Repaso Relaciones Entre Clases – Contenedores - Herencia

Modelar y codificar un software con las siguientes características.

1. Una empresa comercializa dos tipos de productos: 1) Los que fabrica ella misma, 2) Los que compra a otra empresa para revender.

Los productos que fabrica requieren una **secuencia de procesos de n pasos** para la fabricación.

Los productos que solo comercializa, cuentan con un precio de transporte hacia la empresa que nos cobra el proveedor.

1. Se debe crear una clase ***Proceso*** que encapsule todos los datos necesarios para cada uno de los n pasos del proceso de fabricación para el primer tipo de productos.
2. Se cuenta con un *Almacen* para contener todos los productos. El *Almacen* se modela como un *ArrayList* que contiene objetos ***Producto*** (de ambos tipos).
3. El *Almacen* debe tener un **contador** que vaya contabilizando cada *Producto Fabricado* (del primer tipo) que se incorpora. El contador **NO** debe contabilizar los productos adquiridos para revender (del segundo tipo).
4. Implementar todos los métodos faltantes y necesarios para obtener costos de fabricación de los Productos (de ambos tipos).
5. Cada objeto *Proceso* debe contener un método ***getTodosCostos()****,* que devuelva la suma del costo propio del *Proceso*, mas todos los costos que se encuentran en los Procesos que siguen al mismo en la secuencia. Ejemplo: si el producto cuenta con 4 procesos de fabricación, al activar el método ***getTodosCostos()*** al primer proceso, debe retornar la sumatoria de los costos de los 4 procesos.

**Resolución**:

Un posible diseño para el sistema sería:



Algunas cosas para tener en cuenta a partir del diseño anterior:

1. De Producto no se pueden crear objetos porque es abstracta.
2. Un ProductoFabricado puede estar relacionado (o no) con solo un Proceso.
3. Un Proceso esta relacionado con el “siguiente” Proceso.
4. Un Almacen puede tener 0 o más Productos.
5. Un Almacen es una especialización (herencia) de ArrayList. La especialización consiste en agregar un contador de objetos, y sobrescribir el método *add()*.