



Segundo examen parcial Fundamentos de lenguajes de programación

Duración: 2 horas

Carlos Andres Delgado S, Ing *

05 de Junio 2017

Nombre: _____

Código: _____

1. Conceptos teóricos [30 puntos]

1. (15 puntos) ¿Cual es la diferencia entre los lenguajes de programación **fuertemente** y **débilmente** tipados? De un ejemplo práctico de cada caso.

a) Lenguajes fuertemente tipados: Requieren que los tipos sean definidos y no permiten errores de tipos.

```
int a = 3;
a = "perro" //Es un error
```

b) Lenguajes debilmente tipados: Tienen cierta flexibilidad para los errores de tipos (aunque si existen)

```
a = 3;
b = a + 3
a = "perro"
b = a + 3 //Da error
```

2. (15 puntos) Cual es la diferencia entre encapsulamiento y agrupación en la programación orientada a objetos. Dé un ejemplo en cada caso.

Agrupación se refiere a las clases, a partir de las cuales se instancian los diferentes objetos.

Encapsulación, es la capacidad de los objetos de integrar su estado (atributos) y su comportamiento (métodos) de tal forma se pueda controlar que se puede utilizar y visualizar desde el exterior.

2. Inferencia de tipos [35 puntos]

Para la siguiente expresión:

```
let
  f = proc(? x, ? y)
    if zero?(y) then proc (? k, ? l) >(k,l) else x
  g = proc(?z, ?a, ?b)
    if (a b 9) then a else b
  in let
    h = proc(?c, ?d)
      (d d c c)
  in
    (h f g)
```

Importante: Utilizar el nombramiento de variables indicado en la tabla, será penalizado si no lo hace. Para facilitar el proceso, se recomienda solucionar en el siguiente orden:

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

Expresión	Variable
f	t_f
g	t_g
h	t_h
x	t_x
y	t_y
k	t_k
l	t_l
z	t_z
a	t_a
b	t_b
c	t_c
d	t_d
let ... in (h f g)	t_1
if zero?(y) then proc (? k, ? l) >(k,l) else x	t_2
zero?(y)	t_3
proc (? k, ? l) >(k,l)	t_4
>(k,l)	t_5
if (a b 9) then a else b	t_6
(a b 9)	t_7
(d d c c)	t_8

Tabla 1: Variables de tipo

1. **(10 puntos)** Plantee las ecuaciones de los procedimientos y las evaluaciones que encuentre en el código

$$\begin{aligned}
t_f &= (t_x * t_y) \rightarrow t_2 \\
t_4 &= (t_k * t_l) \rightarrow t_5 \\
t_g &= (t_z * t_a * t_b) \rightarrow t_6 \\
t_h &= (t_c * t_d) \rightarrow t_8
\end{aligned}$$

$$\begin{aligned}
t_a &= (t_b * int) \rightarrow t_7 \\
t_d &= (t_d * t_c * t_c) \rightarrow t_8 \\
t_h &= (t_f * t_g) \rightarrow t_1
\end{aligned}$$

2. **(10 puntos)** Plantee las ecuaciones de las expresiones condicionales y primitivas de la expresión

$$\begin{aligned}
(t_y) \rightarrow t_3 &= (int) \rightarrow bool \\
t_x &= t_4 \\
t_2 &= t_4 = t_x \\
(t_k * t_l) \rightarrow t_5 &= (int * int) \rightarrow bool
\end{aligned}$$

$$\begin{aligned}
t_7 &= bool \\
t_a &= t_b
\end{aligned}$$

3. **(10 puntos)** Resuelva el sistema de ecuaciones a partir de las ecuaciones anteriores, para encontrar el valor los tipos de las variables de la tabla 1

1)	2)
$ \begin{aligned} t_y &= int \\ t_a &= t_b \\ t_k &= int \\ t_3 &= bool \\ t_l &= int \\ t_5 &= bool \\ t_7 &= bool \end{aligned} $	$ \begin{aligned} t_h &= (t_c * t_d) \rightarrow t_8 \\ t_h &= (t_f * t_g) \rightarrow t_1 \\ &\quad \vdots \\ t_f &= t_c \\ t_g &= t_d \end{aligned} $
3)	4)
$ \begin{aligned} t_d &= (t_d * t_c * t_c) \rightarrow t_8 \\ t_g &= (t_z * t_a * t_b) \rightarrow t_6 \\ &\quad \vdots \\ t_z &= t_g = t_d \\ t_a &= t_c = t_f = t_b \\ &\quad t_8 = t_6 \end{aligned} $	$ \begin{aligned} t_a &= (t_b * int) \rightarrow t_7 \\ t_f &= (t_x * t_y) \rightarrow t_2 \\ &\quad \vdots \\ t_a &= t_b = t_f \\ &\quad t_x = t_b \\ &\quad t_y = int \\ t_7 &= t_2 = t_x = t_4 = bool \end{aligned} $

Si observa tenemos la siguiente inconsistencia:

$$\begin{aligned}
&t_4 = bool \\
&t_4 = (int * int) \rightarrow bool
\end{aligned}$$

4. **(5 puntos)** A partir de la resolución del sistema de ecuaciones indique el tipo t_1 de la expresión
Hay error de tipos

3. Objetos [35 puntos]

Para la siguiente expresión:

```
class p1 extends object
  field a
  field b
  field c

  method initialize(f, g, h)
    begin
      set a = f;
      set b = g;
      set c = h;
    end
  end

  method setValor(p,q)
    begin
      set a = +(a,p);
      set b = +(b,q);
      set c = +(c,b);
      send self getValor(a,b)
    end
  end

  method getValor(p,q)
    +(p, *(q,a))
  end
end

class p2 extends p1
  field d
  field e
  method initialize(k, l, m)
    begin
      super initialize(k,l,m);
      set a = super getValor(k,l);
      set b = super getValor(l,m);
      set c = super getValor(m,k);
    end
  end

  method getValor(m,n)
    +(d, *(m,n))
  end
end

class p3 extends p2
  field f
  field g
  method initialize(k, l, m)
    begin
      super initialize(k,l,m);
      set f = super getValor(k,l);
      set f = super getValor(l,m);
    end
  end

  method getValor(m,n)
    +(g, +(m,n))
  end
end

let
  o1 = new p1(4,3,1)
  o2 = new p2(5,4,2)
  o3 = new p3(3,4,2)
in
  let
    a = send o1 setValor(1,4)
    b = send o2 setValor(3,2)
    c = send o3 setValor(4,8)
  in
    +(a,+(b,c))
  end
end
```

Estructuras de los objetos en Dr Racket:

```

;;;;;Simples
;;o1
(#(struct:a-part p1 #(5 7 8)))
;;o2
(#(struct:a-part p2 #(0 0)) #(struct:a-part p1 #(28 56 183)))
;;o3
(#(struct:a-part p3 #(8 0)) #(struct:a-part p2 #(0 0)) #(struct:a-part p1 #(19 42 89)))

;;;;;Planos
;;o1
#(struct:an-object p1 #(5 7 8))
;;o2
#(struct:an-object p2 #(28 56 183 0 0))
;;o3
#(struct:an-object p3 #(19 42 89 0 0 8 0))

```

1. **(15 puntos)** Dibuje la representación de los objetos o1, o2 y o3 después de ejecutar $+(a,+(b,c))$ usando representación simple

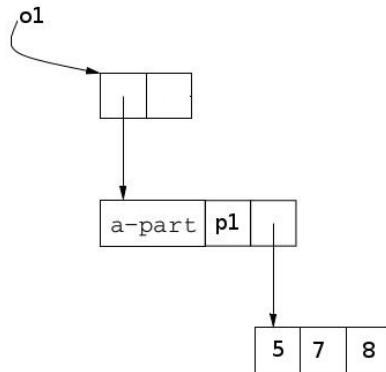


Figura 1: Representación simple de o1

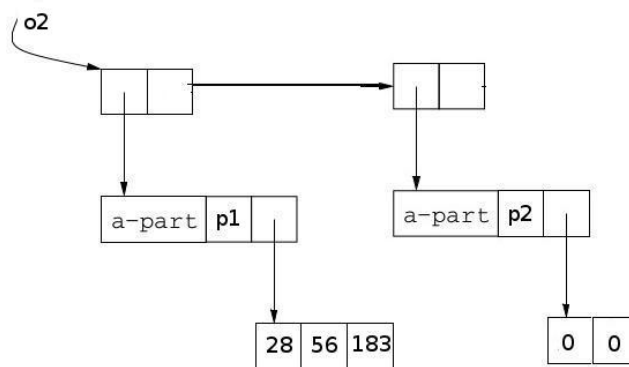


Figura 2: Representación simple de o2

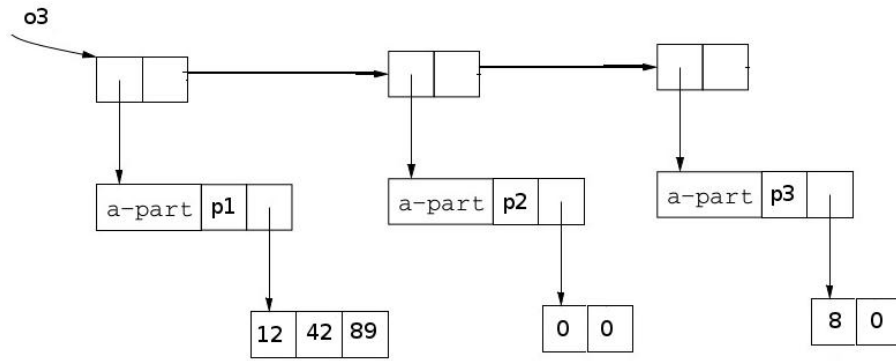


Figura 3: Representación simple de o3

2. **(15 puntos)** Dibuje la representación de los objetos o1, o2 y o3 después de ejecutar $+(a,+(b,c))$ usando representación plana



Figura 4: Representación plana de o1



Figura 5: Representación plana de o2



Figura 6: Representación plana de o3

3. **(5 puntos)** Cuales son los valores de a,b y c, al finalizar la expresión.

- a: 40
- b: 1568
- c: 61

¡Éxitos!