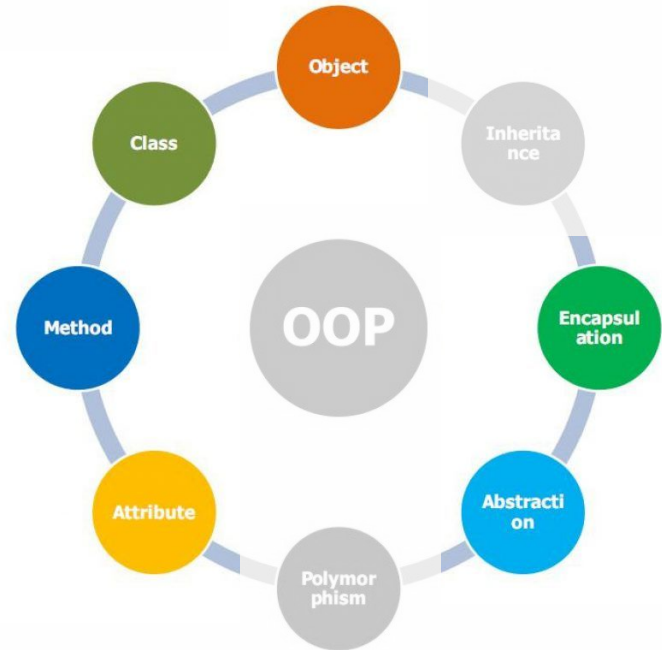


Métodos de Clases

1

Seguimos recorriendo y completando este gráfico.
Sumemos ahora los métodos de una clase.



Métodos de Clases

2

Hasta ahora, nuestras clases tienen **Nombre**, y **Atributos**.
Nombre para nombrarla... Atributos para modelar sus características.

Y también habíamos mencionado que las clases tienen **responsabilidades**.

- » Estas **responsabilidades** son lo **métodos** de las Clases.

Métodos de Clases

3

¿Qué son los métodos (responsabilidades) de una clase?

Los métodos de las clases serán el conjunto de sus responsabilidades.

Y así como el conjunto de atributos de un objeto determinará su **estado**, el **conjunto de métodos** determinará su **comportamiento**.

Por ejemplo la clase **CuentaBancaria**, tendrá como responsabilidades, **informarTitular()**, **informarSaldo()**, **extraer()**, **depositar()**...

Métodos de Clases

4

En el ejemplo de **CuentaBancaria**: seguramente tendrá **Saldo** como atributo, y más que seguramente, este atributo será privado.

No se podrá acceder al valor de **Saldo** desde fuera de la **CuentaBancaria**.

Entonces: **¿Cómo se puede acceder o modificar ese Saldo?**

Se podrá hacerlo a través de sus métodos:

- » **depositar(550)** → incrementa su saldo en \$ 550.-
- » **extraer(200)** → decrementa su saldo en \$ 200.- (si tiene disponible)

Métodos de Clases

5

Encapsulamiento

Recordamos que dijimos que por diseño, comenzamos modelando todos los **atributos** como **privados**.

A esta propiedad del **ocultamiento del estado**, se la conoce como **encapsulamiento**.

Entonces, si todos los atributos son privados, invisibles para el exterior...

¿cómo se puede acceder al valor de un atributo?...

Métodos de Clases

6

Para acceder al valor de un atributo con visibilidad privada, lo tenemos que hacer a través de algún método del propio objeto.

Es decir, el **estado interno** del objeto es **privado**, está **encapsulado**, y solo podés acceder a los valores de sus atributos a través de sus métodos.

Métodos de Clases

7

Siguiendo con el ejemplo de **CuentaBancaria**: además de **saldo**, la cuenta tendría también un **titular**, y un **numeroDeCuenta**.

Por ejemplo, si quiero obtener el saldo de la cuenta, lo haría a través de la función `obtenerSaldo()`, que me devolvería el valor del atributo, saldo.

Así podría tener también, `conocerTitular()`, `leerNumeroDeCuenta()`...

Métodos de Clases

8

Siguiendo con este ejemplo de **CuentaBancaria**, con atributos **saldo**, **titular**, y **numeroDeCuenta**.

Ya mencionamos que nos interesa poder operar con estos atributos. Y de hecho, es tan común *pedirle a las clases los valores de sus atributos*, para verlos o para modificarlos, que se hace a través de métodos que se llaman **getters y setters**.

Métodos de Clases

9

Getters y setters

Este es el nombre que se le da a los métodos que operan con los atributos de una clase. Son métodos comunes, normales.

Se escriben anteponiendo “set” o “get” al nombre del atributo. Es un estándar.

Así, en el ejemplo, en vez del método obtenerSaldo(), usaríamos **getSaldo()**.

Para obtener el titular, usaríamos **getTitular()**, y para obtener el número de cuenta, usaremos **getNumeroDeCuenta()**;

Para modificar el saldo directamente, usaríamos un método llamado **setSaldo()**; (con cuidado en este caso, conviene que el saldo sólo se modifique a través de los métodos que vimos antes: depositar(), extraer()).

UML - Métodos

10

Siguiendo con el **Diagrama de Clases** de UML, veamos ahora cómo modelar en UML los Métodos

Sumaremos al nombre de la Clase y a sus Atributos, sus **métodos**.

UML - Métodos

11

¿Cómo dibujamos una Clase con atributos y métodos en UML?
Continuamos con el rectángulo dividido en tres, y sumamos los métodos en el tercero, con su visibilidad, tipo de retorno y parámetros.

Persona
- nombre:String - edad: Integer - dni: String
+getNombre() +cumplirAnios()

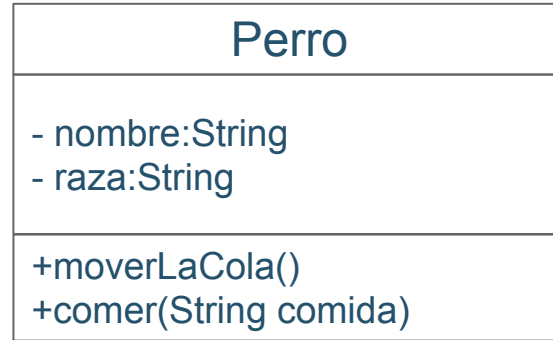
Perro
- nombre:String - raza:String
+moverLaCola() +comer(String comida)

AnimalDomestico
- nombre: String
+emitirSonido()

UML - Métodos

12

En detalle:



→ Nombre de la clase

→ Atributos de la clase

→ Métodos de la clase

+ comer (String comida)

Visibilidad: público

nombre

Tipo y nombre
del Parámetro