

**EXAMEN FINAL**

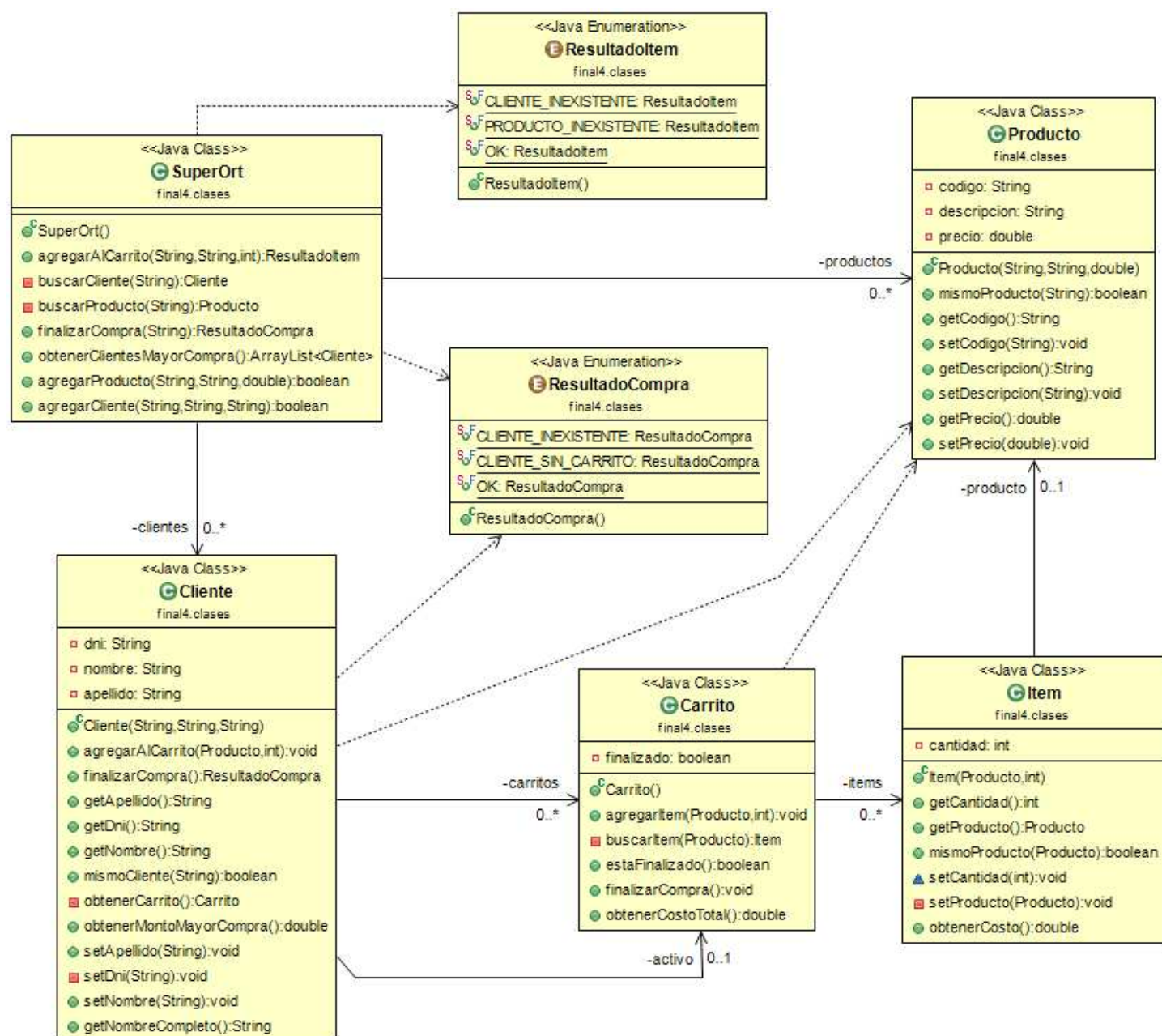
Leé con cuidado el enunciado y por lo menos dos veces para resolver lo pedido. Pensá bien la estrategia de resolución antes de comenzar el desarrollo de lo que te solicitan. El objetivo de este examen es **evaluar la correcta aplicación de los conceptos y técnicas** vistos hasta el momento:

- Correcta definición de clases y asignación adecuada de sus responsabilidades.
- Encapsulamiento, ocultamiento de información y uso de getters y setters sólo cuando corresponda.
- Modularización reusable y mantenible con uso de métodos con correcta parametrización.
- Correcta aplicación conceptual de las relaciones entre clases.
- Importación y exportación de Proyectos Java desde Eclipse.

**Enunciado**

SuperORT es un carrito de compras on-line que ya fue casi totalmente desarrollado. Sin embargo, ya al final de la etapa de desarrollo se encontraron algunas fallas en el diseño, fallas que debemos resolver además de agregar datos para que el programa pueda usarse.

El diseño de clases actual es el siguiente:



Lo que nos piden es:

**EXAMEN FINAL**

- Desarrollar la carga de clientes y productos en **SuperOrt**, cuidando que no haya elementos repetidos. La “clave” del cliente es el dni y la del producto el código de producto.
- Resolver el problema de integridad causado por la generación del “listado” de mejores clientes (`obtenerClientesMayorCompra`) que obtiene todos aquellos clientes con el carrito de mayor precio total. Para eso debemos realizar los siguientes cambios:
  - Crear una nueva clase **MejorCliente** que solo tenga dni y nombre completo.
  - Utilizarla para generar el listado de mejores clientes (modificando `obtenerClientesMayorCompra()` de **SuperOrt**).
  - Modificar la clase **Test** donde haga falta para que muestre todos los datos de los mejores clientes (según la nueva clase).
- Contemplando la inflación que sufrimos a diario:
  - Modificar la clase **Item** para que guarde el precio “congelado” del producto. Para esto, el precio debe copiarse del producto al ítem al momento de “finalizar” un carrito. Tener en cuenta que este cambio influirá en la obtención del precio total del carrito.
  - Agregar un método para la actualización del precio de un producto. Debe ser la única manera posible de actualizar el precio del producto. Debe mostrarse el producto actualizado o un mensaje de error.
- Resolver todos los problemas de integridad causados por la exposición inadecuada de atributos que deberían ser privados.