


Condiciones de aprobación

Para aprobar es necesario simultáneamente: <ul style="list-style-type: none"> completar el 60% del examen, y obtener al menos la mitad de los puntos en cada paradigma. 	En todas tus respuestas sé puntual , no pierdas el foco de lo que se pregunta. Respuestas en exceso generales son tan malas como respuestas incompletas. 
--	---

Parte A

Contamos con estas dos funciones, que no hacen exactamente lo mismo, pero tienen algunas similitudes:

<pre>f h j = sum . map j . filter h</pre>	<pre>function totalSueldosDeJuniors(empleados) { var sumatoria = 0; for (var empleado of empleados) { if (esJunior(empleado)) { var sueldo = empleado.snd; sumatoria += sueldo; } } return sumatoria; }</pre>
---	---

- Inferir el tipo de la función *f*.
 - Utilizar *f* para:
 - Crear una función equivalente a *totalSueldosDeJuniors* (asumir que contamos con “*esJunior*”).
 - Hacer una consulta usando *f*, para algún otro ejemplo que no tenga nada que ver con empleados.
 - Indicar en cada una de las siguientes afirmaciones si es Verdadera o Falsa, **justificando en todos los casos su respuesta**:
 - La función *f* no es de orden superior, pero usa funciones de orden superior.
 - Se utiliza aplicación parcial en la definición de *f*.
- Y más allá de que tengan objetivos diferentes:
- La definición de *f* es más expresiva que la función *totalSueldosDeJuniors*.
 - La definición de *f* es más declarativa que la función *totalSueldosDeJuniors*.

Parte B

Dados los siguientes predicados:

<pre>% acuerdo(pais_quevende, pais_quecompra, cosa) acuerdo(argentina, china, trigo). acuerdo(argentina, rusia, trigo). acuerdo(eeuu, argentina, carne). acuerdo(china, rusia, arroz).</pre>	<pre>acuerdo(argentina, francia, carne). acuerdo(argentina, brasil, carne). acuerdo(francia, arabia, carne). acuerdo(brasil, zimbabwe, carne).</pre>
--	--

- Queremos reflejar que no hubo acuerdo entre EEUU y China por el comercio de armas. ¿Cómo reflejaría esto en la base de conocimiento? Justifique relacionándolo con algún concepto.
- Resuelva el predicado *recibe*(Pais1, Pais2), que indica si el segundo país puede recibir algo del primero. Por ejemplo, Argentina recibe carne de EEUU por el acuerdo que firmaron. Pero también podría pasar que Brasil reciba de EEUU indirectamente. Contemplar los casos posibles.
- Indicar qué concepto entra en juego en su solución para determinar que hay una relación entre EEUU y Zimbabwe por la carne.
- Se necesita verificar, a partir de los predicados que ya tenemos, si algún país no tiene ningún acuerdo con ningún otro. Si es posible, realizar la consulta y justificar conceptualmente. Si no es posible, indicar por qué.

Parte C

Una farmacia quiere calcular a qué precio vender un medicamento a sus clientes. Para cada medicamento conocemos el precio, y dependiendo de cuál sea la situación de cada cliente, se realizan descuentos sobre el mismo.

Esto se resuelve en base a los siguientes criterios:

- Si el cliente que compra es un jubilado, tiene un descuento de 50% si cobra la jubilación mínima, o un 40% en caso contrario.

- Si el cliente es discapacitado (para lo cual se registró según la ley 22431), no debe pagar nada.

El descuento a aplicar debería ser el mayor en caso de que aplique más de una de estas condiciones (ej. un jubilado discapacitado no debería pagar nada).

Tenemos el siguiente código en Wolok / Smalltalk para el problema indicado:

<pre> class Farmacia { method descuentoCliente(cliente){ if(cliente.esDiscapacitado()){ return 1 } else { if(cliente.esJubilado()){ if(cliente.cobraJubilacionMinima()){ return 0.5 } else { return 0.4 } } else { return 0 } } } method precioDeVenta(medicamento, cliente){ medicamento.descuento(self.descuentoCliente(cliente)) return medicamento.precioFinal() } } class Medicamento { var property precio var descuento method descuento(procentaje){ descuento = porcentaje } method precioFinal() = precio - descuento * precio } class Cliente { var property esDiscapacitado = false var property jubilacionQueCobra = 0 var property jubilacionMinima method esJubilado() = jubilacionQueCobra > 0 method cobraJubilacionMinima() = jubilacionQueCobra == jubilacionMinima } </pre>	<pre> #Farmacia >> descuentoCliente: unCliente unCliente esDiscapacitado ifTrue: [^ 1] ifFalse: [cliente esJubilado ifTrue: [unCliente cobraJubilacionMinima ifTrue: [^ 0.5] ifFalse: [^ 0.4]] ifFalse: [^ 0]] >> precioDeVentaDe: unMedicamento para: unCliente unMedicamento descuento: (self descuentoCliente: unCliente). ^ unMedicamento precioFinal #Medicamento (v.i. precio, descuento) >> descuento: porcentaje descuento := porcentaje >> precioFinal ^ precio - (descuento * precio) #Cliente (v.i. esDiscapacitado, jubilacionQueCobra, jubilacionMinima) >> esJubilado ^ jubilacionQueCobra > 0 >> cobraJubilacionMinima ^jubilacionQueCobra = jubilacionMinima </pre>
--	---

- Indicar verdadero o falso y **justificar** en cada caso con prosa.
 - No hay problemas de responsabilidades, porque la farmacia delega adecuadamente tanto al cliente como al medicamento.
 - Sería posible usar polimorfismo en la solución replanteando cómo fueron modelados los clientes.
 - La forma de resolver precio de venta de un medicamento en la clase Farmacia es correcta, ya que asegura que el medicamento conozca el descuento a aplicar antes de mandarle el mensaje precioFinal.
 - No es correcto que haya un atributo jubilacionMinima en el cliente.
- Implementar una nueva solución (con código y diagrama estático) que resuelva los problemas detectados en el punto anterior.
- Comparar la solución inicial con la desarrollada en el punto anterior analizando cómo se comporta ante este cambio: "Hay que contemplar también que para los clientes que tienen obra social, se considere el descuento que la misma ofrezca". En caso de detectar nuevas mejoras sobre la solución anterior, explicarlas con prosa y/o código.