Práctica UT1.1 KOTLIN

PRIMERA PARTE (50%)

Haz una clase llamada Persona que siga las siguientes condiciones::

- Sus atributos son: nombre, edad, DNI, sexo (HOMBRE, MUJER), peso y altura. No queremos que se accedan directamente a ellos. Piensa que modificador de acceso es el más adecuado, también su tipo. Si quieres añadir algún atributo puedes hacerlo.
- Por defecto, todos los atributos menos el DNI serán valores por defecto según su tipo (0 números, cadena vacía para String, etc.). Sexo será hombre por defecto, usa un **enum class**
- Se implantarán varios constructores:
    - Un constructor por defecto.
    - Un constructor con el nombre, edad y sexo, el resto por defecto.
    - Un constructor con todos los atributos como parámetro.
- Los métodos que se implementaran son:
    - calcularIMC(): calculara si la persona está en su peso ideal (peso en kg/(altura^2 en m)), si esta fórmula devuelve un valor menor que 20, la función devuelve un -1, si devuelve un número entre 20 y 25 (incluidos), significa que está por debajo de su peso ideal la función devuelve un 0 y si devuelve un valor mayor que 25 significa que tiene sobrepeso, la función devuelve un 1. Te recomiendo que uses constantes para devolver estos valores.
        - esMayorDeEdad(): indica si es mayor de edad, devuelve un booleano.
        - comprobarSexo(aqui la clase enum anteror): comprueba que el sexo introducido es correcto. Si no es correcto, será H. No será visible al exterior.
        - toString(): devuelve toda la información del objeto.
        - generaDNI(): genera un número aleatorio de 8 cifras, genera a partir de este su número su letra correspondiente. Este método será invocado cuando se construya el objeto. Puedes dividir el método para que te sea más fácil. No será visible al exterior

    - Métodos set de cada parámetro, excepto de DNI.

Crea las **funciones ejecutables auxiliares que necesites y el main() de kotlin** con la siguiente funcionalidaad:

- Pide por teclado el nombre, la edad, sexo, peso y altura.
- Crea 3 objetos de la clase anterior, el primer objeto obtendrá las anteriores variables pedidas por teclado, el segundo objeto obtendrá todos los anteriores menos el peso y la altura y el último por defecto, para este último utiliza los métodos set para darle a los atributos un valor.
- Para cada objeto, deberá comprobar si está en su peso ideal, tiene sobrepeso o por debajo de su peso ideal con un mensaje.
- Indicar para cada objeto si es mayor de edad.
- Por último, mostrar la información de cada objeto.

SEGUNDA PARTE (50%)

En las siguientes clases debéis completar los detalles (métodos, atributos, etc) siguiendo la lógica del enunciado.

Crear una clase **Empleado** derivada de Persona que añade como atributos:

- departamento (enumerado : CONTABILIDAD, RRHH,VENTAS, ALMACEN
- sueldo base
- año de incorporación al puesto de trabajo

Y los métodos:

- sueldoBruto(extra: Float) que se calcula con el sueldo base más el extra por cada año trabajado y un incremento del 10% del sueldo base si es del departamento de VENTAS. El extra debe ser menor de 200 €
- Redefinir toString()

Crear una clase "**Empresa**" con una propiedad que sea una lista de empleado con el siguiente método

- listarEmpleados(…) el parámetro será una función (normalmente una lambda) que aplique un filtro sobre empleados.
- Atributos y métodos para insertar empleados

Para realizar las pruebas de este programa podéis crear una función que inserte los datos sin necesidad de pedir desde consola.