



PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE
ESCUELA DE INGENIERÍA
DEPARTAMENTO DE CIENCIA DE LA COMPUTACIÓN

IIC2233 - Programación Avanzada
2° semestre 2015

Actividad 10

Metaclasses II

Instrucciones

En esta actividad usted deberá crear dos metaclasses que le permitan manejar comportamientos muy específicos. En primer lugar se busca crear una metaclassa que permita restringir la escritura de atributos luego de la instanciación de un objeto. En segundo lugar se busca crear una metaclassa que le permite simular *Singletons*.

Requerimientos

1. RestrictedAccess

Deberá crear la metaclassa `RestrictedAccess` de tal forma que cumpla el siguiente comportamiento:

```
class Person(metaclass=RestrictedAccess):
    attributes = ["name", "lastname", "alias"]

p = Person("Bruce", "Wayne", "Batman")

print(p.name, p.lastname, "es", p.alias, "!")
## Bruce Wayne es Batman !

p.alias = "Robin"
## AttributeError: cant set attribute
```

Es decir, la clase se debe poder instanciar dada una lista de atributos `attributes`. Todos estos atributos se podrán acceder sin problema alguno, sin embargo *NO* se podrán modificar directamente una vez instanciada la clase. A parte de esto, la lista `attributes` deberá ser eliminada de la clase lo antes posible.

2. Singleton

Un singleton es una clase que solo se puede instanciar una única vez. De ser instanciada más veces, no lanzará error, pero se retornará el primer objeto que se creó. Lo más importante es que *nunca* se cree otra instancia de la clase. Esto implica que todo chequeo deberá ser previo a su instanciación. Esta es la razón por la cual usted creará la metaclassa `Singleton` para manejar este comportamiento. Observe un ejemplo:

```

class A(metaclass=Singleton):
    def __init__(self, value):
        self.val = value

a = A(10)    # Se crea una instancia de A
b = A(20)    # Se retorna la instancia que ya estaba creada

print(a.val, b.val)
## 10 10

print(a is b)
## True

```

To - DO

- (4.00 pts) Programar metaclasses `RestrictedAccess`
- (2.00 pts) Programar metaclasses `Singleton`

Tips

- Recuerde el comportamiento y las funcionalidades de las `properties`, le pueden ser útiles.
- Las funciones `getattr` y `setattr` le permiten obtener y setear atributos en un objeto de la siguiente forma:

```

a = obj.attr    <- es igual a ->    a = getattr(obj, "attr")

obj.attr = 10   <- es igual a ->    setattr(obj, "attr", 10)

```

Estos le podrían servir para setear los atributos en la metaclasses `RestrictedAccess` a partir de la lista `attributes`.