



Programación Orientada a Objetos

Clase 3 - Smalltalk. Diseño de clases simples.

PARADIGMAS DE PROGRAMACION

UTN - La Plata



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 3.

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. Contiene los métodos de creación de objetos y manipulación de variables de clase.

Protocolo de instancia: es la descripción del protocolo entendido por las instancias de una clase. Contiene los métodos de manipulación de las variables de instancia de un objeto.

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

Permite almacenar los datos de un libro de la biblioteca y el dni del socio que lo retiró

- **Clase Libro**

Subclase de: Object

Variables de instancia: isbn titulo autor editorial estado dni

Métodos de clase:

>>**crearLibro**isbn:unlsbn **tit:**unTit **aut:**unAut **edit:**unaEdit

“Retorna una instancia de Libro inicializada”

Métodos de instancia:

>>**iniLibro**isbn:unlsbn **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 socio que sacó el libro; se le asigna 0 si no está prestado”

Especificación de la clase Libro

>>**modilsbn**:unIsbn

“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 socio que retiró el libro; se debe asignar un 0 si nadie lo tiene”

Aplicación

3) Aplicación: Crear 2 libros y luego mostrar el título del libro con menor isbn si los libros tienen el mismo autor.

| lib1 lib2 i t a e|

En Dolphin NO es necesario declarar las variables temporales **en la aplicación**. En Smalltalk Express si deben declararse.

“creo y cargo una instancia de un libro y la guardo en lib1”

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

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

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

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

lib1:= Libro **crearLibro**isbn: i **tit:** t **aut:** a **edit:** e.

Aplicación

“creo y cargo una instancia de otro libro y la guardo en lib2”

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

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

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

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

lib2:= Libro **crearLibro** isbn: i **tit:** t **aut:** a **edit:** e.

(lib1 **verAutor**=lib2 **verAutor**) ifTrue:[

 (lib1 **verIsbn** < lib2 **verIsbn**) ifTrue:[(lib1 **verTitulo**) inspect]

 ifFalse:[(lib2 **verTitulo**) inspect]

]

ifFalse:[Transcript nextPutAll: '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:unlsbn tit:unTit aut:unAut edit:unaEdit

“Retorna una instancia de Libro inicializada”

^(self new) **iniLibro** isbn:unlsbn **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. “se asigna false cuando no está prestado”

dni:=0. “se asigna 0 en la creación ya que no está prestado”

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 socio que retiró el libro”

^ dni.

>> **modifsbm:unlsbn**

“Modifica el isbn del libro”

isbn:=unlsbn.

>> **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. “le asigna el valor opuesto al estado”

>> **modiDni:**unDni

“Modifica el dni del socio que retiró el libro; asignar 0 en caso de que no lo tenga nadie”

dni:=unDni.

5) Actividad 3

1) a) Especificar e implementar la Clase Punto.

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:

posx retorna la coordenada x del objeto receptor

posy retorna la coordenada y del objeto receptor

modx: otrox modifica la coordenada x del objeto receptor con otroX

mody: otroy modifica la coordenada y con otroy

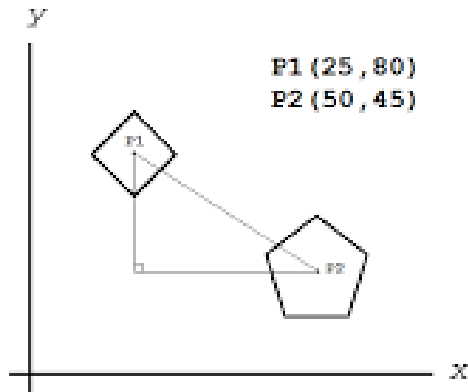
El método de clase para crear un punto del plano es:

crearConX: abscisa **conY:** ordenada

5) Actividad 3

- 1) b) Realizar una aplicación que calcule y retorne la distancia entre dos puntos.
Desarrolle la siguiente aplicación en un Workspace,

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



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

