

Paradigmas de Programación - Examen Final

25 de Junio de 2007

Instrucciones: A la parte 1 le corresponde los ejercicios {1,2,3}, mientras que a la parte 2, los ejercicios {4,5,6}

1. Se puede definir un árbol n-ario de la siguiente manera:

```
<NTree> ::= tree(node:<Int> sons:<List <NTree>>
```

Implementar la función {SumLeaf T} que suma los valores de todas las hojas del árbol n-ario T. Notar que una hoja será un árbol de la forma tree(N nil). Esta función debe ser recursiva a la cola.

2. (2.5 puntos) Escriba una función {SwitchMap Xs F G} que tome una lista Xs y dos funciones unarias F y G. Devuelve una lista donde los elementos en posiciones pares son obtenidos de los elementos de Xs en posiciones pares a los cuales se les aplicó F y a los impares se les aplicó G. Por ejemplo, con las definiciones fun {Inc N} N+1 end fun {Dec N} N-1 end la llamada {SwitchMap [1 1 2 2] Inc Dec} devuelve [2 0 3 1]. La función debe ser recursiva a la cola y no se permite usar ninguna función auxiliar.
3. ° Modifique la semántica de las instrucciones asociadas a procedimientos de manera que la semántica que le resulte implemente "dynamic binding" para las variables libres dentro del cuerpo de un procedimiento.
4. Que ventajas tendría usar colecciones indexadas con estado en lugar de colecciones sin estado en las definiciones de clases? Argumente.
5. Defina el concepto de extensión conservativa para herencia y de dos ejemplos, uno donde la extensión sea conservativa y otro en el que no.
6. Modifique la definición de una clase y de la función New de manera que permita tener variables cuyos valores sean compartidos por todas las instancias de una clase.