

Ejercicio 1

Utilizando la herramienta Ozono cree una lección llamada “Contador” e implemente un objeto contador que entienda los mensajes `#ponerACero` e `#incrementar`. Escriba los tests que muestren que dichos mensajes funcionan correctamente. Analice qué casos de prueba debe considerar.

- Extienda el contador para que pueda contar hasta un determinado valor y al llegar a dicho valor se coloque en 0. Modifique los tests existentes y agregue casos nuevos acorde a los cambios en su modelo.

Nota: los números entienden el mensaje `\\`, que retorna el módulo de la división entre dos números.

- Extienda ahora el contador para que el valor inicial no sea 0, sino un número arbitrario. Modifique los tests existentes y agregue casos nuevos acorde a los cambios en su modelo.

Nota: los números entienden el mensaje `#max:`, que retorna el mayor de dos números.

- Guarde la imagen y/o exporte la lección. En Ozono, diríjase a la referencia del contador y seleccione del menú contextual la opción “Eliminar referencia a este objeto”. Diríjase al diagrama de instancias en Ozono; ¿qué diferencia nota? ¿puede enviarle mensajes a dicho objeto? ¿porqué? ¿qué sucede al ejecutar los tests? Finalmente, vaya a la papelera y seleccione “Recuperar objeto”, utilizando el mismo nombre que tenía anteriormente. Ejecute nuevamente los tests y verifique que pasan correctamente.

Ejercicio 2

Utilizando la herramienta Ozono cree una lección llamada “Robot” e implemente un objeto “robot” con las siguientes características:

- Un robot posee una batería en la que almacena su energía. La carga de la batería se encuentra en el rango de 0 a 100.
- Un robot puede:
 - `#caminar: metros`. Caminar le consume 1 unidad de batería cada 10 metros que recorre.
 - `#cargarBateria: unaCarga`. Le agrega `unaCarga` unidades de carga a la batería (recordar que la carga de la batería no puede superar las 100 unidades).
 - `#dispararA: unObjetivo`. Genera un disparo y le consume el 10% de la carga de la batería.

Realice los tests correspondientes para verificar el correcto funcionamiento del robot. Indique con un breve comentario en cada test cuál es el propósito del mismo.

Ejercicio 3 (Entregar - Fecha tope: 25/08/2015)

Extienda la lección “Robot” agregando un objeto “alien”. Un alien tiene una energía, que está en el rango de 0 a 5 y comprende los siguientes mensajes:

- `#estaVivo` - Retorna verdadero sólo si su energía es mayor a 0.
- `#recibirDisparo` - Decrementa la energía del alien en 1.
- `#reponerse` - Aumenta la energía del alien en 2 unidades.

Realice los tests correspondientes para verificar el correcto funcionamiento del alien. Indique con un breve comentario en cada test cuál es el propósito del mismo.

Extienda el mensaje `#dispararA: unObjetivo` en el objeto “robot” para que, además de consumir energía al robot, le indique a `unObjetivo` que recibió un disparo.

Modifique los tests correspondientes al robot para verificar el correcto funcionamiento de ésta modificación.

Nota: La entrega de éste ejercicio debe constar del archivo generado al exportar la lección de la herramienta Ozono. Utilice el menú contextual de la lección y seleccione “Exportar”; esto generará un archivo con el nombre de la lección (`robot.ob`) en la carpeta “`./shared`”.

Ejercicio 4

Extienda la lección “Robot” agregando un objeto “mosquito” tal que el siguiente test se ejecute exitosamente:

```
mosquito inicializar.  
self assert: mosquito recibioDisparo not.  
robot dispararA: mosquito.  
self assert: mosquito recibioDisparo.
```

Pasos sugeridos:

1. Identifique los mensajes que debe entender el objeto “mosquito”.
2. Implemente dichos mensajes y escriba los tests que prueben su correcto funcionamiento.
3. Implemente el test planteado originalmente y verifique que se ejecute correctamente.

Ejercicio 5

Sintaxis Smalltalk:

1. ¿Qué significa “self”? ¿Para que se utiliza?
2. ¿Cuál es el uso que se da al punto en Smalltalk? ¿y al punto y coma?
3. Identifique objetos y mensajes y compute los resultados de las expresiones paso a paso (utilice un workspace cuando sea necesario):
 - 42 negated.
 - 3 + 2 * 4.
 - (3 + 2) * 4.
 - 3 + (2 * 4).
 - 'Hola Mundo' size + 1.
 - ' Robot ' trim size.
 - ' Robot ' trim; size.
 - 10 >= 4 factorial negated.
 - 'Hola Mundo' findString: 'Mun' , 'do'.
 - 1@2 min: 2@1+1 max: 'Hola' size@1.