



Segundo examen opcional

Fundamentos de lenguajes de programación

Duración: 2 horas

Carlos Andres Delgado S, Msc.

`carlos.andres.delgado@correounivalle.edu.co`

06 de Octubre 2020

1. **[10 puntos]** ¿Cual es el mecanismo para implementar asignación en el interpretador? Explique que cambia con respecto a los intepretadores anteriores.
2. **[10 puntos]** ¿Cual es el mecanismo para implementar paso por referencia en los procedimientos? Explique claramente y muestre un ejemplo.
3. **[30 puntos]** Utilizando paso por referencia, considere la siguiente expresión considerando como ambiente inicial vacío:

```
let
  x = 2
  y = 1
  z = 5
  m = proc(s j k)
    begin
      set s = +(j ,k) ;
      set j = -(s ,k) ;
      set k = *(s ,j) ;
      +(s ,+(2 ,-(j ,k)))
    end

  n = proc(a b c f)
    begin
      set a = (f a b c) ;
      set b = (f b c a) ;
      set c = (f c b a) ;
      +(a , -(b ,c ))
    end

in
  begin
    set x = (n x y z m) ;
    set y = (n x y z m) ;
    -(x , +(y , z))
  end
```

Dibuje los ambientes que se generan y muestre claramente cómo cambian los valores a medida que se hacen los llamados de procedimientos.

4. [30 puntos] En el interpretador de chequeo de tipos implemente:
- a) (15 puntos) El tipo de dato string, el cual es un identificador que inicia y termina con comillas. El tipo es **string**.
 - b) (15 puntos) El tipo de dato float, el cual el cual tiene un formato de número punto número, modifique la especificación léxica para aceptarlos y agregue la gramática. El tipo es **float**

Explique claramente dentro del código con comentarios que cambios hizo, agregue ejemplos que permitan evaluar su trabajo y entregue el interpretador modificado como evidencia del cumplimiento de este punto.

5. [20 puntos] Considere la siguiente expresión de objetos:

```
class c1 extends object
  field a
  field b

  method initialize ()
    begin
      set a = 5;
      set b = 7;
    end

  method setup(u,v)
    begin
      set a = +(u,b);
      set b = *(v,a);
      +(a,b)
    end

  method m1(x,y)
    begin
      set a = +(a,x);
      set b = -(b,y);
      send self m2(+(x,b),-(y,a))
    end

  method m2(w,z)
    begin
      set a = -(w,a);
      set b = *(b,z);
      +(w,+(z,a))
    end

class c2 extends c1
  field a
  field b
```

```

field c

method initialize() 0

method setup(u,v)
begin
    set a = -(u,3);
    set b = +(v,3);
    set c = +(u,v);
    super setup(+(a,b),+(b,c))
end

method m1(x,y)
begin
    set a = +(a,b);
    set b = -(b,x);
    set c = -(c,y);
    super m1(x,y)
end

let
    o1 = new c1()
    o2 = new c2()

    in
        let
            x = send o1 setup(1,4)
            y = send o2 setup(3,4)
            i = send o1 m1(4,5)
            j = send o1 m2(3,4)
            k = send o2 m1(1,3)
            in
                +(+(i,j),-(k,x))

```

Dibuje los ambientes generados por los llamados:

```

x = send o1 setup(1,4)
y = send o2 setup(3,4)
i = send o1 m1(4,5)
j = send o1 m2(3,4)
k = send o2 m1(1,3)

```

Además, muestre cómo los valores de los campos de los objetos van cambiando usando una representación simple.

Importante: En este punto es necesario dibujar los ambientes de cada llamado, si sólo dibuja los objetos no se tendrá en cuenta para la evaluación.