

3.D. Métodos.

✓ Hecho

1. Utilización de métodos.

1.2. Constructores.

¿Recuerdas cuando hablábamos de la creación e instanciación de un objeto? Decíamos que utilizábamos el operador `new` seguido del nombre de la clase y una pareja de abrir-cerrar paréntesis. Además, el nombre de la clase era realmente el constructor de la misma, y lo definíamos como un método especial que sirve para inicializar valores. En este apartado vamos a ver un poco más sobre los constructores.

Un constructor es un método especial con el mismo nombre de la clase y que no devuelve ningún valor tras su ejecución.

Cuando creamos un objeto debemos instanciarlo utilizando el constructor de la clase. Veamos la clase `Date` proporcionada por la Biblioteca de Clases de Java. Si queremos instanciar un objeto a partir de la clase `Date` tan sólo tendremos que utilizar el constructor seguido de una pareja de abrir-cerrar paréntesis:

```
Date fecha = new Date();
```

Con la anterior instrucción estamos creando un objeto `fecha` de tipo `Date`, que contendrá la fecha y hora actual del sistema.

La estructura de los constructores es similar a la de cualquier método, salvo que no tiene tipo de dato devuelto porque no devuelve ningún valor. Está formada por una cabecera y un cuerpo, que contiene la inicialización de atributos y resto de instrucciones del constructor.

Estructura interna de un método constructor

```
public NombreClase (par1, par2, ..., parN)
{
    // Inicializacion de atributos
    // Resto de instrucciones del constructor
} // Fin del metodo
```

parámetros opcionales

puede llamar a otros métodos de la clase aunque no es recomendable

el modificador normalmente siempre es public

El método constructor tiene las siguientes particularidades:

- **El constructor es invocado automáticamente en la creación de un objeto,** y sólo esa vez.
- **Los constructores no empiezan con minúscula,** como el resto de los métodos, ya que se llaman igual que la clase y las clases empiezan con letra mayúscula.
- **Puede haber varios constructores** para una clase.
- Como cualquier método, el constructor **puede tener parámetros** para definir qué valores dar a los atributos del objeto.
- **El constructor por defecto** es aquél que no tiene argumentos o parámetros. Cuando creamos un objeto llamando al nombre de la clase sin argumentos, estamos utilizando el constructor por defecto.
- **Es necesario que toda clase tenga al menos un constructor.** Si no definimos constructores para una clase, y sólo en ese caso, el compilador crea un constructor por defecto vacío, que inicializa los atributos a sus **valores por defecto**, según del tipo que sean: 0 para los tipos numéricos, false para los boolean y null para los tipo carácter y las referencias. Dicho constructor lo que hace es llamar al constructor sin argumentos de la superclase (clase de la cual hereda); si la superclase no tiene constructor sin argumentos se produce un error de compilación.

Cuando definimos constructores personalizados, el constructor por defecto **deja de existir**, y si no definimos nosotros un constructor sin argumentos cuando intentemos utilizar el constructor por defecto nos dará un error de compilación.

◀ 3.C. Trabajando con objetos.

Ir a...

3.E. Librerías y paquetes. ▶