



IIC2233 – Programación Avanzada  
Examen

24 de Junio 2015

Duración: 3 horas. Sin consultas.

1. a) (3 pts.) ¿Para qué sirve sobre-escribir el método `__call__` en una clase que hereda de `type`?
- b) (3 pts.) ¿Cuál es la diferencia entre un iterador y un generador?.
- c) (4 pts.) Muestre un ejemplo que ilustre la diferencia entre herencia y composición en modelamiento OOP.
- d) (5 pts.) Usando la función `reduce`, en una línea de código transforme una lista de dígitos  $L$  en el número resultante de “juntar” los dígitos de la lista, ejemplo:  $[2,4,6,1] \rightarrow 2461$ .
- e) (3 pts.) ¿Qué diferencia práctica existe entre realizar llamadas a un servidor usando los métodos GET y POST?
- f) (7 pts.) Implemente un decorador que controle el acceso a la función decorada de tal forma de que no pueda volver a ser ejecutada mientras ya esté siendo ejecutada.
- g) (5 pts.) Dado el siguiente código, escriba lo que debería salir como output:

```
1 def f1(_input):
2     if type(_input) in [int, float, bool, str]:
3         _input = 2
4     else:
5         _input[0] = _input[-1]
6     return True
7
8 def f2(_input):
9     print(_input)
10
11 class A:
12     def __init__(self, a = 0):
13         self.atributo = a
14
15     def __str__(self):
16         return "{}".format(self.atributo)
17
18
19
20
21
22
23
```

```

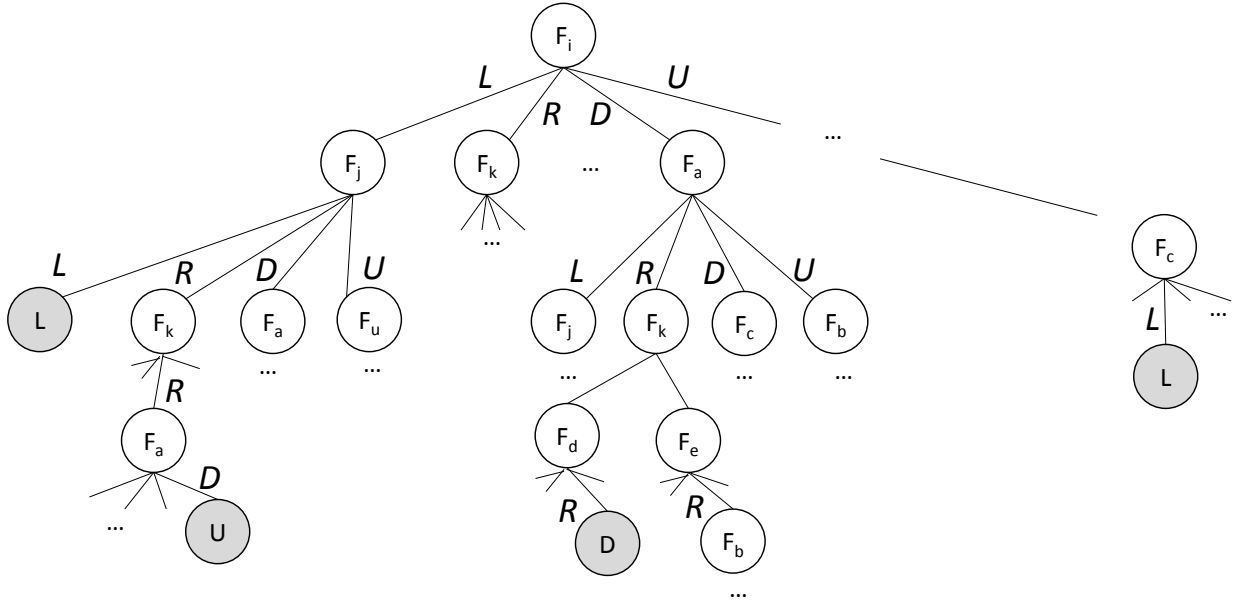
24 if __name__ == '__main__':
25
26     a = 0
27     f1(a)
28     f2(a)
29
30
31     b = [0, 1, 2, 3, 4]
32     f1(b)
33     f2(b)
34
35     c = A()
36     f1(c.atributo)
37     f2(c)
38
39     d = "hola mundo"
40     f1(d)
41     f2(d)

```

2. a) (15 pts.) Las estrellas cuyo brillo varía a lo largo del tiempo son de gran interés para la comunidad Astronómica. Una estrella pertenece a una de las posibles clases de estrellas variables (RRLyrae, Eclipsing Binaries, Mira, Cepheids, etc.). Cada estrella tiene una posición en el espacio medida en dos coordenadas (RA y DEC), un id que la identifica únicamente en un catálogo y un número (variable) de observaciones. Cada observación está compuesta por tres números: el instante de tiempo en que la observación fue realizada, la magnitud del brillo que se observó y un rango de error correspondiente a la medición. El cielo está compuesto de varios *fields*, y cada *field* contiene muchas estrellas. Se requiere de un programa que estructure toda esta información usando modelamiento orientado a objeto. Además se requiere conocer para cada estrella el promedio y varianza de la magnitud de su brillo. Usted debe implementar las clases necesarias, con los respectivos datos y comportamientos. Haga los supuestos que considere necesarios.
  
- b) (15 pts.) Supongamos que existe un algoritmo vigilante que controla el acceso al catálogo que contiene la base de datos de estrellas. El vigilante permite acceder simultáneamente sólo a  $C$  procesos para que puedan leer datos desde el catálogo, y al resto los deja en espera, es decir, el siguiente proceso puede acceder una vez que cualquiera de los  $C$  procesos que estaba leyendo termina. Implemente en Python el algoritmo del vigilante usando sólo un Lock de la clase thread.

3. a) (20 pts) Una empresa ha generado un algoritmo de inteligencia para ayudar a Poc-Mon a escapar de los  $F$  fantasmas cada vez que éstos se acercan. El algoritmo está representado como un árbol, donde cada nodo corresponde a un fantasma y cada arista que sale de cada nodo corresponde a la dirección hacia donde se está moviendo el fantasma (son 4 posibles direcciones: U, D, L, R).

La siguiente figura muestra un ejemplo de árbol que representa este tipo de inteligencia. En primer lugar se revisa hacia dónde se mueve el fantasma  $F_i$ , dependiendo de esto revisa un siguiente fantasma y así hasta llegar a una hoja que le indica a Poc-Mon hacia donde ir dado el movimiento actual de los fantasmas.



Asuma que los fantasmas siempre quieren acercarse a usted (el objetivo de ellos es atrapar a Poc-Mon). La empresa le pide a usted desarrollar un algoritmo que dado un árbol que representa la inteligencia de Poc-Mon, determine cuál es el fantasma más temido por Poc-Mon (el fantasma del cual Poc-Mon más trata de escapar). Debe desarrollar usted mismo una forma de cuantificar la cobardía de Poc-Mon respecto a un fantasma dado.

- b) (20 pts.) La empresa ha puesto a prueba su propio árbol de inteligencia y a construido una base de datos con  $N$  situaciones en que Poc-Mon se enfrentó a un conjunto de  $F$  fantasmas. La base de datos es una tabla con  $N$  filas y  $F + 2$  columnas, donde cada fila corresponde a una situación. El elemento  $x_{i,j}$  ( $i \in [0, \dots, N - 1]$ ;  $j \in [0, \dots, F + 1]$ ) de la tabla contiene la dirección en que se movió el fantasma  $j$  en la situación  $i$ , el elemento  $x_{i,F}$  contiene la dirección hacia donde se movió Poc-Mon en la situación  $i$  y el elemento  $x_{i,F+1}$  contiene una  $M$  o una  $E$  dependiendo de si Poc-Mon murió o escapó respectivamente en la situación  $i$ .

La empresa le pide a usted que desarrolle un algoritmo donde dado un árbol de inteligencia y su respectiva base de datos de situaciones de prueba determine: la probabilidad de sobrevivencia asociada a cada hoja del árbol de inteligencia. Considere que cada situación puede estar repetida muchas veces ( $N$  es muy grande), y no siempre el resultado de Poc-Mon fue el mismo al tratar de escapar bajo una misma situación.

**Nombre:**

**Sección:**

**Nombre:**

**Sección:**

**Nombre:**

**Sección:**

**Nombre:**

**Sección:**

**Nombre:**

**Sección:**