



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

IIC2233 - Programación Avanzada
1^{er} semestre 2017

Actividad 08

Metaclases

Instrucciones

El malvado Dr. Mavrakis ha atacado otra vez. Después de sus infructuosos intentos de obtener todo el dinero del universo y haber tenido una pésima experiencia con una pizza de piña, este villano ha conseguido acceso al código del universo (¡sí, el universo está escrito en Python!) Inspirado en una escena¹ de su serie favorita, ha creado una version mejorada de su virus, el MavraVirus2.0, con lo que ha hecho desastres en el universo, creando así uno donde las pizzas están hechas de bebida y comen personas! Luego de esto, descubre que posees una habilidad llamada el "[Reading Steiner](#)", por lo que recuerdas la jerarquía del universo antes del ataque del Dr. Mavrakis. En tu intento por salvar al universo te topas con uno de tus antiguos clientes, la pizzería Daddy Juan's. Sabiendo la debilidad del Dr. Mavrakis ante las pizzas, decides que es una buena idea comenzar por restaurar el orden en esta empresa, con el fin de poder utilizarla contra los planes de este villano (y recibir mucha pizza gratis también).

El caos generado por el virus ha causado que tu trabajo, que costó tanto esfuerzo, ahora se rija por las nuevas reglas del universo. Junto al enunciado se encuentra el código que usaba Daddy Juan's para mantener su negocio. Este archivo se llama `ACorrupta.py` y está dañado. Esta nueva habilidad también implica que debes ser muy cuidadoso a la hora de cambiar cosas, ya que si la gente se entera que recuerdas la realidad anterior, puedes provocar una paradoja y acabar con el universo entero. Por esto, debes encontrar los errores y corregirlos sin tocar el código original. Se destinará un espacio para que puedas escribir tu solución.

El comportamiento esperado de tu programa es el siguiente:

■ Clase `Chef`

- Debe verificar si hereda de la clase `Person`
- Debe poseer el método `cook()`, que debe retornar una instancia de `Plate` con una comida y una bebida. Pueden hacer uso de los métodos `choose_food()` y `choose_drink()` que sobrevivieron al ataque del virus.
- Debe guardar el nombre del restaurant para el que trabaja. Queda prohibido guardar la instancia del restaurante dentro de la clase `Chef`.

■ Clase `Client`

¹<https://www.youtube.com/watch?v=28JerOPNPII>

- Debe verificar si hereda de la clase `Person`
- Debe poseer el método `eat()`, que reciba una instancia de `Plate` y rechace el plato si no está compuesto correctamente. El modo de rechazo debe ser un mensaje por consola que diga la razón del rechazo (i.e. “Mi plato tiene dos bebidas”, “Mi plato tiene dos comidas”).
- Si el plato es correcto (1 comida 1 bebida) entonces se imprime un mensaje que diga “Que delicia!” si los puntos de calidad son sobre 50 y “Esto no es digno de mi paladar” en otro caso.

■ Clase **Restaurant**:

- No puede ser instanciada sin que se entregue una lista con al menos un Chef válido como parámetro (cuándo un chef es válido se explica abajo).
- Un Chef no puede trabajar en dos restaurantes. Esto significa que si a un nuevo restaurante se le asigna un Chef que ya pertenece a otro restaurante, el Chef abandona el viejo y comienza a trabajar en el nuevo. Si un restaurant tiene un solo Chef este no podrá dejar su trabajo, o sea no es válido para la instanciación.
- Si la instanciación es exitosa se debe mostrar un mensaje que lo evidencie, junto a nombrar los Chefs recién contratados.
- Deben implementar el método `llega_cliente()` que acepte solo instancias de la clase `Cliente` y las agregue a la lista de clientes del restaurant. También debe existir el método `cliente_se_va()`, que acepta el nombre de un cliente y lo retira de la lista si este está.
- El restaurant posee el método `start()`, pero deben asegurarse de que tenga clientes, si no este no se ejecuta y en su lugar deben imprimir “{Nombre Restaurant} no tiene clientes, que pena”

Requerimientos

- (2.00 pts) Clases `MetaPerson`
- (4.00 pts) Clase `MetaRestaurant`.

Notas

- Recuerden leer el código entregado para que entiendan el modelamiento :)
- **Importante:** No pueden sobrecribir sobre las cosas ya hechas en el archivo que se les entrega, sólo pueden hacerlo en la metaclass `MetaPerson` o en `MetaRestaurant` según corresponda.

Entrega

- **Lugar:** GIT - Carpeta: Actividades/AC08
- **Hora:** 16:55