

TEMA: DESARROLLO DE CLASES (PARTE II)

Taller de Programación.

Módulo: Programación Orientada a Objetos

Constructores. Introducción.

- Hasta ahora ...

seguimos los siguientes pasos para instanciar e iniciar un objeto

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

¿Existe una forma más sintética?

```
Libro miLibro = new Libro( datos, para, iniciar, el, objeto );
```

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

- Devolver el valor de cada atributo.
- Modificar el valor de cada atributo.
- Calcular su precio final (incluyendo IVA)
- Devolver su representación en formato String.

Libro
título, 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>double getPrecioFinal()</code> <code>String toString()</code>

Constructores. Introducción.

- **Constructor:** código que se ejecuta *tras* reservar memoria para el objeto e iniciar las v.i. al valor por defecto o explícito. Toma valores pasados en el mensaje de creación. Su objetivo es iniciar el objeto.
- **Declaración de constructor:** dentro de la clase en cuestión

```
public class MiClase{
    public MiClase( lista, de, parámetros, formales ){
        /* Código del constructor */
    }
}
```

Si la clase no declaró constructor,
Java incluye uno sin parámetros
y sin código (*constructor nulo*).

- **Instanciación de objeto:** en el main

(Hasta Ahora) `MiClase objeto= new MiClase();` //Invoca al constructor nulo

[illegible]

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;  
    }  
  
    //...  
}
```

Lleva el mismo nombre que la clase

No tiene valor de retorno

Recibe 0, 1, o más parámetros

Constructores. Ejemplo.

- **Ejemplo de instanciación:** en main

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

```
Libro libro3 = new Libro();
```

 ¿Funciona ahora?

Si el programador generó un constructor,

Java no incluye el constructor nulo.

Constructores. Sobrecarga.

- Una clase puede tener varios constructores. ¿Para qué?

Distintas maneras de iniciar el objeto instanciado



Editorial por defecto
Kapelusz y año de
edición actual

- Importante:**
 - Al instanciar el objeto (new) se ejecutará un sólo constructor.
 - Java identifica cuál ejecutar según el número y tipo de parámetros pasados al instanciar.
 - No puede haber dos constructores con igual número y tipo de parámetros.

Constructores. Sobrecarga. Ejemplo.

Editorial por defecto
Kapelusz y año de
edición actual


```
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 unPrimerAutor,  
              String unISBN,  
              double unPrecio){  
    titulo = unTitulo;  
    editorial = "Kapelusz";  
    añoEdicion= Calendar.getInstance().get(Calendar.YEAR);  
    primerAutor = unPrimerAutor;  
    ISBN = unISBN;  
    precio = unPrecio;  
}  
  
public Libro(){  
  
}  
  
//...  
}
```

3 constructores distintos

Constructores. Sobrecarga. Ejemplo.

```
public class Demo01Constructores {  
  
    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("Practicas del Lenguaje",  
                                "Asociacion Maestros Argentinos","978-9501314694",19.95);  
        System.out.println(libro1.toString());  
        System.out.println(libro2.toString());  
        System.out.println("Editorial del libro2: " + libro2.getEditorial());  
        System.out.println("Año edición del libro2: " + libro2.getAñoEdicion());  
        Libro libro3= new Libro();  
    }  
}
```

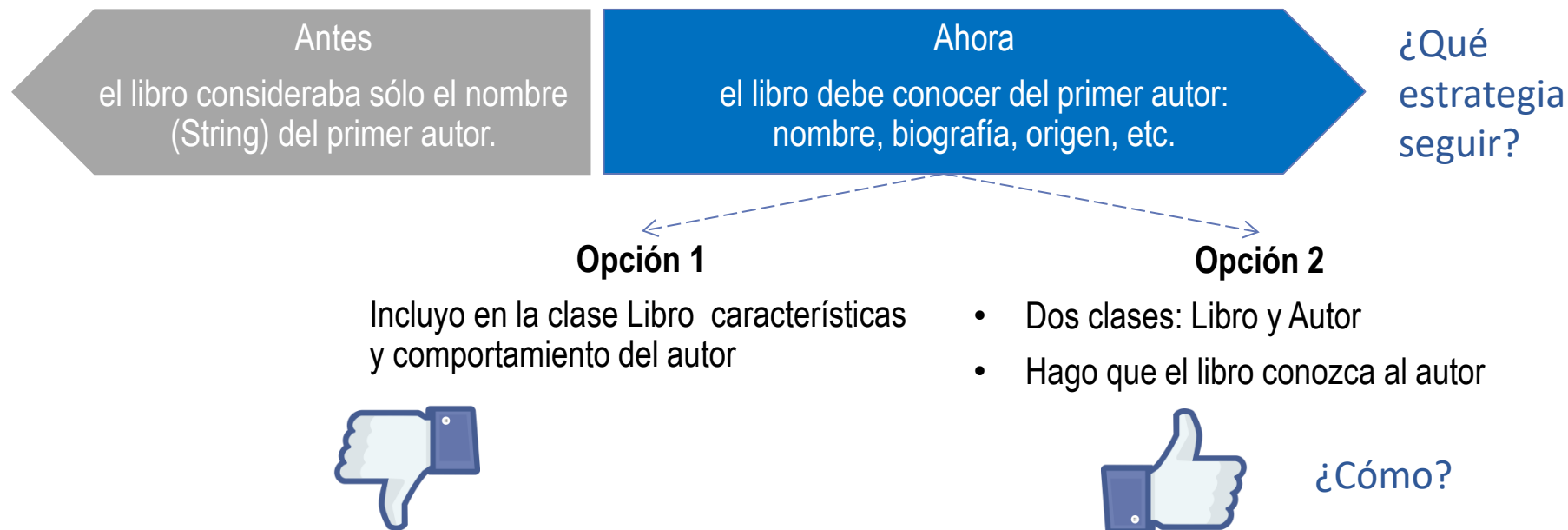


¿Funciona ahora? ¿Por qué?

Interacción entre objetos

- Normalmente un programa OO tiene objetos, de distintas clases, ...

que se relacionan y interactúan (vía mensajes) para llevar a cabo la tarea



Interacción entre objetos. Modificaciones.

Gráfico de un objeto libro

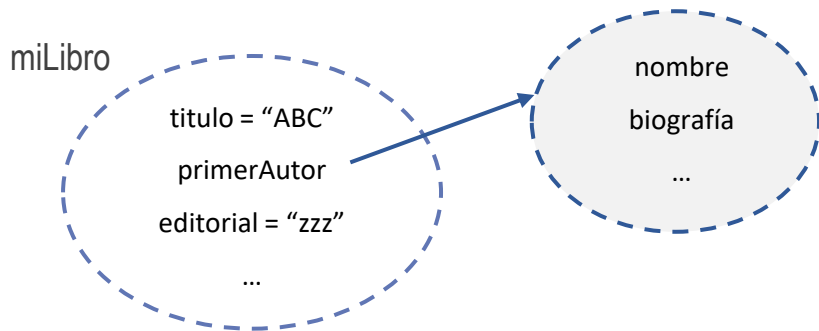
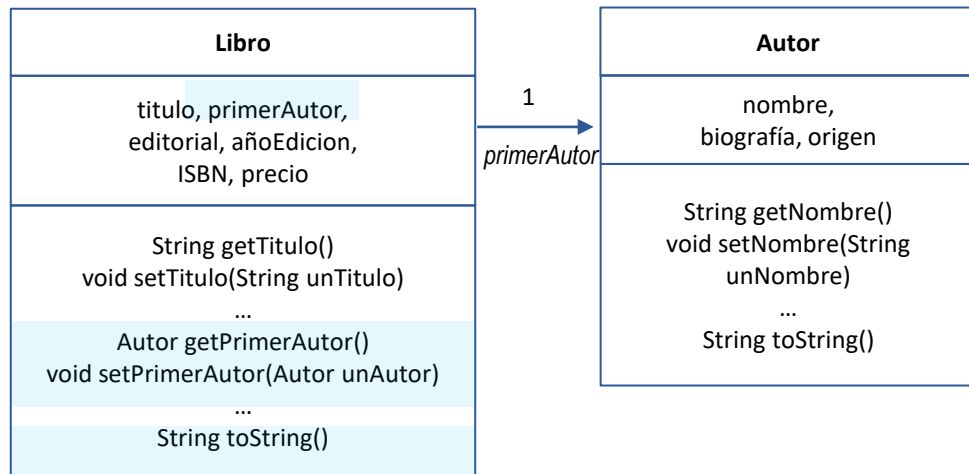


Diagrama de clases



- **Modificaciones en el código visto**

- Generar la clase Autor
- Modificar la clase Libro
- Modificar el programa

Interacción entre objetos. Modificación Libro.

```
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 String getTitulo(){
        return titulo;
    }

    public void setTitulo(String unTitulo){
        titulo = unTitulo;
    }
    ...
}
```

primerAutor ahora será instancia
de clase Autor

constructor ¿qué debe recibir?

getPrimerAutor
¿qué debe devolver?

setPrimerAutor
¿qué debe recibir?

¿cómo obtengo el nombre del
primerAutor?

```
...

public String getPrimerAutor() {
    return primerAutor;
}

public void setPrimerAutor(String unPrimerAutor){
    primerAutor=unPrimerAutor;
}

@Override
public String toString(){
    String aux;
    aux= titulo + " por " + primerAutor + " - " +
        añoEdicion + " - " + " ISBN: " + ISBN;
    return ( aux);
}
}
```

¿Cómo instancio un libro en el programa?

¿Cómo instancio dos libros de un mismo autor?

¿Cómo imprimo la representación del autor de un libro?

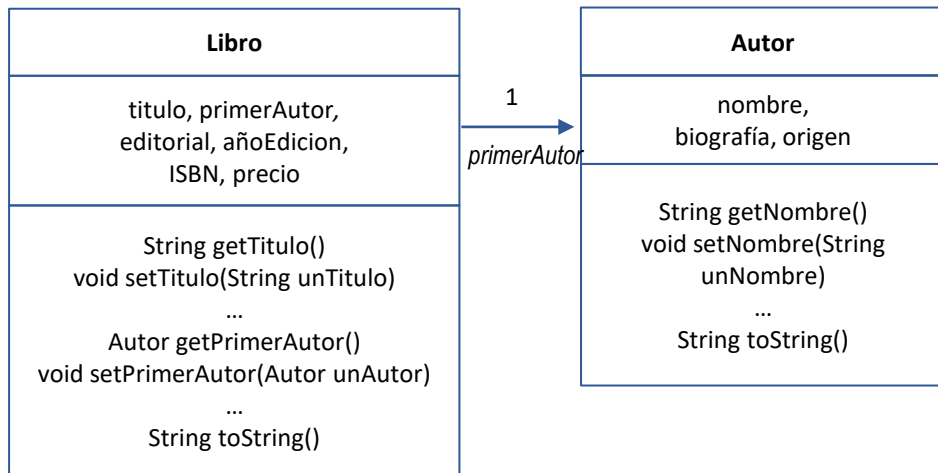
Interacción entre objetos. Modificación programa.

Diagrama de clases

En el programa dado el objeto miLibro ...

¿Cómo imprimo la representación del autor de miLibro?

- Pido a miLibro que me devuelva su autor ¿cómo?
- Pido al autor que me devuelva su representación e imprimo ¿cómo?



La referencia `this`. Introducción.

Dentro de un *método de instancia* o de un *constructor*, la referencia **this** representa al objeto que recibió el mensaje o que está siendo inicializado, respectivamente.



Veamos su uso

La referencia this. Autoenvío de mensaje.

- **Problema:** Incluir el precio final (con IVA) en la representación String del libro.

```
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 double getPrecioFinal(){  
            return precio + (precio*0.21);  
        }  
  
        public String toString(){  
            String aux = titulo + " por " +  
                primerAutor + " - " +  
                añoEdicion + " - ISBN: " +  
                ISBN + " - precio final: " +  
                ???;  
            return aux;  
        }  
    }
```

¿Cómo disparo la ejecución de un método desde otro?

¿A qué objeto le envío el mensaje?

La referencia this. Autoenvío de mensaje.

- **Problema:** Incluir el precio final (con IVA) en la representación String del libro.

```
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 double getPrecioFinal(){
        return precio + (precio*0.21);
    }

    public String toString(){
        String aux = titulo + " por " +
            primerAutor + " - " +
            añoEdicion + " - ISBN: " +
            ISBN + " - precio final: " +
            this.getPrecioFinal();

        return aux;
    }
}
```

this.nombreMétodo(parámetros)

El objeto que está ejecutando (**this**) se autoenvía un mensaje
La búsqueda del método a ejecutar inicia en la clase del objeto

La referencia this. Autoenvío de mensaje.

- **Problema:** Incluir el precio final (con IVA) en la representación String del libro.

```
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 double getPrecioFinal(){
        return precio + (precio*0.21);
    }

    public String toString(){
        String aux = titulo + " por " +
            primerAutor + " - " +
            añoEdicion + " - ISBN: " +
            ISBN + " - precio final: " +
            getPrecioFinal();

        return aux;
    }
}
```

Obviando la palabra **this** obtenemos el mismo efecto

La referencia this. Referirse a v.i. dentro de la clase.

- **Problema:** Dentro de un método o de un constructor tengo un parámetro nombrado igual que una variable de instancia.

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

```
    public void setTitulo(String titulo){  
        ??? = titulo;  
    }  
}
```

¿Cómo accedo a la variable de instancia
que quedó *ocultada*?

La referencia this. Referirse a v.i. dentro de la clase.

- **Problema:** Dentro de un método o de un constructor tengo un parámetro nombrado igual que una variable de instancia.

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

```
    public void setTitulo(String titulo){  
        this.titulo = titulo;  
    }  
}
```

Para referirse a la variable de instancia del objeto
usar **this.nombreVariableInstancia**

Más información sobre this en:

<https://docs.oracle.com/javase/tutorial/java/javaOO/thiskey.html>