Paradigmas de Programación **Examen Final** **14/12/2019**

Condiciones de aprobación	
Para aprobar es necesario simultáneamente: • completar bien 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

La fábrica de muebles Armando requiere de su manejo de stock de los productos que realiza en sus distintos locales. Manejan los siguientes productos:

- *Sillones: que tiene un tipo (común, cama, reclinable) y cantidad de módulos*
- *Mesas: forma (rectangular, cuadrada, circular) y material (madera, vidrio)*
- *Sillas: material (metal, madera)*

Se tiene la siguiente base de conocimiento:

```
%stock(sucursal, producto, cantidad)
stock(boedo, sillon(comun, 3), 4).
stock(boedo, silla(madera), 12).
stock(flores, sillon(cama, 2), 1).
stock(flores, silla(metal), 4).
stock(belgrano, sillon(reclinable, 2), 3).
stock(belgrano, silla(madera), 8).
```

Realizar la **codificación** y las **justificaciones** para cada punto:

1. Sabiendo que tenemos los siguientes clientes:
   - Mati, que busca una mesa circular de vidrio y 4 sillas de metal.
   - Leo, que busca un sillón cama de 2 módulos y otro reclinable de 1.

   Agregar la información a la base de conocimientos, sabiendo que se debe poder responder la consulta "¿Qué busca Leo?" (por ejemplo). ¿Hace falta usar listas para representar la información? Si es posible, hacerlo sin usar listas y explicar los conceptos que lo permiten, y en caso contrario hacerlo con listas y explicar por qué son necesarias.

2. Saber si una sucursal trabaja un determinado material. Trabaja el mismo si alguno de sus artículos son de ese material, y se sabe que todos los sillones que trabajan son de madera. ¿Qué concepto resalta en la resolución de este punto y dónde puede verse?

3. Saber si hay una sucursal ideal para un cliente, del cual se conoce su nombre y la información que se agregó en los puntos anteriores. Una sucursal es ideal si tiene en stock todo lo que el cliente busca. ¿Qué concepto aparece que no estaba siendo usado antes?

Parte B

Se tiene la siguiente función:

```
funcion x y lista = (filter (> x) . map (\ f -> f y)) lista
```

1. Explicar lo que hace y proponer mejoras en términos de expresividad. Filtra los valores > a x de haberse transformado

2. ¿Sería posible evaluar la función con una lista infinita de modo que dicha evaluación termine? Justificar conceptualmente y plantear un ejemplo para fundamentar la respuesta.

3. Indicar cuáles de las siguientes expresiones son válidas. Para las que son válidas indicar además el tipo de retorno y para las que no son válidas justificar el motivo.
   a. funcion 3 'hola' []
   b. funcion 3 7
   c. funcion 'chau' 'hola' [length]

14/12/2019

Paradigmas de Programación — **Examen Final**

Parte C

Un taller de confección de ropa de moda nos pide un sistema para calcular el tiempo de confección de sus pedidos. Existen faldas, blusas y shorts, y las prendas se pueden hacer utilizando distintas telas: modal, lycra o denim. Las faldas tienen un tiempo de fabricación de 120 minutos, las blusas 200 minutos y 5' adicionales por cada botón, y para los shorts el tiempo depende de cada uno. Ahora bien, realizar prendas en lycra aumenta el tiempo 20% porque el corte y la costura de telas elásticas es más difícil, y como no hay máquinas para coser denim, se tercériza, por lo que el tiempo con denim aumenta 25 minutos por prenda. Con modal no aumenta el tiempo, es el de cada prenda.

Se tiene el siguiente código:

```
class Pedido {
  var faldas
  var blusas
  var shorts
  method tiempoDeConfeccion(){
    var tiempoTotal = 0
    faldas.forEach({falda =>
      tiempoTotal += falda.tiempo()})
    blusas.forEach({blusa =>
      tiempoTotal += blusa.tiempo()})
    shorts.forEach({short =>
      tiempoTotal += short.tiempo()})
    return tiempoTotal
  }
}

class FaldaModal {
  method tiempo(){
    return 120
  }
}
class BlusaModal {
  var cantBotones
  method tiempo(){
    return 200 + cantBotones * 5
  }
}
class ShortModal {
  var tiempoNecesario
  method tiempo(){
    return tiempo
  }
}

class FaldaLycra {
  method tiempo(){
    return 120 * 1.2
  }
}
class BlusaLycra {
  var cantBotones
  method tiempo(){
    return (200 + cantBotones * 5) * 1.2
  }
}
class ShortLycra {
  var tiempoNecesario
  method tiempo(){
    return tiempoNecesario * 1.2
  }
}
class FaldaDenim {
  method tiempo(){
    return 120 + 25
  }
}
class BlusaDenim {
  var cantBotones
  method tiempo(){
    return 200 + cantBotones * 5 + 25
  }
}
class ShortDenim {
  var tiempoNecesario
  method tiempo(){
    return tiempoNecesario + 25
  }
}
```

1. Responder V o F y justificar conceptualmente en todos los casos.
   a. No está bien aprovechado el polimorfismo entre las prendas. V
   b. Entre las faldas no hay repetición de lógica. V
   c. Usando herencia puedo resolver toda repetición de lógica de esta solución. F
   d. La solución es poco declarativa. V
2. A partir de los problemas descubiertos en el punto anterior, formular una solución superadora que los resuelva, así como también cualquier otro problema detectado. Se pide **código y diagrama estático.**