Programación orientada a objetos

Guía de ejercicios #2

Temas que se abordan en esta guía: clases, herencia y constructores. Diagramas UML de clases, con relación de herencia y dependencia.

Algunos de los siguientes ejercicios propuestos serán resueltos en la práctica por los instructores de forma explicativa:

1. Realizar el diagrama UML de un empleado, que contenga:
    a. Los siguientes atributos:
        i. Identificación
        ii. Nombre
        iii. Dirección
    b. Métodos getters, setters y *toString* públicos.
    c. El Empleado va a heredar sus atributos a las siguientes clases, y generará un nuevo *toString* de ellas, que incluya sus atributos extras:
        i. *Operario*
            - Fecha de contratación
        ii. *Directivo*
            - Cantidad de empleados a cargo
    d. *Operario* va a heredar sus atributos y métodos a las siguientes clases, y va a generar un nuevo método toString:
        i. *Oficial*
            - Descripción del trabajo.
        ii. *Técnico*
            - Límite de fecha de entrega.
2. Elaborar un programa que contenga:
    a. Una clase abstracta llamada *Empleado* la cual contenga los atributos privados:
        i. *Identificación*.
        ii. *Nombre*.

iii. *Dirección*.
iv. De esta clase *Empleado* se deben heredar sus atributos y métodos a las clases:
    - *Operador*.
    - *Directivo*.
v. Imprimir los datos de al menos cinco empleados.

3. Construir el siguiente código fuente:
    a. Una clase *Costo*, con los atributos:
        i. *Precio*.
        ii. *Descuentos* (si aplica).
    b. Construir una clase *Factura* que herede de la clase *Costo* y además debe tener los atributos:
        i. *Emisor*.
        ii. *Cliente*.
        iii. Y el método *ImprimirFactura*.
    c. Finalmente imprimir el total.
4. Construir una clase final *Mate2*, esta debe heredar los métodos estáticos de *Mate1*:
    a. *máximo*.
    b. *mínimo*.
    c. *sumatorio*.
    d. *mediaAaritmética*.
    e. *mediaGgeométrica*.

    Debe devolver un método *toString* con todos los cálculos anteriores.

Ejercicios para que los alumnos desarrollen en la práctica de laboratorio (al menos uno de los siguientes): construir elaborar

1. Escribir un programa que contenga:
    a. La clase *Abuelo*, con atributos:
        i. *apellido* y *nacionalidad*.
    b. Debe heredarle los atributos y métodos al hijo e imprimir los datos.

c. Luego el hijo debe heredar los atributos del hijo del abuelo e imprimir los datos. Es decir, el nieto heredará los atributos del padre y del abuelo.

2. Elaborar el diagrama UML siguiente:
    a. Crear una clase *Programador* con atributos privados para almacenar:
        i. Nombre.
        ii. Identificación.
        iii. Edad.
        iv. Estado civil.
        v. Salario.
        vi. Líneas de código por hora.
        vii. Lenguaje dominante.
    b. Incluir el constructor, los métodos getters, setters y *toString*.
    c. Luego la clase *Programador* heredará sus atributos al Supervisor Informático y le sumará el salario del programador más un 30% por su servicio.
    d. Finalmente imprimir los datos con un método *toString*.
3. Diseñar el siguiente diagrama UML:
    a. Una clase *Voleibolista* con los atributos privados:
        i. *Id*.
        ii. *Nombre*.
        iii. *Apellidos*.
        iv. *Edad*.
        v. *Peso*.
        vi. *Demarcación*.
    b. Y los métodos:
        i. *Concentrarse*.
        ii. *Viajar*.
        iii. *Partido*.
        iv. *Entreno*.
    c. Luego la clase *Entrenador* heredará todos los atributos excepto:
        i. *Peso*.

ii. *Demarcación*.

d. Y *Entrenador* agregará los siguientes atributos privados:
    i. *Federación*.
e. Además, *Entrenador* no heredará los métodos de:
    i. *Partido*.
    ii. *Entreno*.
f. Pero, *Entrenador* va a añadir métodos para:
    i. Dirigir Partido.
    ii. Dirigir Entrenamiento.
g. Finalmente, la clase *Familiar*, va a heredar los atributos del Voleibolista, excepto:
    i. *Peso*.
    ii. *Demarcación*.
h. Pero va a añadir:
    i. Costo de entrenamiento.
    ii. Aportación de suplementos.
i. Además, *Familiar* heredará todos los métodos de *Voleibolista* excepto:
    i. *Partido*.
    ii. *Entrenar*.
j. Pero va a añadir:
    i. *Motivar*.
k. Lo que debe hacer es, una clase padre llamada *SelecciónDeVoleibol*, que herede todos los atributos y métodos en común para las subclases:
    i. *Voleibolista*.
    ii. *Entrenador*.
    iii. *Familiar*.
l. Y que las subclases, solo tengan los atributos y métodos que no se encuentren previamente establecidos en la clase padre.