



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

IIC2233 - Programación Avanzada  
1<sup>er</sup> semestre 2016

# Actividad 6

## Decoradores

### Instrucciones

A pesar de su experiencia trabajando en MercadoPreso para Bucchi Incorporated, usted sigue como empleado de dicha empresa. Como habrá notado, el ambiente es medio rígido. En esta ocasión, debe coordinar el trabajo de dos colegas. El primero le entregó las clases y funciones que van a usar en el proyecto. El segundo se encargó de hacer la ejecución. Como usted se maneja a la perfección con las buenas prácticas de PEP8, nota que el código que le pasaron está bien, pero no es perfecto. Como los programadores que lo hicieron son mañosos, no toleran ningún cambio a alguna línea que hayan escrito, por lo que usted deberá corregir el programa sin modificar las clases o las funciones.

### Requerimientos

- El programa completo debe correr sin errores.
- Programar y aplicar un decorador que:
  - verifique si los métodos dados, de la clase `Producto`, existen y, en caso de que estén en `CamelCase` transformelos a `snake_case` de forma que cumplan con PEP8.
  - convierta métodos específicos de la clase `Producto` en métodos privados.
  - corrija el IVA en el precio final del producto, para que se aplique un 19 %, en vez del que especificó el otro programador.
  - evite que al incrementar el precio del producto se entregue un valor que no sea del tipo `int`.

### Notas

- No se puede modificar ninguna línea ya existente en el archivo `main.py` entregado. Solo puedes agregar nuevas líneas.
- Puedes probar en la ejecución del programa si el método privado efectivamente lo es.

### To - DO

- (1.50 pts) Crear el decorador `snake_case` que reciba una cantidad no determinada de métodos, verifique si estos existen y, si estos existen, detecte si están en `CamelCase` y transformelos a `snake_case`.

- (1.50 pts) Crear el decorador `protect_methods` que reciba una cantidad no determinada de métodos y que haga privados los métodos de la clase que debiesen serlo.
- (1.00 pts) Crear el decorador `correct_price` que reciba el porcentaje de IVA dado en el programa y el porcentaje correcto a utilizar y que corrija el calculo del precio de un producto.
- (1.00 pts) Crear el decorador `check_input_is` que reciba una cantidad no determinada de tipos y que verifique que los tipos de los argumentos que recibe un método sean del tipo esperado.
- (1.00 pts) Decorar la clase y los métodos con los decoradores y argumentos que correspondan.

## Tips

- Recuerda que un decorador es una función, por lo que puede recibir argumentos.
- Recuerda que las clases también pueden ser decoradas.
- Para extraer, escribir o borrar algún método de una clase puedes usar las siguientes funciones:
  - `getattr(cls, "nombre_metodo")`
  - `setattr(cls, "nombre_metodo", metodo)`
  - `delattr(cls, "nombre_metodo")`
- Para verificar el tipo de dato de algún objeto puedes usar `isinstance(objeto, tipo_esperado)`. Por ejemplo, si definimos `a = [5, 6, 7]` al usar `isinstance(a, list)` obtendremos el valor `True`.

## Entrega

- **Lugar:** GIT - Carpeta: Actividades/AC06
- **Hora:** 16:55