

Ejercicio 1

Implemente en Pharo un modelo de cuentas bancarias como se especifica en el Ejercicio 1 de la Práctica 2 de Programación 3¹. Su modelo debe generar una excepción en el caso que se desee extraer un monto mayor al disponible. Realice los tests necesarios para probar el correcto funcionamiento de su modelo.

¹ Diseñar una modelo de cajas de ahorro cuyo saldo nunca puede ser menor que 0. Las acciones que se pueden llevar a cabo con una caja de ahorro son depositar dinero, extraer dinero y consultar su saldo.



Ejercicio 2

Extienda el modelo del ejercicio anterior agregando un número de cuenta a las cuentas bancarias. Modele una clase Banco que posea una colección de cuentas bancarias. Implemente el mensaje Banco>>obtenerCuenta: unNumeroDeCuenta que retorne la cuenta bancaria correspondiente a dicho número, generando una excepción en el caso que no exista ninguna cuenta con dicho número. Evalúe la información debe llevar consigo dicha excepción. Realice los tests necesarios para probar el correcto funcionamiento de su modelo.

Ejercicio 3

Modifique el Ejercicio 3 de la Práctica 3 para que su sistema genera una excepción en el caso que no haya ningún vehículo que pueda satisfacer una demanda (ej. una flota sólo basada en motos para la cual se pide llevar un envío de 10kg). Realice los cambios necesarios a los tests para verificar esta nueva funcionalidad.

Ejercicio 4

Los qbitos son un pueblo que basa su economía en la forestación. Un mapa en la tierra de los qbitos es una región cuadrada, pudiendo ésta contener tierra libre o tierra plantada. Si la región está totalmente formada por tierra libre, el mapa se representa por medio de un cuadrado blanco  al tiempo que si está forestada se representa por un cuadrado negro . Si la región está formada por partes de tierra libre y tierra forestada, se la denomina mixta y se divide en 4 sub-regiones, las cuales pueden ser tierra libre, plantada o mixta.

A continuación se muestran algunos ejemplos de mapas:



Región simple 100% tierra libre.



Región simple 100% tierra plantada.



Región mixta, 50% tierra libre (2/4) y 50% tierra plantada (2/4).

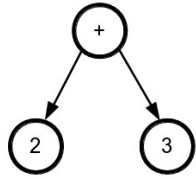
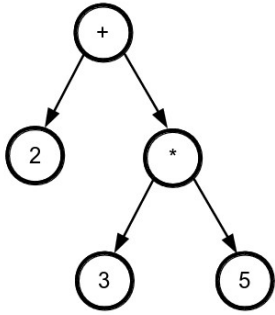
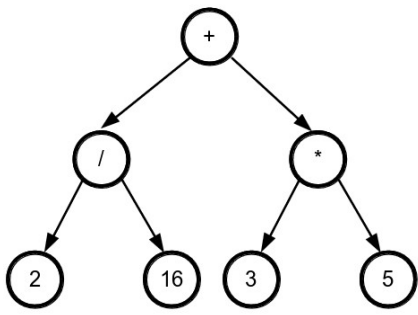


Región mixta, 56,25% tierra libre (9/16) y 43,75% tierra plantada (7/16).

Usted debe realizar un modelo que ayude a los qbitos a representar sus mapas, debiendo éste responder a los mensajes #porcentajeDeTierraLibre y #porcentajeDeTierraPlantada. Realice los tests necesarios para corroborar el correcto funcionamiento de su sistema.

Ejercicio 5

Las operaciones aritméticas pueden representarse por una estructura jerárquica, dónde los nodos intermedios son operadores y los nodos finales los valores. El resultado de una operación representada por ésta estructura está dado por el resultado del operador en la raíz aplicado a los operandos representados por los hijos de la raíz. A su vez, si alguno de estos operandos no es un nodo final (un valor) debe a su vez calcular su resultado antes de que se calcule la expresión que lo involucra.

2 + 3	
2 + 3 * 5	
2 / 16 + 3 * 5	

Implemente un modelo de objetos que permita realizar las cuatro operaciones aritméticas básicas (+, -, * y /). Realice los tests necesarios para verificar el correcto funcionamiento de su modelo.

Ejercicio 6 (Entregar - Fecha tope: 10/11/2015)

El lenguaje HTML está basado en etiquetas demarcadas por los símbolos mayor y menor (ej. `<html>`). A pesar que la mayoría de las etiquetas aparecen de a pares (como por ejemplo los párrafos, `<p>` y `</p>`), existen algunos casos en que sólo se utiliza una etiqueta de apertura (como por ejemplo el tag `` que representa una imagen). En la mayoría de los casos, aquellas etiquetas que poseen apertura y cierre permiten que su contenido sea, a su vez, otra etiqueta (ej. `<p><i> Texto </i></p>`) o una colección de éstas (ej. `<p>Esto se muestra en negritas, esto es texto simple y <i>esto va en cursiva</i></p>`).

Usted debe crear un modelo para representar documentos html en objetos. Dicho documento debe modelar las siguientes etiquetas html:

- Títulos (*Headings*) de 1 a 6. Se representan por medio de los tags “h” y un número que indica la importancia del título. Ejemplos: `<h1>Hola grande</h1>`, `<h6>Hola pequeño</h6>`.
- Párrafos (*Paragraphs*). Se representan por medio del tag “p”. Ejemplo: `<p>Esto es un párrafo.</p>`.
- Texto en negritas (*Bold*). Se representan por medio del tag “b”. Ejemplo: `Esto es un texto en negritas.`.

- Texto en cursiva (*Italic*). Se representan por medio del tag “i”. Ejemplo: <i>Esto es un texto en cursiva.</i>.
- Divisiones de página (*Divs*). Se representan por medio del tag “div”. Ejemplo: <div>Dentro de un div.</div>.

El objetivo final de su modelo es que pueda generar un string con el código html que representa.

Usted debe entregar:

- El modelo de objetos para representar un documento html, permitiendo la anidación de etiquetas html. Dicho documento debe comprender el mensaje #toHtml, que retorne un string con el código html del documento.
- Los tests que verifican el correcto funcionamiento de su modelo.
- Mostrar en un workspace cómo instancia su modelo para representar el siguiente documento:

```
<html>
  <body>
    <h1> El título </h1>
    <p><b><i> Ejemplo de párrafo con mucho énfasis.</i></b></p>
    <p><b>Negritas</b>, texto simple y <i>cursiva</i></p>
    <div><p> Un párrafo dentro de un div. <p></div>
  </body>
</html>
```