

cursos gratuitos
de formación profesional

Módulo 3:

Programación

Orientada a

Objetos

WORKING TIME

Es hora de que pongas en práctica todo lo aprendido. 🧐

Este apartado tiene el objetivo de ayudarte a seguir potenciando tus habilidades, por lo que a continuación encontrarás diferentes **desafíos** que podrás resolver de forma independiente y a tu ritmo.

Más adelante conseguirás las resoluciones para que valides tus respuestas y puedas monitorear tu progreso. 😊

¡Manos a la obra!

1. Introducción:

En este desafío deberás crear una clase que modele objetos del tipo Libro, y algunos métodos para poder interactuar con estos objetos.

Pondremos en práctica lo aprendido en el manual.

2. Desafío

Crear una clase llamada **Libro** que contenga los siguientes atributos: **ISBN (entero)** , **Título (String)**, **Autor (String)**, **Número de páginas (entero)**.

Además, la clase deberá tener un **constructor con todos los atributos pasados por parámetro y un constructor vacío** (constructor por defecto).

En la clase Libro, crear dos **métodos**:

- Método para cargar los datos de los atributos y crear un libro (ingreso de datos por consola). Este método deberá recibir un objeto Libro por parámetro, para luego, dentro del cuerpo de la función, pedir los datos de entrada.
- Método para informar todos los datos de los atributos del objeto (salida).

3. ¿Dónde se lleva a cabo?

Eclipse IDE

4. Tiempo de dedicación

1 hora

Resolución de desafíos

En primer lugar, utilizando el IDE Eclipse, crearemos un nuevo **Java Project**, y en este proyecto crearemos una “**Class**” y le pondremos su nombre haciendo referencia a los objetos que queremos modelar en esta clase. En este caso la clase se llamará **Libro**.

El árbol del proyecto debería verse de esta manera:



A continuación, comenzaremos con la **modelación de la clase Libro**. Debería verse de la siguiente manera:

```
1 import java.util.Scanner;
2
3 public class Libro {
4
5     //declaramos los atributos pedidos:
6     //ISBN (entero) , Título (String), Autor (String), Número de páginas (entero)
7
8     public int isbn;
9     public String titulo;
10    public String autor;
11    public int numeroPaginas;
12
13    // constructor con parámetros
14    public Libro(int isbn, String titulo, String autor, int numeroPaginas) {
15        this.isbn = isbn;
16        this.titulo = titulo;
17        this.autor = autor;
18        this.numeroPaginas = numeroPaginas;
19    }
20
21    // constructor vacío
22    public Libro() {
23    }
24 }
```

El siguiente paso es la creación de los métodos pedidos. Esto lo realizaremos en la misma clase en la que estamos trabajando (Libro), luego de los constructores:

```
// método para cargar los datos de los atributos.

public Libro crearLibro() {
    //creamos un Scanner para poder recibir datos por teclado
    Scanner scan = new Scanner(System.in).useDelimiter("\r\n");

    System.out.println("Ingresa el isbn");
    isbn = scan.nextInt(); //leemos cada ingreso y lo guardamos en el atributo declarada

    System.out.println("Ingresa el título del libro");
    titulo = scan.next();

    System.out.println("Ingresa el autor");
    autor = scan.next();

    System.out.println("Ingresa el número de páginas");
    numeroPaginas = scan.nextInt();

    // creamos el Libro enviando los datos guardados al constructor del objeto:

    Libro libro = new Libro(isbn, titulo, autor, numeroPaginas);

    // retornamos el objeto creado con todos los datos guardados.
    return libro;
}
```

Y luego desarrollamos el segundo método pedido, que en este caso está ejemplificado con el uso de la función “**toString()**” que nos brinda el sistema.

```
51 // creamos una cadena que muestra todos los datos del objeto.
52 //Esto podemos hacerlo gracias al método sobrescrito "toString()" que nos brinda Java.
53 // También puedes crear tu propio método de retorno de cadena, es a elección!
54 @Override
55 public String toString() {
56     return "Libro [isbn=" + isbn + ", titulo=" + titulo + ", autor="
57         + autor + ", numeroPaginas=" + numeroPaginas+ "]";
58 }
```

A continuación, veremos cómo quedaría nuestra clase ejecutable (Main), dónde haremos uso de las llamadas a los métodos para poder ejecutarlos.

```

1
2 public class Main {
3
4     public static void main(String[] args) {
5         //creamos un objeto Libro con su constructor vacío.
6
7         Libro libro = new Libro();
8
9         //a continuación le asignamos el valor que nos retorna el
10        //llamado a la función crearLibro() que creamos anteriormente.
11
12        libro = libro.crearLibro();
13
14        // por último, mostramos los datos por pantalla haciendo uso
15        // del método creado para tal fin.
16
17        System.out.println(libro.toString());
18    }
19
20 }
21

```

Por último, la entrada y salida por consola al ejecutar la main class debería verse de la siguiente manera:

```

Console X
<terminated> Main [Java Application] C:\Users\marie\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0
Ingresa el isbn
12345
Ingresa el título del libro
Ejercicio P00
Ingresa el autor
Anónimo
Ingresa el número de páginas
12
Libro [isbn=12345, titulo=Ejercicio P00, autor=Anónimo, numeroPaginas=12]

```



Muchas gracias



Ministerio de Trabajo,
Empleo y Seguridad Social
Argentina