



Segundo examen parcial
Fundamentos de lenguajes de programación
Duración 2 horas

Carlos Andres Delgado S, Msc
`carlos.andres.delgado@correounivalle.edu.co`

13 de Abril de 2021

Importante: Debe explicar el procedimiento realizado en cada uno de los puntos, no se considera válido únicamente mostrar la respuesta.

1. Reglas

- Entregue archivo PDF con la solución del examen.
- No envíe como solución enlaces externos para las capturas o archivos del examen, no se valdrán.
- Las capturas de los puntos deben estar en buena calidad, si alguna no se entiende no se le valdrá el punto en cuestión.
- El examen opcional puede ser realizado en parejas, hacer **un sólo envío por pareja**
- El examen debe ser entregado en el formulario de google especificado por el docente. El opcional va hasta las 4:00pm, de allí se dan 20 minutos de gracia para entregarlo, es decir se recibe sin penalización hasta las 4:20:00pm
- Usted puede entregar después de las 4:20:00pm pero cada 5 minutos de retraso o fracción le descontaré 0.5 en la nota. Por ejemplo si entrega a las 4:31pm tendrá 11 minutos de retraso y 1.5 menos en la nota del examen.

2. Enunciado

1. (35 puntos) Modifique el interpretador de chequeo de tipos para:

a) Agregue el tipo string, la gramática de una string-exp es:

```
<expresion> ::= " <identificador> "  
              ::= string-exp(dato)
```

En la gramática se agrega así:

```
(expresion ("\" identificador "\" ) string-exp)
```

- 1) (10 puntos) Dar representación a string como dato atómico
- 2) (10 puntos) Agregar las variantes requeridas en evaluar-expresion y type-of-expresion, con su respectivo funcionamiento
- 3) (15 puntos) Crear las siguientes primitivas con sus respectivas reglas:
 - Crear la primitiva &, esta recibe dos o más string y retorna un string que es la concatenación
 - Crear la primitiva length, esta recibe un string y retorna un número que es la longitud del string.

Dejar comentarios en las lineas que modifique del interpretador como soporte de su trabajo.

Como se puede ver la expresión está condicionada por la regla del condicional if.

2. (30 puntos) Dibuje los ambientes de la siguiente expresión considerando paso por referencia.

```
let
  x = 1
  y = 2
  z = 3

  m = proc(s, j, k)
    begin
      set s = +(2, s);
      set j = +(s, j);
      set k = *(k, j);
      +(s, j, k)
    end

  n = proc(r, s, v)
    begin
      set r = +(r, 2);
      set s = -(s, r);
      set v = +(v, s);
      +(r, s, v)
    end

  o = proc(a, b, c, f, g)
    begin
      set a = (f a b c);
      set b = (g a b c);
      set c = (f (g a b c) b c);
      +(a, b, c)
    end

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

El valor de la expresión es 1791. En los ambientes debe estar representada el cambio que tienen las variables.

3. Dibuje los ambientes de la siguiente expresión en POO.

```
class c1 extends object
  field x
  field y

  method initialize ()
    begin
      set x=5;
      set y=10;
      +(x,y)
    end

  method m1()
    send self m4(x)

  method m2(n, t)
    *+(t,x) , -(n,y))

  method m4(n)
    send self m2(n,+(n,n))

class c2 extends c1
  field x
  field y

  method initialize(xin, yin)
    begin
      super initialize();
      set x=xin;
      set y=yin;
      0
    end

  method m1()
    super m4(x)

  method m2(z,p)
    begin
      send self m3();
      +(z, +(p,x))
    end

  method m3()
    begin
      set x=add1(x);
```

```

        set y=add1(y)
    end

class c3 extends c2

    field m
    field n
    method initialize(a,b)
        begin
            super initialize(+ (a,2) , -(b,2) );
            set m = a;
            set n = b
        end

    method m1()
        super m4(+ (n,m) )

    method m5()
        send self m1()

let
    o1 = new c1()
    o2 = new c2(2,3)
    o3 = new c3(4,5)
in
    let
        k = send o1 m1()
        f = send o2 m1()
        g = send o3 m5()
    in
        +(k,f,g)

```

El resultado de la expresión es -32.

- a) **(10 puntos)** Dibuje la representación simple (lista de partes) de o1, o2 y o3 después de su instanciación.
- b) **(25 puntos)** Dibuje todos los ambientes que se generan al resolver los llamados
 - $k = \text{send } o1 \text{ m1}()$
 - $f = \text{send } o2 \text{ m1}()$
 - $g = \text{send } o3 \text{ m5}()$

Recuerde que algunos llamados cambian los valores de los campos de los objetos, sea cuidadoso.