Fase 04

Introducción

Si analizamos las clases Empresa y Persona (junto con sus derivadas ClienteIndividuo y Vendedor) vemos que todas comparten un conjunto de atributos: Email, Direccion y Telefono.

Podemos entonces crear una clase, llamada por ejemplo DatosContacto, que reúna dichos atributos en una entidad. Más adelante sería posible en esta clase implementar métodos que permitiesen validar dichos atributos según las reglas que oportunamente se establecieran.

Por otro lado la clase Empresa posee un atributo llamado Contacto, que representa la persona de contacto en dicha empresa, sea un cliente empresa o proveedor. Podemos entonces definir Contacto como del tipo Persona. Como la clase Persona es abstracta, deberemos convertirla nuevamente en una clase concreta a fin de poder utilizarla.

Por otro lado, la clase Producto tiene dos atributos referidos a su categorización: Categoria y Subcategoria. Estas propiedades están codificadas como string, es decir, podemos básicamente asignarles cualquier valor. Sin embargo, en esta aplicación, las categorías y subcategorías estarán normalizadas, es decir, deberán poder asignarse desde un conjunto predefinido. Para ello es conveniente crear una nueva entidad, a la que llamaremos Categoria y codificar ambos atributos en forma consecuente.

Además, en la misma clase, tenemos un atributo Proveedor, también codificado como string. Sin embargo, ya sabemos que el proveedor es una empresa, y por lo tanto, codificaremos dicho atributo como tal.

Finalmente, revisemos la clase DocumentoComercial y sus derivadas Remito y Factura.

Ambos poseen un atributo que representa el detalle de los productos. Actualmente se encuentra codificado como string, pero evidentemente no es una buena idea codificar una lista de items en una string.

Para resolver este problema crearemos otra clase a la que llamaremos Item y se procederá a codificar el atributo Detalle como una colección de objetos Item.

Habilidades y conocimientos aplicados en esta fase

Comprender las relaciones entre objetos.

Codificar relaciones entres objetos.

Paso 1

Crear en Entidades una clase llamada DatosContacto con los atributos Email, Direccion y Telefono, todos del tipo string y codificados como propiedades autoimplementadas. Implementar su correspondiente constructor parametrizado.

Paso 2

Relacionar esta clase con las clases Empresa y Persona (y sus derivadas).

Paso 3

Realizar las modificaciones necesarias en la consola para corregir los errores que se han producido.

Paso 4

En la clase Empresa definir el atributo Contacto del tipo Persona.

Convertir la clase Persona es una clase concreta y realizar los cambios necesarios en la consola para corregir los errores que se han producido.

Paso 5

En el proyecto Entidades, crear la clase Categoria con su constructor parametrizado. Dicha clase tendrá por ahora un único atributo llamado Nombre del tipo string.

Paso 6

En la clase Producto, cambiar el tipo de dato de los atributos Categoria y Subcategoria a Categoria.

Paso 7

Realizar las modificaciones necesarias en la consola para corregir los errores que se han producido.

Paso 8

En la clase Producto, cambiar el tipo de dato del atributo Proveedor a Empresa.

Paso 9

Realizar las modificaciones necesarias en la consola para corregir los errores que se han producido.

Paso 10

En el proyecto Entidades, crear una clase denominada Item y codificar los siguientes atributos:

Cantidad Entero lectura/escritura

Descripcion String lectura/escritura

PrecioUnitario Decimal lectura/escritura

Importe Decimal lectura

Codificar un constructor parametrizado.

Paso 11

En la clase DocumentoComercial, codificar el atributo Detalle como lista de Item.

Paso 12

Realizar las modificaciones necesarias en la consola para corregir los errores que se han producido.

Paso 13

Para finalizar, podemos modificar el atributo Total de las clase DocumentoComercial codificándolo como atributo de sólo lectura, ya que será la suma del los importes de cada item.

Paso 14

Realizar las modificaciones necesarias en la consola para corregir los errores que se han producido.

Paso 15

En el proyecto Negocio, adaptar la clase admProductos reemplazando los parámetros referidos a categorias y subcategorias definidos como string, por parámetros del tipo Categoria. Renombrar el método ListarDetalle() como Listar(), ya que ahora es posible sobrecargar el mismo.

Finalmente modificar los métodos en la clase admCompras() para que utilicen la clase Item como parámetros.