UML y Java

Se pide resolver cada uno de los ejercicios utilizando diagramas de clases UML y el lenguaje de programación Java.

[E.01] - A modelar

Hay que modelar la clase Persona, la cual tiene como atributos: apellido, nombre y año de nacimiento. Posee métodos que permiten:

- Retornar el nombre completo de la persona.
- Obtener su edad actual.
- Conocer si es mayor que una edad dada.
- Cambiar el nombre.
- Cambiar el apellido.
- Cambiar el nombre y apellido.
- Mostrar toda la información de la persona.

En el método main, probar todos los métodos.

[E.02] - Empleado

Diseñar la clase Empleado, cuyos atributos sean su DNI, nombre, apellido, salario base, estado civil (soltero, casado, divorciado o viudo) y cantidad de hijos.
Se sabe que todos los empleados cobran el salario base más un extra del 4% por cada hijo, con un tope de hasta 15%. Del salario resultante debe descontarse un 4% en caso de que el estado civil sea soltero.
Desarrollar el método *obtenerSalarioFinal*, que no recibe parámetros y que retorna el valor del salario del empleado, según lo descrito anteriormente.
En el método main, instanciar, al menos, tres empleados y probar todos los métodos.

[E.03] - UTN bot

Se necesita un robot (que tiene nombre) que permita atender llamadas telefónicas. La compañía puede detectar algunos clientes según su número de teléfono, sin embargo, en otros casos no. Por ello, el robot debe ser capaz de procesar alguno de los siguientes métodos homónimos:

- saludar(): void
  Muestra por consola un saludo diciendo: "Hola, mi nombre es ___. ¿En qué puedo ayudarte?".

- saludar(Persona): void
  Muestra por consola un saludo diciendo: "Hola ___, mi nombre es ___. ¿En qué puedo ayudarte?".

Modelar la clase Robot en UML y luego codificar en Java.
En el método main, probar varias veces, con diferentes variantes para ver si saluda como corresponde.

[E.04] - La típica cuenta bancaria

Se pide modelar en UML la clase CuentaBancaria, la cual posea como atributos la clave bancaria uniforme CBU), el tipo (caja de ahorro o cuenta corriente) y el saldo (inicialmente en 0). Cómo métodos posee:

- Obtener el saldo actual.
- Depositar dinero en la cuenta (actualizando el saldo).
- Extraer dinero de la cuenta (actualizando el saldo). Solo puede quedar en saldo negativo si es cuenta corriente.
- Obtener los últimos 3 dígitos del CBU.

Codificar en Java y en el método main, crear una cuenta bancaria de cada tipo y probar todos sus métodos.

[E.05] - La clave es la password

Se pide modelar la clase Password, que posee un único atributo, el valor de la contraseña (String). Debe responder a los siguientes métodos:

- boolean esFuerte:
  Devuelve si la password es fuerte o no. Una password es fuerte cuando posea al menos 8 caracteres.
- boolean nuevoValor:
  Establece como nuevo valor de contraseña el recibido como parámetro, siempre y cuando su longitud sea mayor o igual a 6, si no, lo deja como estaba. Devuelve si se pudo o no establecer el valor.
- generarAleatorio():
  Devuelve una cadena que representa un valor de la contraseña aleatoria cuya longitud coincide con el parámetro entero recibido. Si el parámetro es menor que 6, devuelve null.

Además, deben poder crear contraseñas con o sin valor inicial, por ello es que la clase contará con un constructor sobrecargado:

- Password(String):
  Crea una contraseña cuyo valor viene dado por parámetro de tipo cadena que recibe.
- Password():
  Crea una contraseña cuyo valor se crea automáticamente.

[E.06] - Empleado recargado

Se pide el refactoreo de la clase *Persona* del ejercicio [E.01], cambiando el año de nacimiento por su fecha de nacimiento y agregando el atributo domicilio, que contenga calle, altura y barrio.

[E.07] - Cuenta plus

Se pide el refactoreo de la clase *CuentaBancaria* del ejercicio [E.04], agregando el atributo titular, que representa a la persona titular de la cuenta, y el atributo fechaDeApertura.

[E.08] - A combinar todo

Utilizando las clases generadas hasta ahora, se pide primero, modelar en UML, segundo, Francia y tercero codificar en Java la siguiente situación:
Una cuenta bancaria de tipo caja de ahorros le pertenece a Juan Pérez, nacido el 06/04/1986 y otra de tipo cuenta corriente le pertenece a María Fernández, nacida el 09/01/1990. Ambos están casados y viven juntos en Av. Siempreviva 742, Campo primaveral.

Supongamos que Juan y María se mudan a un nuevo hogar: ¿Hay que cambiar el domicilio de cada uno o basta con cambiar uno de los dos?

[E.09] - ¿Imprimido o impreso?

Desarrollar la clase Impresora, que contenga atributos que indicarán si está o no encendida, la cantidad de hojas actualmente en su bandeja y el nivel de tinta. Inicialmente, toda impresora está apagada, sin hojas y con nivel de tinta en 100.
Debe responder a los siguientes métodos:

- nivelSegunCantCaracteres():
  Retorna cuánta cantidad de tinta debería usarse según la cantidad de caracteres recibida por parámetro.
- recargarBandeja():
  Suma a la bandeja una cantidad de hojas. El máximo de la bandeja es de 35 hojas. Se debe verificar no excederse de tal valor. Si el parámetro es negativo, la bandeja se deja como está.
- imprimir():
  Recibe como parámetro un Documento y muestra por consola la fecha, el título y cuerpo del mismo, con el siguiente formato:

Fecha: dd/mm/aaaa TÍTULO Cuerpo

  Al hacerlo, se descuenta 1 punto de nivel de tinta por cada 50 caracteres del cuerpo impresos y se resta 1 hoja de la cantidad en bandeja por cada 20 caracteres del cuerpo impreso. Se debe verificar antes de imprimir que se cuente con nivel de tinta y cantidad de hojas suficientes.