



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

IIC2233 - Programación Avanzada
1° semestre 2016

Actividad 03

Estructuras de Datos y Properties

Instrucciones

El supermercado *SuperSuperPrecios* necesita simular su funcionamiento para tomar decisiones sobre los turnos, los precios de sus productos y el servicio al cliente. El grupo de desarrollo *IIC2233*, al cual usted pertenece, va a realizar la simulación. Como usted es el “novato” dentro del grupo, sólo lo han dejado realizar ciertas partes del programa. Estas se detallan a continuación:

- **ProductStack:** esta estructura se utilizará para simular los productos en las góndolas. Los métodos a implementar son:
 - `stack_product(product)`: agrega un producto a la pila
 - `pop_product()`: retorna el último producto agregado a la pila
- **PayingQueue:** esta estructura se ocupará para simular la cola de la caja. Los métodos a implementar son:
 - `arrive(shopping_cart)` agrega el carro de compras de un cliente a la cola de la caja
 - `leave()`: saca de la cola al carro de compras del cliente que está primero.
- **ShoppingCart:** almacena las compras realizadas por una persona. Tiene una lista de productos comprados y debe implementar los siguientes métodos:
 - `amount_per_product()`: retorna la cantidad comprada por cada tipo de producto
 - `types_of_product()`: retorna una lista con los tipos de productos comprados
 - `add_product(product)`: agrega un producto al carro

Usted notó que el equipo de desarrollo no tomó en cuenta las colas preferenciales. Estas colas pueden tener carros de compra de prioridad normal (prioridad = 1) y carros de compra de mayor prioridad (> 1). El funcionamiento de esta cola es el siguiente: a medida que van llegando carros, estos se ponen a la cola siguiendo el orden dado por las prioridades, como en la siguiente imagen:



Figura 1: `arrive(shopping_cart, priority)` Si hay dos carros con prioridades iguales, queda más cerca de la caja el que llegó primero a la fila

Para sacar carros de la cola, se toma el que está más cerca de la caja. Como en la imagen.

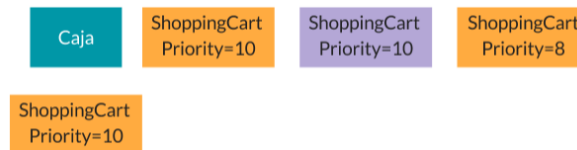


Figura 2: `leave()`

Implemente los siguientes métodos:

- `arrive(shopping_cart, priority)`: ingresa un carro de compra en la estructura de datos, de acuerdo a la prioridad. Recuerde que el de mayor prioridad debe ir primero. Esta cola **siempre** debe estar ordenada.
- `leave()`: saca de la cola al carro de compra que está primero

Además de los requerimientos ya mencionados, le han pedido que se pueda obtener el número de items en cada una de las estructuras haciendo `instancia.length`. También le han dicho que la estructura sólo debe ser modificada con los métodos mencionados arriba, es decir, si la estructura está guardada en la variable `structure` y se realiza lo siguiente: `instance.structure = new_structure`, la estructura no cambia. Si esto llega realizarse debe informar al usuario, por consola, que esto no es posible.

Requerimientos

- Se evaluará que utilice las estructuras de datos de Python que están en el material de clases. Debe elegir la estructura apropiada (lista, diccionario, set,...) de acuerdo a su criterio.

Notas

- NO es necesario realizar la clase producto. Por simplicidad, represéntelos como strings o un número.

To - DO

- (1.10 pts) Clase ProductStack
 - (0.50 pts) `stack_product`
 - (0.50 pts) `pop_product`

- (0.10 pts) obtener largo y restringir modificación
- (1.10 pts) Clase PayingQueue
 - (0.50 pts) `arrive`
 - (0.50 pts) `leave`
 - (0.10 pts) obtener largo y restringir modificación
- (1.10 pts) Clase ShoppingCart
 - (0.50 pts) `amount_per_product`
 - (0.50 pts) `types_of_product`
 - (0.10 pts) obtener largo y restringir modificación
- (1.10 pts) Clase SpecialPayingQueue
 - (0.50 pts) `arrive`
 - (0.50 pts) `leave`
 - (0.10 pts) obtener largo y restringir modificación
- (0.70 pts)
 - Crear un ProductStack con 5 productos
 - Crear una PayingQueue con 5 carros
 - Crear una SpecialPayingQueue con 5 carros
- (0.90 pts): sobre-escriba los métodos necesarios para poder imprimir las estructuras de la forma `print(instancia)`. El output de la consola debe ser claro y entendible.

Entrega

- **Lugar:** GIT - Carpeta: Actividades/AC03
- **Hora:** 16:55