UTNMDP
Regional Mar del Plata

Guia #1
Introducción a la Programación Orientada a Objetos

1. Modele el objeto Empleado que posee las siguientes características, dni, nombre, apellido y salario. El mismo debe contar con la posibilidad de calcular el salario anual. A su vez se requiere otro método que permita aumentar el salario dependiendo del porcentaje que se le pase por parámetro. Considere crear un método que facilite imprimir por pantalla las características del objeto de la siguiente forma:
   Empleado[dni=?, nombre=?, apellido=?, salario=?]

   En el main, realice las siguientes operaciones:
   a. Inicialice un empleado Carlos Gutiérrez, con dni 23456345 y salario inicial de 25000.
   b. Inicialice un empleado Ana Sánchez, con dni 34234123 y salario inicial de 27500.
   c. Imprima ambos objetos por pantalla.
   d. Aumente el salario del empleado Carlos en un 15% e imprima en pantalla el salario anual del mismo.

2. Modele el objeto que representa la cuenta de un banco, con identificador, nombre y balance. Considere los getters, setters y constructores necesarios. Luego, realizar los siguientes métodos:
   a. El método crédito que representa un depósito de dinero en la cuenta. Este método debe devolver el balance luego de la operación.
   b. El método débito que representa una sustracción de dinero de la cuenta. Este método debe devolver el balance luego de la operación. Si el dinero en la cuenta no es suficiente para cubrir la sustracción, se debe imprimir por pantalla un aviso sin permitir dicha sustracción.
   c. Un método que imprima por pantalla las características del objeto.

   En el main, realice las siguientes operaciones:
   1. Inicialice una cuenta con un monto inicial de 15000.
   2. Realice una operación de crédito de 2500.

3. Realice una operación de compra de 1500.
4. Realice una operación de compra de 30000.
5. Imprima por pantalla los valores de la cuenta y verifique que el balance sea correcto.

3. Modela un objeto Libro que tenga las siguientes características: id (único y autoincremental), título, autor, precio y cantidad de copias disponibles. En dicha clase implementa los siguientes métodos:

- Un constructor que inicialice el Libro con su título, autor, precio y cantidad de copias. El id debe ser asignado automáticamente.
- Métodos getters y setters para cada atributo, excepto el id, que solo tendrá un getter.
- Un método que permita vender una cierta cantidad de copias de un libro, disminuyendo la cantidad disponible. Si no hay suficientes copias, debe mostrar un mensaje indicando que la operación no es posible.
- Un método que permita incrementar la cantidad de copias disponibles en el inventario.
- Un método que imprima por pantalla los detalles del libro en el siguiente formato: Libro[id=?, título=?, autor=?, precio=?, copias disponibles=?].

En el main, realiza las siguientes operaciones:

a. Inicializa un libro con el título "El Quijote", autor "Miguel de Cervantes", precio 500, y 10 copias disponibles.
b. Inicializa otro libro con el título "Cien Años de Soledad", autor "Gabriel García Márquez", precio 700, y 5 copias disponibles.
c. Imprime los detalles de ambos libros.
d. Vende 3 copias del primer libro.
e. Imprime los detalles del primer libro.
f. Intenta vender 8 copias del segundo libro.
g. Incrementa en 5 la cantidad de copias disponibles del segundo libro.
h. Imprime los detalles del segundo libro.

4. Modela un objeto ItemVenta que representa un ítem de venta en una tienda. Debe permitir realizar varias operaciones relacionadas con la gestión de ítems. La clase ItemVenta deberá tener como atributos: identificador (int), descripcion (String), cantidad (int) y precioUnitario (double). Los métodos a realizar son:
    - getters y setters para cada atributo.
    - Un método calcularPrecioTotal() que devuelve el precio total (precio unitario * cantidad).
    - Un método que devuelva una representación en cadena del ítem de venta en el formato: ItemVenta[id=?, descripcion=?, cantidad=?, pUnitario=?, pTotal=?].

En el main, realiza las siguientes operaciones **dentro de un switch**:

1. Agregar ítem: Inicializa un objeto ItemVenta con valores proporcionados por el usuario.
2. Imprime el objeto usando el método correspondiente.
3. Permite al usuario ingresar una nueva cantidad y actualiza el atributo cantidad del ítem de venta.
4. Permite al usuario ingresar un nuevo precio unitario y actualiza el atributo precioUnitario del ítem de venta.
5. Imprime el precio total calculado por el método calcularPrecioTotal().
6. Sale del programa.