

Paradigmas de Programación

Recuperatorio y/o final

28 de Junio de 2006

Instrucciones: Indicar en cada hoja que parte están rindiendo. Primera parte, ejercicios 1 y 2, segunda parte, ejercicios 3, 4 y 5.

Ejercicios:

- (2 puntos) Considerar la función `GCDlist`, tal que `(GCDlist Xs)` computa el *máximo comun divisor* de los números de la lista `Xs`. Por ejemplo: `(GCDlist [12 16 20])` retorna 4.
 - (1 punto) Implementar una función auxiliar `(GCD A B)`, que computa el máximo comun divisor entre dos números. Esta debe ser recursiva a la cola y no debe utilizar los operadores `div` y `mod`.
 - (1 punto) Implementar una versión recursiva a la cola de `(GCDlist Xs)` propiamente dicha.
- (3 puntos) Considerar la función `Eval`, tal que: `(Eval Cs X)` toma una lista `Cs` de coeficientes y evalúa un polinomio en el número flotante `X`. Por ejemplo: `(Eval [3 -1 4] 2)` evalúa a $f(x) = 3x^2 - x + 4$ en el punto 2 y retorna 14.
 - (2 puntos) Dar una versión iterativa de la función `Eval`.
 - (1 punto) Mostrar su corrección utilizando invariantes.
- (1.5 puntos) Según el teórico, la función `From` construye la representación de la clase `C1` heredando de `C2` y `C3`, donde `C1`, `C2`, `C3` son definiciones de clases.

La función `New` crea un instancia de una clase. Esta función es independiente de que la clase haya sido construida con herencia.

Claramente la definición de herencia definida en `From C1 C2 C3` funciona solamente si `C1` hereda de dos clases.

 - (0.75 puntos) Redefina `From` para que tome herencia `Multiple`
 - (0.75 puntos) Redefina la definición de clase y al función `New` para que soporte "static binding".
- (1.5 puntos) El capítulo 3 uso el método de Newton para calcular raíces cuadradas. El método puede ser extendido para calcular raíces de cualquier grado. Por ejemplo, el siguiente método calcula raíces cúbicas: Dado un candidato `g` para la raíz cúbica de `x`, un candidato mejorado es $(x/g^2 + 2g)/3$. Escriba un programa ~~deletario~~ para calcular las raíces cúbicas usando el método de Newton. El programa debe constar de un servidor central donde varios clientes puedan pedir el cálculo simultáneamente.
- (2 puntos) Defina un tipo pila cuya definición sea `Abierta`, `Declarativa` y "bundled".

```
end
  {Wrap c(methods:{Adjoin {Adjoin M2 M3} M1}
  attrs:{Union {Union A2 A3} A1})}
end
```

La función `From` definida arriba construye la representación de la clase `C1` heredando de `C2` y `C3`, donde `C1`, `C2` y `C3` son definiciones de clases.

Claramente la definición de herencia de `From` funciona solamente si `C1` hereda de dos clases.

- a)* Redefina `From` para que tome herencia de tres clases.
- b)* Redefina, si corresponde, la función que crea instancias `New`.
- c)* Modifique la representación de clases para que soporte binding estático de procedimientos.