

TEMA: POO UTILIZANDO JAVA. PARTE I

Taller de Programación.

Módulo: Programación Orientada a Objetos

Enunciado

- 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.
 - 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>

Definición de clases.

- Sintaxis

```
public class NombreDeClase {  
    /* Declaración del estado del objeto*/  
    /* Declaración de constructor(es) */  
    /* Declaración de métodos que implementan acciones */  
}
```

Declaración del estado.

- Estado interno:

Ejemplo

- Datos de tipos primitivos* TipoPrimitivo nombreDato; double precio;
- Referencias a otros objetos.* NombreDeClase nombreDato; String titulo;

- Anteponer a la declaración la palabra **private** para lograr encapsulamiento (ocultamiento de la información).

*Las v.i.s. **privadas** pueden ser accedidas sólo dentro de la clase que las declara*

private double precio;

- En la declaración del dato se puede dar un valor inicial (inicialización explícita).

private double precio = 10.5;

private String titulo = "Java: A Beginner's Guide";

Declaración del estado. Ejemplo.

```
public class Libro {  
    /* Declaración del estado */  
    private String titulo;  
    private String primerAutor;  
    private String editorial;  
    private int añoEdicion;  
    private String ISBN;  
    private double precio;  
  
    ....  
}
```

Los datos correspondientes al estado toman un valor por defecto cuando no se inicializan explícitamente.
(numéricos => 0; boolean => false; char => "; objetos => null)

¿Qué debo hacer si quiero que mis libros tengan por defecto año de edición 2015 y precio 100?

Declaración del comportamiento.

- Sintaxis

```
public TipoRetorno nombreMetodo ( lista de parámetros formales ) {  
    /* Declaración de variables locales al método */  
    /* Cuerpo del método */  
}
```

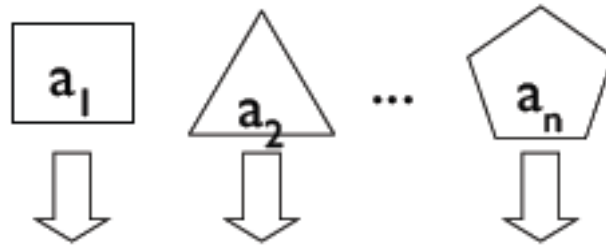
- **public:** indica que el método forma parte de la interfaz.
- **TipoRetorno:** tipo de dato primitivo / nombre de clase / void (no retorna dato).
- **nombreMetodo:** verbo seguido de palabras. Convención de nombres.
- **Lista de parámetros:** datos de tipos primitivos u objetos.
 - TipoPrimitivo nombreParam // NombreClase nombreParam
 - Separación por coma.
 - Pasaje por valor únicamente.
- **Declaración de variables locales.** Ámbito. Tiempo de vida. (Declaración idem que en Main)
- **Cuerpo.** Código puede utilizar estado y modificarlo (v.i.) – devolver resultado **return**

Declaración del comportamiento. Parámetros.

- Gráficamente

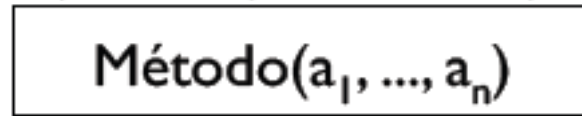
Envío de mensaje

Código llamador
queda pendiente



Parámetros actuales

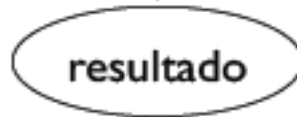
Ejecución del método



Parámetros formales

Retorno del resultado

El control vuelve al llamador



Valor de retorno (puede
no existir - *void*)

Declaración del comportamiento. Parámetros.

- Parámetros: únicamente pasaje por valor

a) *Parámetro dato primitivo:*

- **Parámetro formal** recibe **copia del valor** del parámetro actual .
- Si se modifica el parámetro formal, no altera el parámetro actual.

Main

```
Libro l1 = new Libro();  
...  
int x = 1;  
l1.hacerUno(x);  
System.out.println(x); ¿Qué imprime?
```

Imprime: 1

```
public class Libro{  
    ...  
    public void hacerUno(int y){  
        y++;  
    }  
}
```


Declaración del comportamiento. Parámetros.

- Parámetros: únicamente pasaje por valor

b) *Parámetro objeto:*

- **Parámetro formal** recibe **copia de la referencia** del parámetro actual.
- Si se modifica el estado interno del objeto parám. formal, el cambio en el estado es visible en el parám. actual.

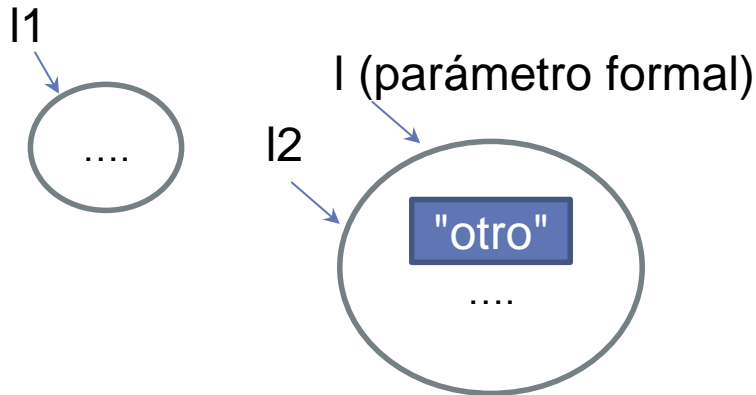
Main

```
Libro l1 = new Libro();  
Libro l2 = new Libro();  
l2.setTitulo("Java");  
...  
l1.hacerDos(l2);  
System.out.println(l2.getTitulo());
```

¿Qué imprime?

Imprime: "otro"

```
public class Libro{  
    ...  
    public void hacerDos(Libro l){  
        l.setTitulo("otro");  
    }  
}
```



Declaración del comportamiento. Parámetros.

- Parámetros: únicamente pasaje por valor

b) *Parámetro objeto:*

- **Parámetro formal** recibe **copia de la referencia** del parámetro actual.
- Si se modifica la referencia del parám. formal, el parám. actual sigue referenciando al mismo objeto.

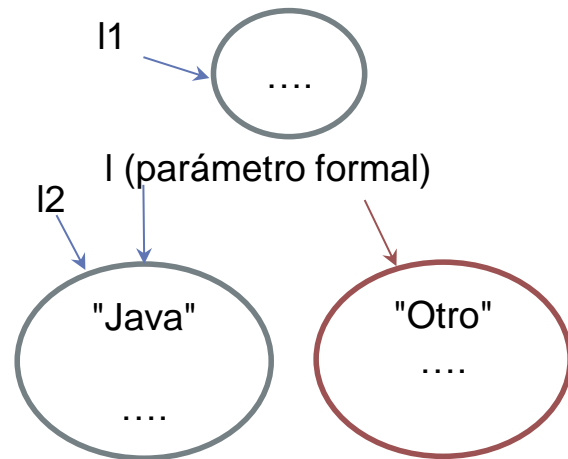
Main

```
Libro l1 = new Libro();  
Libro l2 = new Libro();  
l2.setTitulo("Java");  
...  
l1.hacerTres(l2);  
System.out.println(l2.getTitulo());
```

¿Qué imprime?

Imprime: "Java"

```
public class Libro{  
    ...  
    public void hacerTres(Libro l){  
        l = new Libro();  
        l.setTitulo("otro");  
    }  
}
```



Definición de clases. Ejemplo

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"

```
public class Libro {
```

```
    private String titulo;  
    private String primerAutor;  
    private String editorial;  
    private int añoEdicion;  
    private String ISBN;  
    private double precio;
```

**Estado
(características)**

**Métodos
(acciones)**

Libro.java

```
    public String getTitulo(){  
        return titulo;  
    }  
    public void setTitulo(String unTitulo){  
        titulo = unTitulo;  
    }  
    public double getPrecio{  
        return precio;  
    }
```

```
    ...  
    public void setPrecio(double unPrecio){  
        precio= unPrecio;  
    }  
  
    public String toString(){  
        String aux = titulo + " por " + primerAutor + " - " +  
            añoEdicion + " - ISBN: " + ISBN;  
        return aux;  
    }  
}
```

aux: variable local al método

Repaso. Instanciación (creación de objetos)

- **Declarar variable para mantener la referencia:**

`NombreDeClase miVariable;`

- **Enviar a la clase el mensaje de creación:**

`miVariable= new NombreDeClase();`

- ***Se puede unir los dos pasos anteriores:***

`NombreDeClase miVariable= new NombreDeClase();`

Ejemplo

`Libro libro;`

`libro = new Libro();`

`Libro libro = new Libro();`

- **Secuencia de pasos en la creación:**

- *Reserva de Memoria.* Las variables de instancia se inicializan a valores por defecto o explícito (si hubiese).
- *Ejecución del Constructor* (código para inicializar variables de instancia con los valores que enviamos en el mensaje de creación).
- *Asignación de la referencia a la variable.*

Repaso. Envío de mensaje al objeto

- Sintaxis

objeto.nombreMétodo(parámetros actuales);

Ejemplo *main*

```
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);  
System.out.println(libro.toString());
```

Demo01Libro.java

Output - Pruebas (run)

run:

Java: A Beginner's Guide por Herbert Schildt - 2014 - ISBN: 978-0071809252
BUILD SUCCESSFUL (total time: 0 seconds)

IDE NetBeans

Crear Clase Java

- Click derecho sobre la carpeta contenedora
Ej: "tema 3"
- New → Java Class
- *Class Name*: Poner un nombre
- Finish

Aparecerá un archivo .java con el esqueleto de la clase