

TEMA: POO UTILIZANDO JAVA. PARTE II

Taller de Programación.

Módulo: Programación Orientada a Objetos

Instanciar e iniciar objeto

- Hasta ahora, nuestro main ...

```
public class Demo01Libro {  
  
    public static void main(String[] args) {  
        Libro libro = new Libro();  
        libro.setTitulo("Java: A Beginner's Guide");  
        libro.setEditorial("Mcgraw-Hill");  
        libro.setAñoEdicion(2014);  
        libro.setPrimerAutor("Herbert Schildt");  
        libro.setISBN("978-0071809252");  
        libro.setPrecio(21.72);  
        ...  
    }  
}
```

Generar una clase para representar libros. Un Libro se caracteriza por: título, nombre del primer autor, editorial, año de edición, ISBN, precio

El libro debe saber:

- Devolver el valor de cada atributo.
- Modificar el valor de cada atributo.
- Devolver su representación en formato String.
Repr. *"Java: A Beginner's Guide por Herbert Schildt - 2014 - ISBN: 978-0071809252"*

Libro
titulo, primerAutor, editorial, añoEdicion, ISBN, precio
<code>String getTitulo()</code> ... <code>double getPrecio()</code> <code>void setTitulo(String unTitulo)</code> ... <code>void setPrecio(double unPrecio)</code> <code>String toString()</code>

Declaración de constructores.

- Se ejecuta tras alocar el objeto e inicializar las v.i. (por defecto o explícitamente).
- Objetivo: inicialización de v.i.
- Sintaxis

```
public NombreClase( lista de parámetros formales ) {  
    /* Código */  
}
```

- Si la clase no declara ningún constructor, Java incluye uno sin parámetros y sin código (*constructor nulo*).
- Instanciación de objeto:
NombreClase objeto= new NombreClase(lista de parámetros actuales);

Ejemplo (Hasta ahora) Libro miLibro = new Libro(); *//Invoca al constructor nulo.*

Declaración de constructores. Ejemplo.

```
public class Libro {  
    private String titulo;  
    private String primerAutor;  
    private String editorial;  
    private int añoEdicion;  
    private String ISBN;  
    private double precio;
```

```
    public Libro( String unTitulo, String unaEditorial,  
                 int unAñoEdicion, String unPrimerAutor,  
                 String unISBN, double unPrecio){  
        titulo = unTitulo;  
        editorial = unaEditorial;  
        añoEdicion= unAñoEdicion;  
        primerAutor = unPrimerAutor;  
        ISBN = unISBN;  
        precio = unPrecio;  
    }
```

```
    ....
```

```
}
```

Declaración de constructores. Ejemplo.

- Ejemplo instanciación (en main)

```
Libro libro1= new Libro( "Java: A Beginner's Guide", "Mcgraw-Hill",  
                        2014, "Herbert Schildt",  
                        "978-0071809252", 21.72);
```

- ¿Funciona ahora? Libro libro = new Libro();

Si el programador generó un constructor,
Java no incluye el constructor nulo.

Declaración de constructores. Sobrecarga. Ejemplo.

- Puede haber varios constructores para la clase (sobrecarga).
- Java identifica cuál está siendo invocado por el número y tipo de sus parámetros.
- *Por defecto quiero que el libro tenga año de edición 2015 y precio 100 => Otro constructor*


```
public class Libro {  
    private String titulo;  
    private String primerAutor;  
    private String editorial;  
    private int añoEdicion;  
    private String ISBN;  
    private double precio;  
  
    public Libro( String unTitulo, String unaEditorial,  
int unAñoEdicion, String unPrimerAutor, String unISBN, double unPrecio){  
        titulo = unTitulo;  
        editorial = unaEditorial;  
        añoEdicion= unAñoEdicion;  
        primerAutor = unPrimerAutor;  
        ISBN = unISBN;  
        precio = unPrecio;  
    }  
}
```

```
    public Libro( String unTitulo, String unaEditorial, String  
unPrimerAutor, String unISBN){  
        titulo = unTitulo;  
        editorial = unaEditorial;  
        añoEdicion= 2015;  
        primerAutor = unPrimerAutor;  
        ISBN = unISBN;  
        precio = 100;  
    }  
  
    public Libro(){  
  
    }  
    ...  
}
```

3 constructores distintos

Declaración de constructores. Sobrecarga. Ejemplo.

```
public class Demo01ConstructoresLibro {  
    public static void main(String[] args) {  
        Libro libro1= new Libro( "Java: A Beginner's Guide", "Mcgraw-Hill", 2014,  
                                "Herbert Schildt", "978-0071809252", 21.72);  
        Libro libro2= new Libro("Learning Java by Building Android Games",  
                                "CreateSpace Independent Publishing",  
                                "John Horton", "978-1512108347");  
        System.out.println(libro1.toString());  
        System.out.println(libro2.toString());  
        System.out.println("Precio del libro2: " +libro2.getPrecio());  
        System.out.println("Año edición del libro2: " +libro2.getAñoEdicion());  
        Libro libro3= new Libro();  
    }  
}
```



¿Funciona?

Interacción entre objetos (Ej 3)

- Normalmente un Prog. OO tiene objetos de distintas clases.
- Los objetos cooperan (enviándose mensajes) para llevar a cabo una tarea común ...
- **Antes:** nuestros libros consideraban el nombre del primer autor (String).
- **Ahora:** quiero que el libro conozca del primer autor nombre, biografía, etc
- ¿Qué estrategia seguir?

~~¿Incluir al libro todos los datos del primer autor
... y comportamiento?~~

¿Hacer que el libro conozca a un obj. autor?



Interacción entre objetos (Ej 3)

- Un libro conoce a su autor (obj).

Diagrama de clases

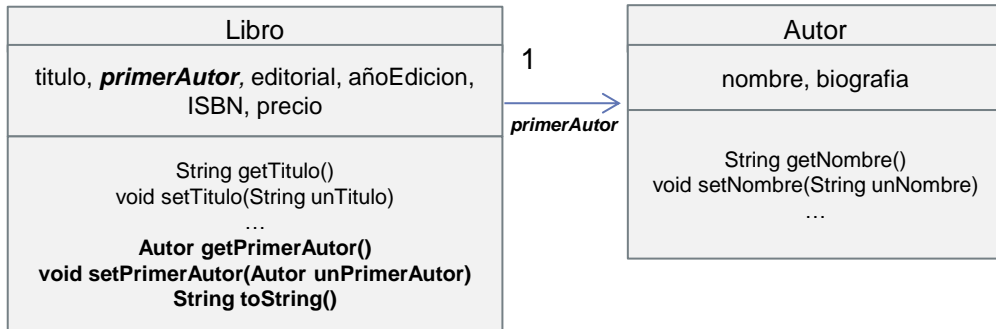
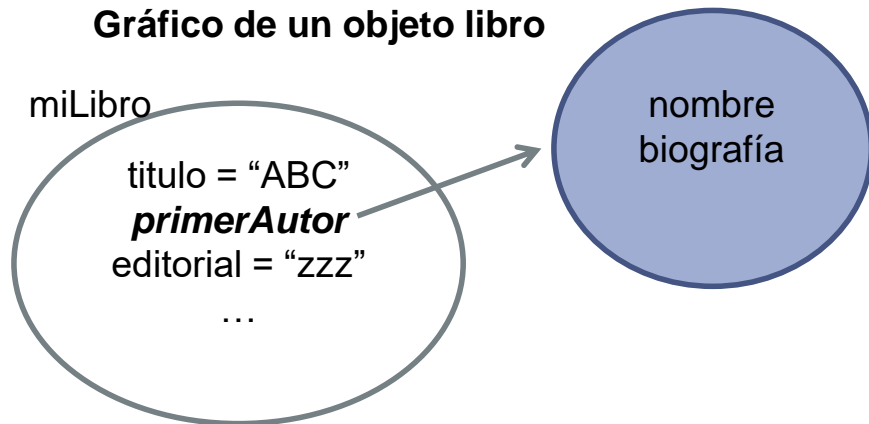


Gráfico de un objeto libro



- Modificaciones en el código

- Generar la clase Autor
- Modificar Libro (primerAutor ahora es instancia de Autor; encabezado de métodos setPrimerAutor, getPrimerAutor; método toString; constructor)
- ¿Cómo instancio un libro en el prog. ppal?

Interacción entre objetos (Ej 3)

- Dado un objeto libro ¿qué pasos sigo para imprimir el nombre del autor del libro?

Diagrama de clases

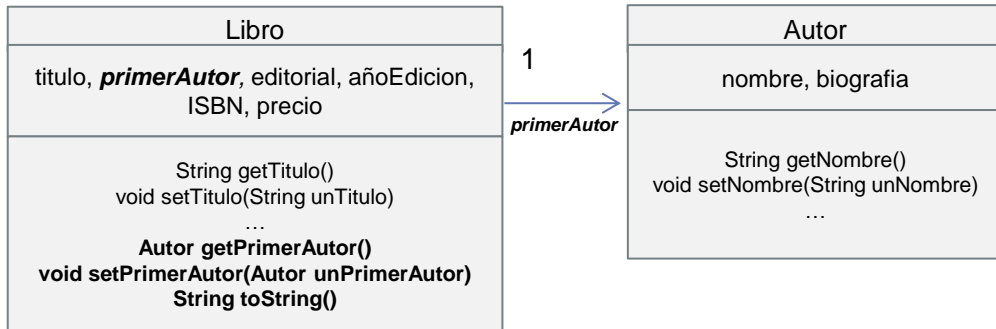
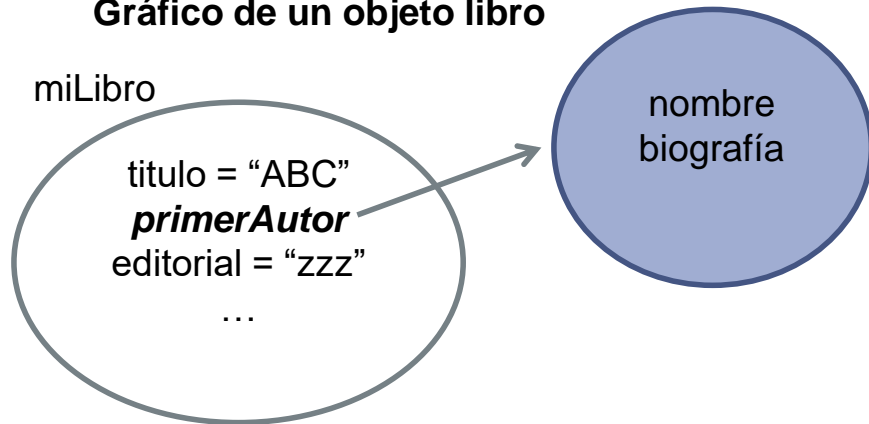


Gráfico de un objeto libro



En prog. ppal...

- Pido al objeto libro que me devuelva su autor ... ¿cómo?
- Pido al autor que me devuelva su nombre y lo imprimo ... ¿cómo?