2º Parcial - Turno Mañana - 2023 TEMA 1

Apellido y Nombres: Enriquez Arturo

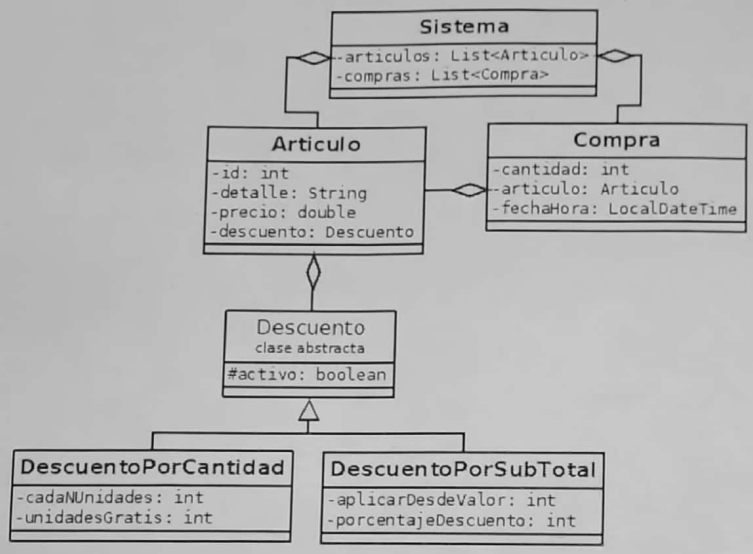
DNI: 18.511.610

Firma:

UNIVERSIDAD NACIONAL DE LANÚS UNLa.

Licenciatura en Sistemas
Orientación a Objetos 1

Proyecto: Crear una carpeta /tuApellidoNombre/tuNroDni (el nombre del proyecto es tu DNI)
Entrega: Aula Virtual subir la carpeta comprimida con tuApellidoNombre



Notas:

- **El examen acredita puntos por la resolución del modelo y test de cada CU.**
- El marco teórico del modelo del parcial está reducido para que se pueda llevar adelante en el examen. No se contemplaron varios casos de uso de la realidad. Solo hay que hacer los que se piden a continuación.

Casos de uso:

1) **+ traerArticulo(String detalle): Articulo**

   //En caso de que el artículo no exista en la lista devolver null.

2) **+ agregarArticulo(String detalle, double precio, Descuento descuento): boolean**

   //Se debe calcular el id de forma autoincremental, teniendo en cuenta que la lista puede tener altas y bajas.

   //Lanzar excepción en caso de que ya existe un artículo con el detalle que entra por parámetro.

3) **+ agregarCompra(int cantidad, Articulo articulo, LocalDateTime fechaHora): boolean**

   //Lanzar excepción caso de que la cantidad de la compra sea igual o menor a 0.

4) **# abstract calcularDescuentoParticular(int cantidad, double precio): double**

   //Para la clase **DescuentoPorCantidad**:

   //Por cada 'cadaNUnidades' se regala N 'unidadesGratis'. Por ejemplo, si cadaNUnidades=3 y unidadesGratis=2. Si la cantidad a comprar es igual a 7, se van a regalar 4 unidades. Por lo que el descuento sería 4 (de las unidades a regalar) * precio del artículo.

   //Ayuda: Para sacar la parte entera de una división se puede hacer el siguiente cálculo

   // int parteEntera = valor / divisor;

   //Para la clase **DescuentoPorSubTotal**:

   //Si el total a pagar es mayor o igual a 'aplicarDesdeValor' se calcula el descuento del atributo 'porcentajeDescuento'.

5) **+ aplicarDescuento(int cantidad, double precio): double**

   //En caso de que el descuento esté activo (activo sea igual a true) se aplica el descuento particular. En caso contrario, el descuento será igual a 0.

6) **+ precioTotal(): double**
   //Devuelve el precio total de la compra incluyendo el cálculo del descuento.
7) **+ traerCompras(LocalDateTime desde, LocalDateTime hasta): List<Compra>**
   //Los intervalos son cerrados, se incluyen los extremos.
8) **+ traerComprasDescuentoPorCantidad(LocalDateTime desde, LocalDateTime hasta): List<Compra>**
   //Traer todas las compras, entre fechas, que tengan un artículo con clase de descuento igual a 'DescuentoPorCantidad'.
   //Los intervalos son cerrados, se incluyen los extremos.
9) **+ traerArticulosConDescuento(boolean activo): List<Articulo>**

TestOO1Tema1.java

Test 1:
Agregar los siguientes artículos al sistema y luego imprimir la lista.
Nota: Para cargar los descuentos se puede hacer el new de la sub clase que corresponda en el test directamente.
Articulo [id=1, detalle=art1, precio=100.0, descuento=DescuentoPorCantidad [activo=true, cadaNUnidades=3, unidadesGratis=1]]
Articulo [id=2, detalle=art2, precio=150.0, descuento=DescuentoPorCantidad [activo=true, cadaNUnidades=5, unidadesGratis=2]]
Articulo [id=3, detalle=art3, precio=200.0, descuento=DescuentoPorSubTotal [activo=false, aplicarDesdeValor=500, porcentajeDescuento=10]]
Articulo [id=4, detalle=art4, precio=500.0, descuento=DescuentoPorSubTotal [activo=true, aplicarDesdeValor=1000, porcentajeDescuento=50]]

Test 2:
Tratar de agregar el con nombre 'art1' nuevamente.

Test 3:
Agregar las siguientes compras al sistema y luego imprimir la lista.
Compra [cantidad=6, articulo.detalle=art1, fechaHora=2023-10-20T09:00]
Compra [cantidad=5, articulo.detalle=art2, fechaHora=2023-10-20T09:30]
Compra [cantidad=10, articulo.detalle=art2, fechaHora=2023-10-20T10:00]
Compra [cantidad=3, articulo.detalle=art3, fechaHora=2023-10-20T10:30]
Compra [cantidad=4, articulo.detalle=art4, fechaHora=2023-10-20T11:00]

Test 4:
Tratar de agregar la compra del artículo con nombre 'art1' nuevamente, pero, con cantidad=0.

Test 5:
Calcular el precio total de las compras con índice 2, 3 y 4 de la lista de compras.
Para ello, se puede hacer desde el test lo siguiente: '**sistema.getCompras().get(2).precioTotal()**'.
Lo normal sería pedir al sistema la compra por alguna clave, pero se prioriza el desarrollo de los métodos de herencia.
Ejemplo del formato de salida:
Total compra índice 2: **Valor**.
Total compra índice 3: **Valor**.
Total compra índice 4: **Valor**.

Test 6:
Traer todas las compras entre las fechas desde=2023-10-20T09:15 y hasta=2023-10-20T10:30.
Nota: Los intervalos son cerrados, se incluyen los extremos.

Test 7:
Traer todas las compras que tengan un artículo con clase de descuento igual a 'DescuentoPorCantidad' entre las fechas desde=2023-10-20T09:15 y hasta=2023-10-20T10:30.
Nota: Los intervalos son cerrados, se incluyen los extremos.

Test 8:
Traer todos los artículos con descuentos activos.