

Abraham Ramírez Moreno

1902627

Programación Orientada a Objetos

Grupo 32

Jorge Alberto Islas Pineda

Actividad 1

Tema: Conceptos básicos

## Sobrecarga de métodos

La sobrecarga de métodos ocurre cuando dentro de la misma clase 2 o más métodos del mismo nombre se declaran. Para que los métodos sobrecargados sean válidos deben de tener diferentes conjuntos de parámetros; es decir, el tipo de los parámetros, la cantidad de parámetros o el orden de los parámetros debe de diferir entre los métodos sobrecargados del mismo nombre. Notar que el tipo de dato que retorna el método no debe ser necesariamente igual.

Al llamar métodos sobrecargados, el programa escoge el método de acuerdo a los argumentos usados al llamarlo. El método llamado va a ser el que tenga los parámetros que concuerden con los argumentos dados en la llamada al método.

La sobrecarga es muy comúnmente usada para crear diferentes métodos que realizan funciones muy parecidas que pueden ser abstraídas dentro del mismo nombre.

## Sobreescritura de métodos

Una subclase puede redefinir los métodos de la clase de la que proviene. Esto se hace declarando en la subclase un método con el mismo nombre, conjunto de parámetros y tipo de retorno. Cuando un objeto de la subclase llama a ese método se va a llamar al método de la subclase y no al de la clase de la que se deriva (superclase).

En java se puede llamar al método sobrescrito colocando "super." Antes de llamarlo. También se puede colocar @Override antes de una declaración que sobrescribe un método para evitar errores; de esta manera le avisas al compilador que estas tratando de sobre escribir, y en caso de que no hallas sobrescrito te avisa mediante un error al compilar.

## Constructores

Los constructores son métodos especiales llamados cada vez que se crea un objeto de una clase. Estos permiten una inicialización personalizada de los objetos.

Para crear un constructor se debe declarar un método con el mismo nombre que la clase. Hay que notar que los constructores no tienen tipo de retorno.

Para llamar a un constructor se escribe la palabra new y luego se llama al método como cualquier otro.

Si no se declara un constructor, al crear un objeto se usará el constructor default. Este constructor no tiene parámetros e inicializa a las variables del objeto con sus

valores default. Si se declara un constructor en la clase no va a tener constructor default.

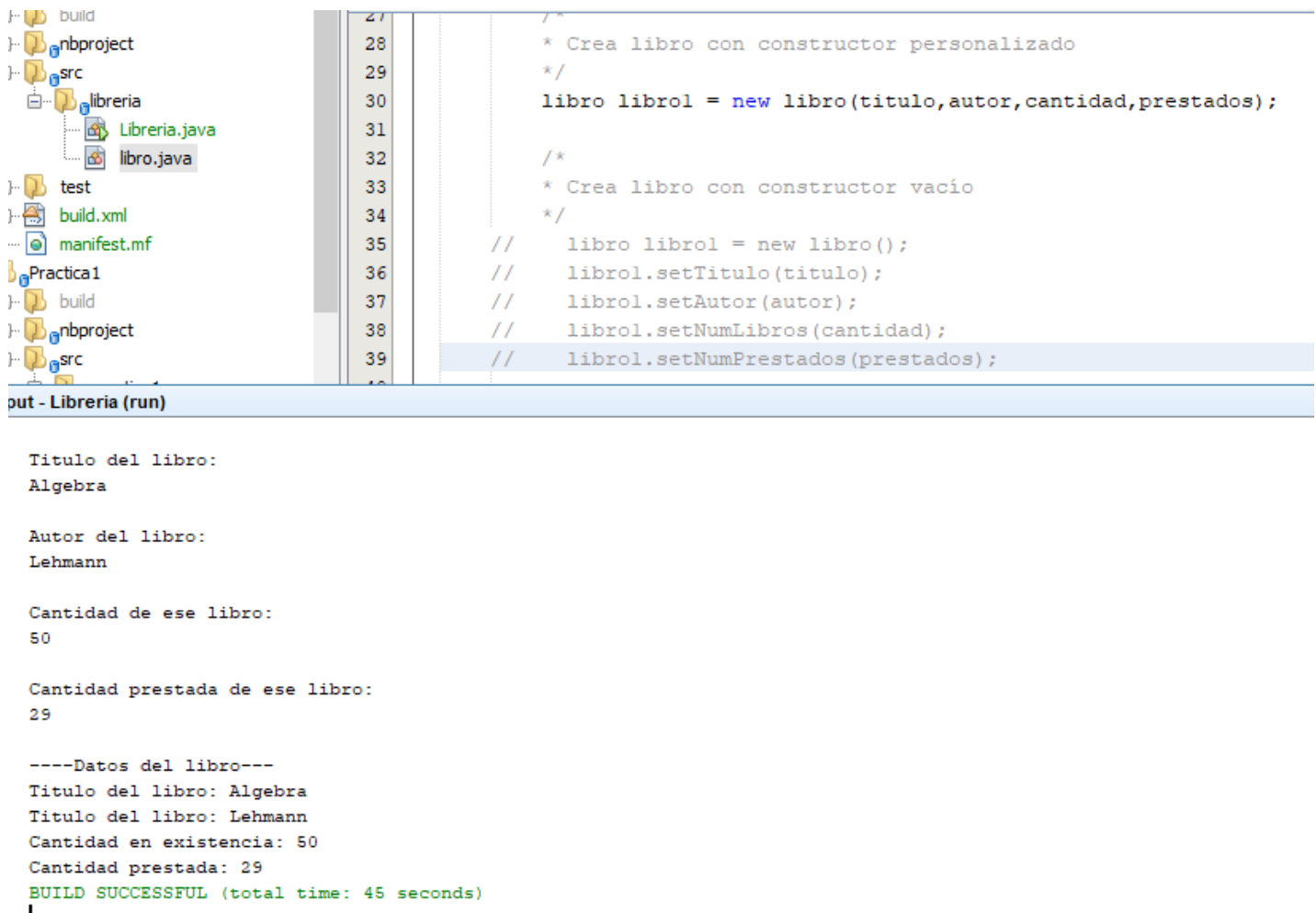
Los constructores pueden ser sobrecargados, permitiendo diferentes constructores para una misma clase dependiendo los parámetros.

Fuente:

[1] Paul Deitel, Harvey Deitel (2014) *Java How to Program (late Objects)* 10<sup>th</sup> Pearson.

## Screenshots programa librería

### 1- Usando constructor personalizado



The screenshot displays an IDE with a project structure on the left and a code editor on the right. The project structure includes folders for 'build', 'nbproject', 'src', and 'test'. The 'src' folder contains 'libreria' and 'libro.java'. The 'test' folder contains 'build.xml' and 'manifest.mf'. The 'libreria' folder contains 'Libreria.java' and 'libro.java'. The code editor shows the following Java code:

```
27  /*
28  * Crea libro con constructor personalizado
29  */
30  libro librol = new libro(titulo,autor,cantidad,prestados);
31
32  /*
33  * Crea libro con constructor vacío
34  */
35  // libro librol = new libro();
36  // librol.setTitulo(titulo);
37  // librol.setAutor(autor);
38  // librol.setNumLibros(cantidad);
39  // librol.setNumPrestados(prestados);
```

Below the code editor, the output of the program is shown:

```
out - Libreria (run)

Titulo del libro:
Algebra

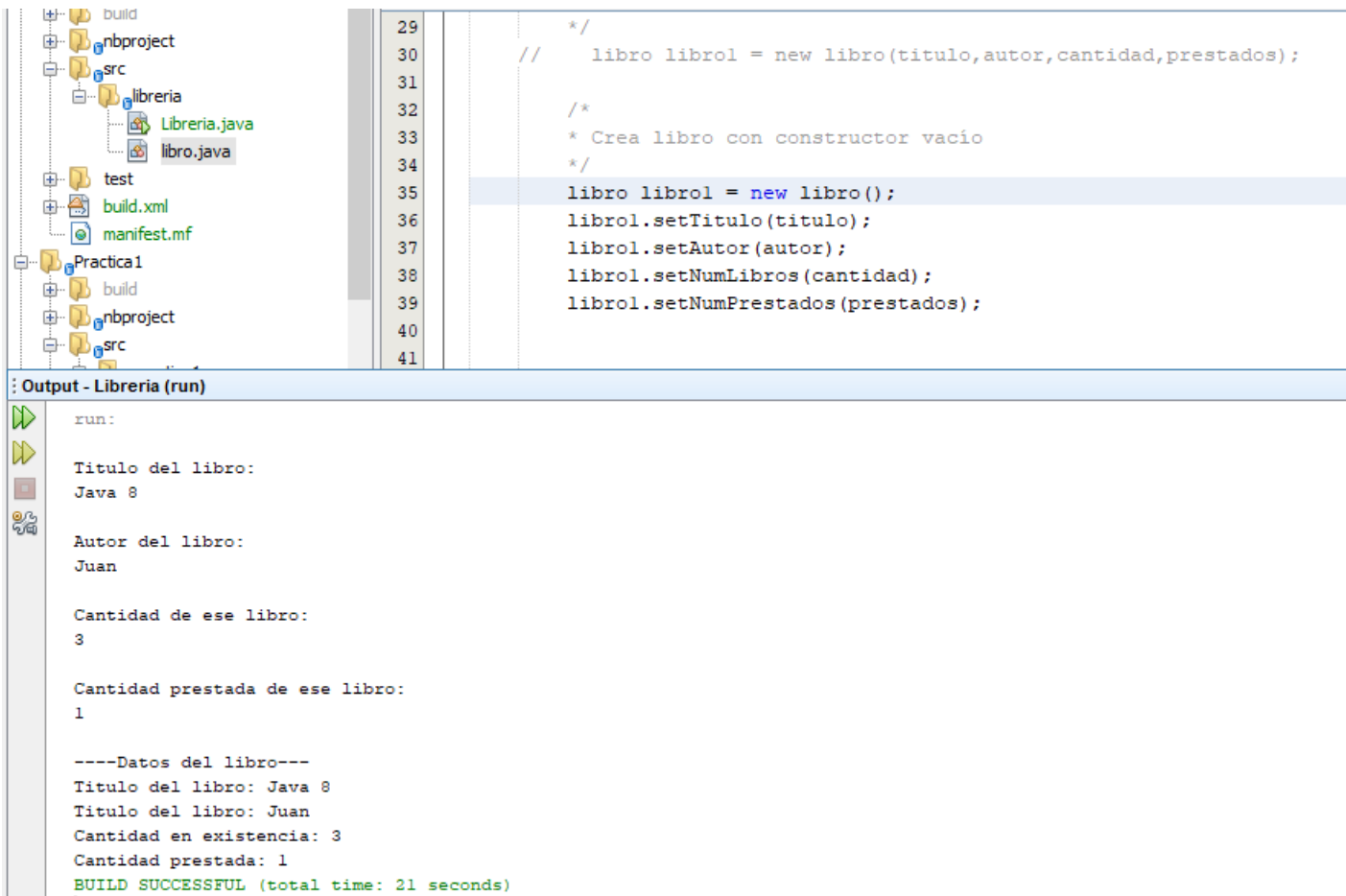
Autor del libro:
Lehmann

Cantidad de ese libro:
50

Cantidad prestada de ese libro:
29

----Datos del libro---
Titulo del libro: Algebra
Titulo del libro: Lehmann
Cantidad en existencia: 50
Cantidad prestada: 29
BUILD SUCCESSFUL (total time: 45 seconds)
```

## 2- Usando métodos set



The screenshot displays an IDE interface. On the left, a project tree shows a folder named 'Practical1' containing subfolders 'build' and 'nbproject', and a file 'src'. The 'src' folder is expanded, showing 'Libreria.java' and 'libro.java'. The main editor window shows the code in 'libro.java' with line numbers 29 to 41. The code defines a 'libro' class with a constructor and several 'set' methods. The 'Libreria' class uses these methods to create and configure a 'libro' object. The bottom panel, titled 'Output - Libreria (run)', shows the execution output, which includes prompts for book details and a summary of the book's data.

```
29      */
30      //      libro librol = new libro(titulo,autor,cantidad,prestados);
31
32      /*
33      * Crea libro con constructor vacío
34      */
35      libro librol = new libro();
36      librol.setTitulo(titulo);
37      librol.setAutor(autor);
38      librol.setNumLibros(cantidad);
39      librol.setNumPrestados(prestados);
40
41
```

**Output - Libreria (run)**

```
run:
Titulo del libro:
Java 8

Autor del libro:
Juan

Cantidad de ese libro:
3

Cantidad prestada de ese libro:
1

----Datos del libro---
Titulo del libro: Java 8
Titulo del libro: Juan
Cantidad en existencia: 3
Cantidad prestada: 1
BUILD SUCCESSFUL (total time: 21 seconds)
```