

Guia de Objetos II

modelado e implementación

Sobre esta guía

Lea atentamente cada uno de los enunciados y antes de llevar a código cada ejercicio realice en papel o con la ayuda de algún software, el correspondiente diagrama de UML, detallando los atributos, métodos y constructores que considere necesarios. Una vez terminado escriba el código y ejecute las pruebas que se detallan.

Enunciados

1. Es necesario modelar el objeto de tipo Libro con las siguientes características, titulo, precio, stock y Autor, este último posee las características de nombre, apellido, email y genero ('M' o 'F'). Para este ejercicio vamos a asumir que un libro tiene un único autor. Ejecute las siguientes instrucciones:
 - a. Inicialice un objeto de tipo Autor, "Joshua", "Bloch", "joshua@email.com", 'M'.
 - b. Imprima por pantalla al autor previamente instanciado.
 - c. Inicialice el libro "Effective Java" del Autor "Joshua Bloch" que cueste 450 pesos con una cantidad de 150 copias.
 - d. Imprima por pantalla el libro instanciado.
 - e. Modifique el precio del libro "Effective Java" a 500 pesos y aumente la cantidad en 50 copias.
 - f. Imprima por pantalla los atributos del Autor Joshua, accediendo desde el Libro "Effective Java".
 - g. Agregue un método a la clase Libro que posibilite imprimir en pantalla el siguiente mensaje:
"El libro, {título} de {nombre del autor}. Se vende a {precio} pesos."
 - h. Modificar la clase Libro, para que acepte más de 1 Autor. Y realizar los cambios necesarios en el método del punto g, para imprimir un nuevo mensaje que liste los autores que contribuyeron a ese libro.
2. Nos contratan para hacer un programa que lleve el control de las ventas de un local. Para esto es necesario modelar la clase Cliente, que posee un atributo id como identificador del cliente, el mismo debe ser un valor compuesto por letras y números aleatorios que se generan automáticamente al crear un Cliente. El Cliente también posee un nombre, un email y un porcentaje de descuento.

Por otro lado vamos a tener el objeto Factura que representa una venta del local, cada Factura posee un identificador de las mismas características que el usado en Cliente. A su vez cada factura posee un monto total, una fecha y el Cliente que generó la compra. Para la fecha de la venta se le va a asignar la fecha y hora al momento de creación del objeto Factura. El tipo Factura debe contar con un método que calcule el monto final luego de aplicarle el descuento que posee el cliente.

- a. Investigue la clase UUID y sus métodos estáticos para la generación de los ids. Investigue la clase LocalDate y sus métodos estáticos para la generación de la fecha.
 - b. Cree un objeto de tipo Cliente, imprima sus detalles por pantalla. Para esto haga uso de un método que facilite la impresión del mismo.
Cliente[id=?, nombre=?, email=?, descuento=?]
 - c. Cree un objeto de tipo Factura que posea al Cliente anteriormente creado. Una vez hecho esto, imprima por pantalla el monto total de esta Factura y el monto total luego de aplicarle el descuento.
 - d. Cree un método que facilite la impresión del objeto de tipo Factura y que siga el siguiente formato.
Factura[id=?, fecha=?, monto=?, montoDesc=?, Cliente[id=?, nombre=?, email=?, descuento=?]]
 - e. Analizar de agregar el tipo ItemVenta, que representa un producto que forma parte de la venta. El mismo contiene un id, un nombre, descripción y precio unitario. Considere las modificaciones necesarias en el tipo Factura para que el mismo pueda almacenar múltiples Ítems de venta y a su vez calcular los montos totales con y sin el descuento aplicado.
3. Necesitamos crear un programa para manejar los datos de una Cuenta bancaria de un Cliente. Para esto identificamos los atributos id, nombre y género (M o F) para el Cliente. Por otro lado tenemos el tipo Cuenta que también posee un identificador, un balance y un Cliente que es el dueño de la cuenta. La Cuenta debe exponer los métodos depositar y extraer que modifican el balance de la misma. Algo a tener en cuenta es que el método extraer no puede extraer dinero si el balance total no lo permite, si esto sucede se debe imprimir un mensaje por pantalla que indique que la cuenta no posee saldo suficiente.
 - a. Crear un Cliente e imprimirlo en pantalla.
 - b. Crear una Cuenta bancaria para el Cliente anterior con un saldo inicial de 10000.
 - c. Realizar operaciones de depósito y extracción para probar esa funcionalidad.
 - d. Es necesario agregar una variante de Cuenta que permita un saldo deudor, de máximo 2000 pesos. Esto significa que la cuenta puede aceptar un balance negativo con un máximo de - 2000. Realice los cambios y pruebas necesarias.
 - e. Analice cómo implementaría con las herramientas conocidas hasta el momento, llevar un registro de como máximo 10 operaciones de depósito y extracción que se realizaron en la cuenta. En donde se almacene en memoria de alguna forma la siguiente cadena de texto:
 - i. Depósito: "El {NombreCliente}, depositó {MontoDepositado}"
 - ii. Extracción: "El {NombreCliente}, retiró {MontoRetirado}"