

Testing can be used to show the presence of bugs, but never to show their absence - E. Dijkstra

Práctica nº 7

Ejercicio 1

Un alumno de la Universidad modeló e implementó las pilas en Smalltalk con una clase `Stack`, pero no creó tests de unidad. Usted deberá proveer tests de unidad sabiendo que en la clase se definieron los siguientes mensajes:

```
>> new
    que crea una pila nueva vacía

>> push: obj
    que agrega obj al tope de la pila

>> pop
    que remueve el objeto que está en el tope de la pila o nil en caso que no haya
    elementos en la pila

>> isEmpty
    que indica si la pila tiene o no elementos

>> top
    que retorna el objeto que está en el tope de la pila, sin removerlo de ella. Si no hay
    elementos, retorna nil.

>> size
    que retorna la cantidad de elementos que contiene la pila
```

Considere que el comportamiento de las pilas se puede explicar a partir de las siguientes propiedades (considere que `e` es una instancia de `Stack` y `obj` un objeto cualquiera):

```
(Stack new) isEmpty ≡ true
e push: obj; isEmpty ≡ false
e push: obj; top ≡ obj
e push: obj; pop ≡ obj
Stack new size ≡ 0
e push: obj; size ≡ 1 + e size
```

Ejercicio 2

Realice los diagramas de clase de los ejercicios:

- a - Práctica 3) ejercicio 4
- b - Práctica 4) ejercicio 6
- c - Práctica 5) ejercicio 10
- d - Práctica 6) ejercicio 6 y 7.

Ejercicio 3

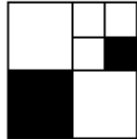
Un objeto Topografía representa la distribución de agua y tierra de una región cuadrada del planeta. Está formada por porciones de agua y de tierra.



Si la topografía está totalmente formada por agua, tiene el siguiente aspecto



Si la topografía está totalmente formada por tierra, tiene el siguiente aspecto



Si la topografía está formada por partes de agua y tierra, se descompone en 4 topografías de agua o tierra. Éstas a su vez podrían descomponerse en 4 más y así sucesivamente.

En la figura compuesta del ejemplo, la proporción de agua es 5/16. La proporción siempre es un valor entre 0 y 1.

a) Implementa las clases necesarias para que sea posible:

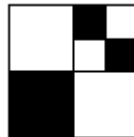
_ crear topografías

_ calcular la proporción de agua y tierra de una topografía dada

_ comparar igualdad entre topografías (dada por igualdad proporción de agua y tierra e igual distribución)

No es correcto construir una topografía compuesta por cuatro topografías del mismo tipo (sólo agua o sólo tierra).

b) Escribir las sentencias Smalltalk necesarias en el ambiente de VisualWorks, para poder reproducir la siguiente tipografía. Realice además el diagrama de objetos correspondiente.



Ejercicio 4

Realice *test cases* para verificar el correcto funcionamiento de las clases de los ejercicios:

a - Práctica 5) ejercicio 5

¿Será necesario escribir tests para todos los métodos del `BoundedBag`? Suponiendo que ya existe una clase de test para `Bag`, ¿de dónde cuelga la clase de test de `BoundedBag`?

b - Práctica 5) ejercicio 6.

c - Práctica 5) ejercicios 8 y 9.

d - Práctica 5) ejercicio 11.

e - Práctica 5) ejercicio 12.

f - Práctica 6) ejercicio 4.

Ejercicio 5

Escriba un diagrama de estados UML para describir los estados posibles en que se puede encontrar

a) un pedido de comida para delivery. Ej: tomado, iniciado, descartado, en camino, entregado.

b) un teléfono. Ej: libre (colgado), levantado, marcando número, ocupado, sin línea, etc).

Ejercicio 6

Modele el famoso juego piedra papel y tijera y realice los tests correspondientes.

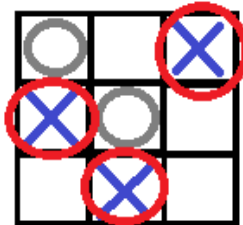
Ejercicio 7

Dos amigos estaban aburridos en casa y pensaron en jugar al TA-TE-TI, pero lo único que tenían a mano era una pc con Smalltalk, así que procedieron a programarlo.

- 1) Realice los tests correspondientes.
- 2) Realice la clase `Tablero`, la cual responde a.
- 3) `#ponerEnX: coordX y: coordY`, que pone la ficha en la posición `coordX@coordY`
`#limpiarTablero`, que reinicia el juego
`#ganador`, que devuelve quien es el ganador
`#jugadorActual`, que devuelve quien es el jugador actual

Para la realización de estos métodos es importante tener en cuenta que existen 2 tipos de fichas "X" y "O", el primer jugador siempre es la "X" y después de poner la ficha el cambio de jugador se hace automáticamente.

Los amigos después de un rato decidieron que el juego era un poco aburrido, entonces decidieron cambiar un poco las reglas, ahora no solo se gana completando una fila, columna o diagonal, sino que también si completan alguna diagonal corta, por ejemplo



Las celdas rodeadas en rojo son las que representan la diagonal corta.

- 4) Agregue el nuevo comportamiento en la clase.
- 5) Testee las nuevas formas de ganar
- 6) ¿Tuvo que modificar el test? ¿por qué?

Ejercicio 8

Haga el diagrama de objetos que representa a la familia Simpson, utilizando el ejercicio 1 de la practica 3.

Ejercicio 9

Los conjuntos matemáticos poseen algunas propiedades interesantes respecto de la operación de unión, que escribimos a continuación:

Siendo A y B conjuntos cualesquiera,

- idempotencia: $A \cup A = A$

- asociatividad: $(A \cup B) \cup C = A \cup (B \cup C) = A \cup B \cup C$

- conmutatividad: $A \cup B = B \cup A$

- identidad: $A \cup \emptyset = A$

Escriba distintos tests para verificar si los conjuntos en Smalltalk (Set) cumplen estas propiedades.