

UNLP - Programación Orientada a Objetos 1 2018 - Redictado de 2017

Examen Parcial - Segunda Fecha - 19 de junio de 2018

En el sitio de compras de Amazon existen diferentes módulos que permiten al cliente comprar un producto. Los clientes pueden estar vinculados a un programa de beneficios. Existen diferentes programas de beneficios. Un cliente puede pertenecer a 0 o un único programa. De un cliente se conoce el nombre y la dirección. Actualmente existen dos programas de beneficios: Prime y Premium.

Se necesita implementar el SistemaDeVentas, que “usando” otras partes del sistema, ejecuta el proceso completo de ventas. En Amazon ya existen y deben usarse (NO IMPLEMENTAR) los siguientes objetos:

```
class SistemaDeEnvios
>>cotizarEnvio: unProducto
"Retorna un número indicando el costo del envío para unProducto"
>>despacharEnvioNormal: unaOrden
"Envía el producto de unaOrden al cliente correspondiente usando el mecanismo de envío normal"
>>despacharEnvioUrgente: unaOrden
"Envía el producto de unaOrden al cliente correspondiente usando el mecanismo de envío urgente"

class SistemaDeStock
#hayStock: unProducto
"Retorna true si hay stock disponible para unProducto"
#decrementarStock: unProducto
"Reduce una unidad del stock disponible de unProducto"

class SistemaDeCobros
#cobrar: unaOrden
"Ejecuta el cobro al cliente del producto de unaOrden."
```

Se debe implementar la funcionalidad para realizar la venta en un SistemaDeVentas considerando lo siguiente:

- Los productos se venden de a uno. De un producto se conoce el nombre y el precio. El sistema de ventas debe generar una Orden que especifica el producto comprado, el cliente y el costo total. El costo total de una Orden es la suma del valor del producto con el costo de envío. El costo de envío se obtiene invocando SistemaEnvios#cotizarEnvio:. Los clientes con programa **Prime** tienen bonificado el 50% del envío, y los clientes con programa **Premium** directamente no pagan envío.
- El sistema de ventas generará una Orden únicamente si hay stock suficiente del producto. Para ello debe invocar SistemaStock#hayStock:. Si hay stock suficiente, se genera la Orden y el sistema debe notificar al sistema de stock que disminuya el stock (SistemaStock#decrementarStock:)
- Una vez creada la Orden el sistema debe cobrar la orden invocando SistemaCobros#cobrar:
- Finalmente, el sistema despacha la orden para ser enviada. En el caso de ser un cliente vinculado al programa Prime, se despacha por envío urgente (SistemaEnvios#despacharEnvioUrgente:). Para un cliente sin programa de beneficios y para los clientes Premium se despacha por envío normal (SistemaEnvios#despacharEnvioNormal:).

Tareas:

- 1) Diagrama de clases UML
- 2) Implemente completamente (todo excepto las clases y métodos de los sistemas mencionados arriba):

```
#vender: unProducto a: unCliente
"Si hay stock suficiente, crea una Orden, la cobra y envía de acuerdo a TODOS los puntos mencionados anteriormente, y almacena la orden"

#totalVentasSinBeneficios
"Retorna la suma de todos los costos totales de las ordenes hechas a clientes sin programa de beneficios"

#listadoDeClientesSinBeneficios
"Retorna una lista sin repetición de los clientes que no están vinculados a ningún programa, ordenados por nombre de manera ascendente"
```

4) Muestre en el workspace como:

- a) Instancia el SistemaDeVentas (existen objetos ya creados de SistemaDeEnvios, SistemaDeStock y SistemaCobros)
- b) Instancia un cliente sin ningún tipo de programa de beneficios y realice una compra para ese cliente de un producto X.
- c) Muestre cómo haría ahora para asociarlo al programa Prime y realice una segunda compra con ese cliente