

Metodologías de programación II

Práctica Extra 2

1) a) En las siguientes expresiones indicar quién es el objeto receptor, cuál es el mensaje, el selector, cuáles son los argumentos del mensaje y cuál es el valor de retorno.

b) Dar un ejemplo de polimorfismo.

- i) 'casa' isNil.
- ii) $9 + 3 * 2$.
- iii) true & false.
- iv) #(12 65 'olas' true) includes: 'viento'.
- v) $3 * 2$ squared.
- vi) $4 + 2$ negated between: $3 + 4 * 5$ and: 'hello' size * 10.
- vii) 'objetos' includes:\$e.
- viii) #('alumno', 'profesor', 'aula') size.
- ix) 'superclase' size.
- x) 'an', 'te', 'ayer'.
- xi) #(4 6 5 3) at: 2 put:#(2,3).

2) Dar un ejemplo de jerarquía de herencia orientada a objetos, donde cada nivel es una forma más especial del precedente. Dar otro ejemplo de jerarquía de la vida real que no cumpla con la relación de herencia (ejemplo: organigrama de una empresa).

3) Escribir un código Smalltalk para:

i) If (num > 6) then
begin
b:= b+ 1
num:= num – 1
end
else
num:=num + 1;

ii) Escribir un código Smalltalk que sume y retorne los números múltiplos de 7 que no sean pares, dentro de los 100 primeros números naturales.

4) Uso del ambiente Smalltalk.

a) Probar las siguientes expresiones en el ambiente usando un Workspace

- i) #('alumno' 'profesor' 56 'aula') size.
- ii) 25 notNil.
- iii) $1 + 2$ negated.
- iv) $1 + 2 * 3$.
- v) $4 + 8$ factorial between: $3 + 4 * 10$ and: 'hola' size * 8.
- vi) 'objetos' includes:\$e.
- vii) ('an', 'te', 'ayer') size.
- viii) 6 class.

- ix) Integer superclass.
- x) 'hola' isKindOf: Integer.
- xi) (2/3) inspect.
¿Para qué sirve un Inspector? ¿qué información contiene?

b) ¿Qué responde Smalltalk cuando se le pide evaluar las siguientes expresiones?

```
|n m x|
n:=3.
m:=1.
x:= ((n+m) factorial) squared.
```

c) Uso de la ventana de Transcripción del Sistema (System Transcript).

Probar las siguientes expresiones todas juntas y determinar qué hace cada una de ellas:

- i) Transcript nextPutAll: 'Página de prueba'.
- ii) Transcript cr.
- iii) 'EDAD :' printOn: Transcript.
- iv) Transcript space.
- v) 56 printOn: Transcript.

d) Operaciones de interacción con el usuario: uso de interrogadores, cuadros de mensajes e información. Uso de la clase Prompter (interrogador, para ingreso de datos) y de la Clase MessageBox (cuadro de confirmación y/o de información) . Probar los siguientes mensajes en un Workspace.

- i) nombre:= Prompter prompt: 'Ingrese su nombre' default: ' ' .
¿De qué clase es el objeto que retorna este mensaje?
precio:= Prompter prompt: 'ingrese un precio' default: ' ' . ¿Y en este caso de qué clase es el objeto ingresado?
- ii) MessageBox message: 'Es una prueba nada más'.
¿Qué hace este mensaje?
- iii) sigue:= MessageBox confirm: '¿Desea seguir ingresando datos?'.
¿De qué clase es el objeto que retorna este mensaje?
- iv) ¿Cuál es la diferencia entre estos tres mensajes?

5) Desarrolle las siguientes aplicaciones en un Workspace.

a) 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 que 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

Usando estos métodos, desarrolle una aplicación en la cual:

- i) Cree dos puntos del plano
- ii) Determine cuál es la distancia entre los mismos
- iii) Indique si alguno de los puntos pertenece al 1º cuadrante.

b) Suponer que se tiene una grilla que representa una ciudad de 50*50 cuadras.

En dicha ciudad se mueve un robot que entiende los siguientes mensajes.

-crearRobot:unNombre

“Retorna al robot posicionado en la esquina (1,1) con orientación hacia el Norte, con el nombre unNombre”

-mover

“Posiciona al robot en la esquina siguiente sin modificar su orientación”

-derecha

“Rota 90° hacia la derecha al robot”.

-izquierda

“Rota 90° hacia la izquierda al robot”

-posx

“Retorna la coordenada x de la esquina donde está posicionado el robot”

-posy

“Retorna la coordenada y de la esquina donde está posicionado el robot”

-posx:unX posy:unY

“Posiciona al robot en las esquinas x e y”.

Desarrollar una aplicación que permita crear un robot llamado ‘Rolo’, posicionarlo en la esquina (20,30) y hacer que recorra un rectángulo de 5 cuadras de altura por 10 cuadras de ancho.

- 6) a) Especificar la clase COMPRA, teniendo en cuenta que una compra posee un monto, día, hora, nombre del vendedor y cantidad de productos comprados.
- b) Escribir una aplicación que permita crear y cargar dos instancias de la clase Compra y retorne el nombre del vendedor de la compra de mayor monto.