

Ejercicio 1

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).

Usted debe crear un modelo sencillo para representar documentos html en objetos. Dicho documento debe modelar los siguientes elementos 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>`.
- 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:

- Realizar un diagrama de clases que documente su solución.
- Implementar los métodos asociados a los mensajes definidos en dicho diagrama.
- Mostrar en un diagrama de instancias los objetos de su modelo para representar el siguiente documento:

```
<html>
<body>
  <h1> El título </h1>
  <p> Ejemplo de párrafo.</p>
  <p> Otro ejemplo de párrafo.</p>
  <div> Un div para el cierre.</div>
</body>
</html>
```

Ejercicio 2

Algunos elementos HTML poseen atributos, que son propiedades que se indican en la apertura del tag. Extienda el modelo del punto anterior para contemplar los siguientes elementos:

- Hipervínculos (*Links*). Se representan por medio del tag “a” e indican la url de destino en el atributo “href”. Ejemplo: `Link de ejemplo`.
- Quiebre de línea (*Line break*). Se representan por medio del tag “br” y no llevan cierre. Ejemplo: `
`
- Imágenes (*Images*). Se representan por medio del tag “img” e indican en el atributo “src” el path de la imagen. No llevan tag de cierre. Ejemplo: ``

Ejercicio 3

Abra una imagen de Pharo y diríjase a la clase `Object`. Investigue los mensajes `#copy`, `#shallowCopy` y `#postCopy`. Explique cómo se aplica el patrón Template Method (cuál es la clase abstracta, qué método es el template method, al menos un ejemplo de clase concreta afectada por el template method, etc).

Ejercicio 4

Usted debe implementar un sistema para el manejo de eventos en un calendario. De los eventos se sabe su título, la hora de comienzo, la hora de finalización y que pueden ser simples o recurrentes. Asimismo el usuario puede modificar la configuración de un evento con el paso del tiempo (por ejemplo, convertir un evento semanal en uno mensual).

Los eventos simples se llevan a cabo en una determinada fecha, al tiempo que los recurrentes tienen un patrón de repetición. Dicho patrón puede ser:

- Semanal. Se repite en un determinado día de todas las semanas (ej. todos los lunes).
- Mensual. Se repite los días X de todos los meses (ej. los días 8 de todos los meses).

Su sistema debe permitir obtener todos los eventos que sucederán en una determinada fecha. Para esto usted debe:

- Realizar un diagrama de clases que documente su solución.
- Implementar los métodos asociados a los mensajes definidos en dicho diagrama.
- Mostrar en un diagrama de instancias un ejemplo de su modelo que represente:
 - Un evento simple, "Cena con mis suegros" a llevarse a cabo el 8/10/2055 de 21:15 a 21:20.
 - Un evento recurrente semanal, "Chess Boxing", todos los lunes de 20:00 a 22:00.
 - Un evento recurrente mensual, "Pagar el alquiler", todos los días 10 del mes, de 8:00 a 8:30.

Ejercicio 5

Para la página web de la dependencia impositiva de la república de Orwell se pide que implemente un calculador impositivo, con el cual un contribuyente pueda calcular los impuestos que debe pagar. El calculador debe recibir el nombre del usuario, su categoría de contribuyente y las facturas. De las facturas se sabe el número, el tipo, la descripción del servicio brindado y el monto cobrado.

Con esta información el usuario le pide al sistema que calcule el impuesto para un rango de las facturas cargadas; dicho rango se expresa por medio de dos números (desde y hasta). Como resultado su modelo debe retornar una instancia de la clase LiquidaciónDePago que comprende los siguientes mensajes:

- #nombreDelContribuyente
- #montoFijoTotal. Se cobra \$3 por cada factura.
- #montoVariableTotal. Monto variable por categoría mas descuentos o recargos según el tipo de factura (ver tablas).
- #montoAPagar. Es el monto fijo total + monto variable total.
- #primeraFacturaConsiderada.
- #ultimaFacturaConsiderada.

Categoría	Boxer	Moses	Napoleón
Monto variable	50% de lo facturado	10% de lo facturado + \$10 por cada factura de mas de \$3000	3% de lo facturado + \$1 por factura

Tipo de Factura	A	B	C
Alteración	-	Recargo de \$100	Descuento del 5%

Usted debe

- Realizar un diagrama de clases que documente su solución.
- Implementar los métodos asociados a los mensajes definidos en dicho diagrama.

Ejercicio 6

Extienda el Ejercicio 5 para que su sistema registre una entrada en un log por cada factura que es agregada al calculador. Usted debe permitir que el log se realice en un archivo, en el Transcript o que simplemente no haya ningún tipo de log.

Ejercicio 7

Usted pertenece a una empresa que está desarrollando una página para rankear bancos de plazas. Los usuarios de dicha plataforma toman fotos de bancos de plazas y las suben al sitio, ganando puntos (*banquines*) que luego pueden canjear por dinero.

Junto con la foto del banco de plaza que un usuario carga debe indicar la ubicación (ej. “Plaza Sarmiento”) y un valor de 1 a 5 indicando el estado del banco de plaza. Por cada nuevo ítem que un usuario carga en el sistema se le acredita un “banquín” en su cuenta.

Por último el sistema le permite al usuario cambiar banquines por dinero, considerando que:

- Se pagan 3 ctvs por banquín.
- El usuario debe tener al menos 500 banquines de saldo para poder hacer una extracción.

La empresa le ha encomendado a usted desarrollar los módulos del sistema que permitan la carga de información así como el manejo del balance contable de los usuarios. En particular se pide que implemente:

- a) La carga de información sobre bancos de plazas.
- b) Registrar en la cuenta de un usuario la acreditación de banquines en base a la información capturada.
- c) Registrar en la cuenta de un usuario la extracción de banquines a cambio de dinero.
- d) Obtener el saldo de una determinada cuenta de usuario.
- e) Saber todos los movimientos de una determinada cuenta entre dos fechas.
- f) Saber cuanta plata cobró un usuario hasta el día de la fecha.

Para eso debe:

- Realizar un diagrama de clases que documente su solución.
- Implementar los mensajes necesarios (puede obviar getters, setters y constructores).
- Mostrar un diagrama de interacción para el punto e).
- Mostrar un diagrama de interacción para el punto f).

Ejercicio 8

Extienda el Ejercicio 4 para agregar a los eventos recurrentes una fecha de caducidad, la cual indica que a partir de ese momento el evento no es válido. Dicha fecha de caducidad es optativa.