

Programación Orientada a Objetos

Profesora Claudia Cappelletti

Definición de la clase Libro y Aplicación

- 1) Definiciones.
- 2) Especificación de la clase Libro.
- 3) Aplicación.
- 4) Implementación de la clase Libro en Smalltalk.
- 5) Actividad 4.



Definiciones

1) Diseño de una nueva clase:

Se debe especificar la clase para definir el protocolo o vista externa de la nueva clase.

Protocolo de clase: es la descripción del protocolo entendido por una clase.

Protocolo de instancia: es la descripción del protocolo entendido por las instancias de una clase.

Variables de clase: son variables cuyo valor es compartido por todas las instancias de una clase.

Variables de instancia: denotan la información privada o estado de una instancia de una clase.



Especificación de la clase Libro

2) Especificación de la clase Libro

- **Clase Libro**

Subclase de: Object

Variables de instancia: isbn titulo autor editorial estado dni

Métodos de clase:

>>crearLibro isbn:unIsbn tit:unTit aut:unAut edit:unaEdit

“Retorna una instancia de Libro inicializada”

Métodos de instancia:

>>iniLibro isbn:unIsbn tit:unTit aut:unAut edit:unaEdit

“Inicializa una instancia de Libro”



Especificación de la clase Libro

>>verIsbn

“Retorna el isbn del libro”

>>verTitulo

“Retorna el título del libro”

>> verAutor

“Retorna el autor del libro”

>> verEditorial

“Retorna la editorial del libro”

>> verEstado

“Retorna el estado del libro”

>> verDni

“Retorna el dni del libro”



Especificación de la clase Libro

>>modilsbn:unlsbn

“Modifica el isbn del libro”

>>modiTítulo:unTit

“Modifica el título del libro”

>> modiAutor:unAut

“Modifica el autor del libro”

>> modiEditorial:unaEdit

“Modifica la editorial del libro”

>> modiEstado

“Modifica el estado del libro”

>> modiDni:unDni

“Modifica el dni del libro”



Aplicación

3) Aplicación: Mostrar el título del libro con menor isbn si los libros tienen el mismo autor.

|l1 l2 i t a e|

“creo y cargo una instancia de un libro y la guardo en l1”

i:= Prompter prompt: 'Ingrese un isbn'.

t:= Prompter prompt: 'Ingrese un título'.

a:= Prompter prompt: 'Ingrese un autor'.

e:= Prompter prompt: 'Ingrese una editorial'.

l1:= Libro crearLibro isbn:i tit:t aut:a edit:e.



Aplicación

“creo y cargo una instancia de otro libro y la guardo en l2”

i:= Prompter prompt:'Ingrese un isbn'.

t:= Prompter prompt:'Ingrese un título'.

a:= Prompter prompt:'Ingrese un autor'.

e:= Prompter prompt:'Ingrese una editorial'.

l2:= Libro crearLibro isbn:i tit:t aut:a edit:e.

(l1 verAutor=l2 verAutor) ifTrue:[

 (l1 verIsbn < l2 verIsbn) ifTrue:[(l1 verTitulo) inspect]

 ifFalse:[(l2 verTitulo) inspect]

 ifFalse:[^'Los libros no tienen el mismo autor'].



Implementación de la clase Libro en Smalltalk

4) Implementación de la clase Libro

- **Clase Libro**

Subclase de: Object

Variables de instancia: isbn título autor editorial estado dni

Métodos de clase:

>>crearLibro isbn:unIsbn tit:unTit aut:unAut edit:unaEdit

“Retorna una instancia de Libro inicializada”

^(self new) iniLibro isbn:unIsbn tit:unTit aut:unAut edit:unaEdit.



Implementación de la clase Libro en Smalltalk

Métodos de instancia:

```
>>iniLibro isbn:unIsbn tit:unTit aut:unAut edit:unaEdit
```

“Inicializa una instancia de Libro”

```
isbn:=unIsbn.
```

```
titulo:=unTit.
```

```
autor:=unAut.
```

```
editorial:=unaEdit.
```

```
estado:=false.
```

```
dni:=0.
```



Implementación de la clase Libro en Smalltalk

>>verIsbn

“Retorna el isbn del libro”

^ isbn.

>>verTitulo

“Retorna el título del libro”

^ titulo.

>> verAutor

“Retorna el autor del libro”

^ autor.

>> verEditorial

“Retorna la editorial del libro”

^ editorial.



Implementación de la clase Libro en Smalltalk

```
>> verEstado
```

```
    “Retorna el estado del libro”
```

```
    ^ estado.
```

```
>> verDni
```

```
    “Retorna el dni del libro”
```

```
    ^ dni.
```

```
>> modIsbn:unIsbn
```

```
    “Modifica el isbn del libro”
```

```
    isbn:=unIsbn.
```

```
>> modiTit:unTit
```

```
    “Modifica el título del libro”
```

```
    tit:=unTit.
```



Implementación de la clase Libro en Smalltalk

```
>> modiAutor:unAut
```

```
    “Modifica el autor del libro”
```

```
    autor:=unAut.
```

```
>> modiEditorial:unaEdit
```

```
    “Modifica la editorial del libro”
```

```
    editorial:=unaEdit.
```

```
>> modiEstado
```

```
    “Modifica el estado del libro”
```

```
    estado:=estado not.
```

```
>> modiDni:unDni
```

```
    “Modifica el dni del libro”
```

```
    dni:=unDni.
```



5) Actividad 4

- 1) a) Especificar e implementar la Clase Punto.
 - b) Realizar una aplicación que calcule y retorne la distancia entre dos puntos.
Desarrolle las siguientes aplicaciones en un Workspace.
- a) Para realizar este punto tener en cuenta que tenemos un objeto que llamaremos unPunto, que representa una posición en la pantalla.
Dicha posición está dada por una coordenada x y otra coordenada y.
El objeto unPunto, que pertenece a la Clase PuntoDelPlano, entiende los siguientes mensajes:
 - verx retorna la coordenada x del objeto receptor
 - very retorna la coordenada y del objeto receptor
 - modix: unX modifica la coordenada x del objeto receptor con unX
 - modiy: unY modifica la coordenada y con unYEl método de clase para crear un punto del plano es:
crearPuntoX:unX Y:unY



5) Actividad 4

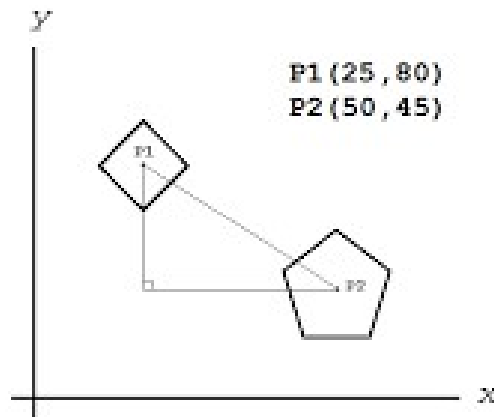
b) Para realizar este punto considerar que la fórmula que vamos a necesitar es la siguiente:

Fórmula de distancia entre dos puntos en el plano cartesiano.

La **distancia entre dos puntos** P_1 y P_2 del plano se denota por $d(P_1, P_2)$.

La **fórmula** de la **distancia** usa las coordenadas de los puntos.

Esta **fórmula** puede ser deducida a partir del Teorema de Pitágoras.



$$\begin{aligned} P_1P_2 &= \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \\ &= \sqrt{(50 - 25)^2 + (45 - 80)^2} \\ &= \sqrt{(25)^2 + (-35)^2} \\ &= \sqrt{625 + 1225} \\ &= \sqrt{1850} \\ &= \boxed{43} \end{aligned}$$