



# Segundo examen opcional Fundamentos de lenguajes de programación

Duración: 2 horas  
Carlos Andres Delgado S, Ing \*  
14 de Junio 2017

Nombre: \_\_\_\_\_  
Código: \_\_\_\_\_

## Instrucciones

Para este opcional debe subir un archivo comprimido al campus virtual que contenga:

- El interpretador de chequeo de tipos modificado
- El interpretador de objetos simples modificado
- Un archivo README, explicando que realizó en cada punto y los ejemplos solicitado
- **Importante:** Al final de su examen, debe sustentar al docente lo que realizó. Esta sustentación tiene un factor entre 0 y 1, el cual se multiplicará por su nota.

### 1. Chequeo de tipos [50 puntos]

Utilizando el **interpretador de chequeo de tipos**. Para cada uno de los siguientes ejercicios, en el README incluya:

- Un ejemplo correcto y otro con error de tipos usando condicionales
- Un ejemplo correcto y otro con error de tipos usando procedimientos

1. **(25 puntos)** Agregue a la gramática las definiciones de listas de números, las cuales tienen la siguiente gramática

```
<expresion> := "[" <number>*(,) "]"
```

Un ejemplo funcional es `[1,2,3]`. Debe generar un nuevo tipo conocido como lista de números. Realice los cambios necesarios para que este tipo sea verificado y genere ejemplos que involucren listas.

2. **(25 puntos)** Extienda el interpretador de tal forma se permita primitivas booleanas, estas tienen la siguiente regla de tipos:

$(int * int) \rightarrow bool$

Implemente las primitivas `>`, `<`, `=`, sin modificar las reglas de tipos de las primitivas que se tienen en el interpretador.

### 2. Objetos [50 puntos]

En el interpretador de **objetos simples**.

1. **(25 puntos)** Agregue a los métodos de los objetos los modificadores:

- **public:** Es accesible desde cualquier parte
- **private:** Es sólo accesible desde la misma clase
- **protected:** Es accesible desde la misma clase y clases hijas

Para este punto incluya un ejemplo de cada tipo de acceso (con error y sin error)

2. **(25 puntos)** Permita la sobrecarga. La firma de un método (su identidad) es normalmente su nombre, sin embargo, en sobrecarga su identidad es su nombre más el número de argumentos que recibe, ejemplo:

```
method a() 1  
method a(x,y) +(x,y)
```

La firma del primer método es **a0** y la del segundo es **a2**, supongamos que `o1` es un objeto de la clase, entonces los métodos los invocamos de la siguiente forma:

```
send o2 a() %Retorna 1  
send o2 a(1,2) %Retorna 3
```

Incluya un ejemplo de un objeto con al menos herencia de 2 clases y 3 métodos sobrecargados incluyendo inicializate.

¡Éxitos!

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