ATRIBUTOS Y MÉTODOS

INTRODUCCIÓN A LA ORIENTACIÓN A OBJETOS

OpenWebinars

ATRIBUTOS

- También conocidos como propiedades de una clase.
- Determinan la estructura de la misma.

```
3  public class Coche {
4
5      private String marca;
6      private String modelo;
7      private int anio;
8
```

ATRIBUTOS

- Suelen indicarse al **inicio** de la clase.
- Por ahora, siempre **private**.
- Hay que indicar el **tipo de dato** y el **nombre** del atributo.
  - ***No se puede usar var***
- Como tipo de dato podemos usar
  - Tipos primitivos
  - Otras clases (por ejemplo String).

ATRIBUTOS

- Los nombres siguen las mismas premisas que las variables.
  - Autodescriptivos
  - Concisos
  - Notación camel case

ENCAPSULAMIENTO

- Un objeto solamente conoce de otro su cara vista (interfaz).
- El resto queda oculta y solamente es conocido por el propio objeto.
- Sobre todo, **se ocultan los atributos**.
- La **cara vista** suelen ser los **métodos**.

MÉTODOS

- Nos permiten definir el **comportamiento** de una clase.
- Por ahora, siempre **public**.
- Pueden devolver un valor, o no devolver nada (*void*)
- Pueden recibir cero, uno o más **argumentos** (valores de entrada).
- Los nombres...
  - Autodescriptivos
  - *camelCase*
  - En ocasiones utilizan **verbos**.

MÉTODOS

- Ejemplo de método que
  - No devuelve nada (void)
  - No recibe argumentos

```
public void arrancar() {
    System.out.println("El coche %s %s %d ha arrancado"
                .formatted(marca, modelo, anio));
}
```

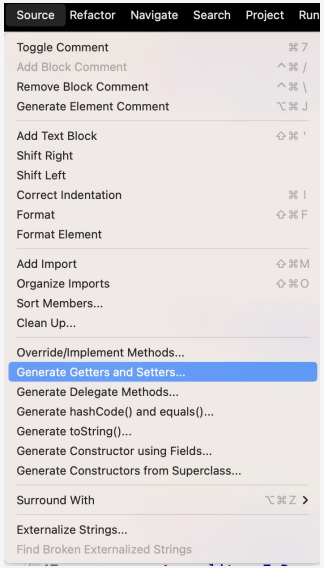
MÉTODOS

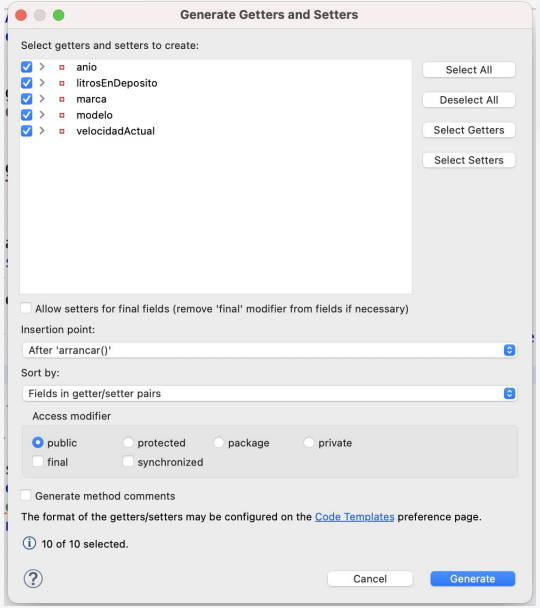
- Otro ejemplo de método que recibe argumentos y tiene valor de retorno.
- Para cada **argumento**, se indica al menos el **tipo de dato** y el **nombre**.
- Valor de retorno con la palabra **return**.

```java
/*
 * Incrementamos el depósito del coche con el
 * número indicado de litros.
 */
public double repostar(double cantidad) {
    this.litrosEnDeposito += cantidad;
    // this no es estrictamente obligatorio si no hay ambigüedad
    return litrosEnDeposito;
}
```

MÉTODOS GETTERS/SETTERS

- Forma conveniente de acceder a los atributos de una clase
  - *Getters*, para obtener el valor de un atributo.
  - *Setters*, para modificar el valor de un atributo.
- Su código se puede generar.





MÉTODOS QUE MODIFICAN ATRIBUTOS

- Cualquier método puede modificar el valor de un atributo.
- No solamente lo pueden hacer los constructores o setters.

```
public double viajar(double kilometros) {

    // Comprobamos que el coche puede hacer el viaje
    if (kilometros * 0.07 <= litrosEnDeposito) {
        double tiempoEnHoras = kilometros / velocidadActual;
        double tiempoEnSegundos = tiempoEnHoras * 60 * 60;
        this.litrosEnDeposito -= kilometros * 0.07;
        System.out.println("Tras el viaje, el depósito ha quedado con %.2f litros"
                .formatted(this.litrosEnDeposito));
        return tiempoEnSegundos;
    } else {
        System.out.println("Para poder hacer ese viaje debes repostar");
        return 0;
    }
}
```