1. Una clase llamada Serie con las siguientes características: Sus atributos son titulo, numero de temporadas, prestado, genero y creador.

Los siguientes métodos:

* entregar(): cambia el atributo prestado a true.
* devolver(): cambia el atributo prestado a false.
* isEntregado(): devuelve el estado del atributo prestado.

1. Una clase Videojuego con las siguientes características:

Sus atributos son titulo, horas estimadas, prestado, genero y compañia.

Los siguientes métodos:

* entregar(): cambia el atributo prestado a true.
* devolver(): cambia el atributo prestado a false.
* isEntregado(): devuelve el estado del atributo prestado.

1. Crea una clase llamada Cuenta que tendrá los siguientes atributos: titular y cantidad (puede tener decimales).

Tendrá dos métodos especiales:

* ingresar(cantidad): se ingresa una cantidad a la cuenta, si la cantidad introducida es negativa, no se hará nada.
* retirar(cantidad): se retira una cantidad a la cuenta, si restando la cantidad actual a la que nos pasan es negativa, la cantidad de la cuenta pasa a ser 0.

1. Una clase llamada Persona que siga las siguientes condiciones:

Sus atributos son: nombre, edad, cedula, sexo (H hombre, M mujer), peso y altura.

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 una constante para ello.

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(sexo): 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.

1. Una clase llamada Password que siga las siguientes condiciones:

Que tenga los atributos longitud y contraseña

Los métodos que implementa serán:

* esFuerte(): devuelve un booleano si es fuerte o no, para que sea fuerte debe tener más de 2 mayúsculas, más de 1 minúscula y más de 5 números.
* generarPassword():  genera la contraseña del objeto con la longitud que tenga.

1. Una clase llamada Electrodomestico con las siguientes características:

Sus atributos son precio base, color, consumo energético (letras entre A y F) y peso.

Los métodos que implementara serán:

* comprobarConsumoEnergetico(letra): comprueba que la letra es correcta, sino es correcta usara la letra por defecto. Se invocará al crear el objeto y no será visible.
* comprobarColor(color): comprueba que el color es correcto, sino lo es usa el color por defecto. Se invocará al crear el objeto y no será visible.
* precioFinal(): según el consumo energético, aumentara su precio, y según su tamaño.

1. Una clase llamada Raices, donde representaremos los valores de una ecuación de 2º grado.  
   Tendremos los 3 coeficientes como atributos, llamémosles a, b y c.   
   Las operaciones que se podrán hacer son las siguientes:

* obtenerRaices(): imprime las 2 posibles soluciones
* obtenerRaiz(): imprime única raíz, que será cuando solo tenga una solución posible.
* getDiscriminante(): devuelve el valor del discriminante , el discriminante tiene la siguiente formula, (b^2)-4\*a\*c
* tieneRaices(): devuelve un booleano indicando si tiene dos soluciones, para que esto ocurra, el discriminante debe ser mayor o igual que 0.
* tieneRaiz(): devuelve un booleano indicando si tiene una única solución, para que esto ocurra, el discriminante debe ser igual que 0.
* calcular(): mostrara por consola las posibles soluciones que tiene nuestra ecuación, en caso de no existir solución, mostrarlo también.

1. Queremos representar con programación orientada a objetos, un aula con estudiantes y un profesor. Tanto de los estudiantes como de los profesores necesitamos saber su nombre, edad y sexo. De los estudiantes, queremos saber también su calificación actual (entre 0 y 10) y del profesor que materia da.  
   Las materias disponibles son matemáticas, filosofía y física.   
   El profesor tiene un 20% de no encontrarse disponible (reuniones, baja, etc.)  
   El aula debe tener un identificador numérico, el número máximo de estudiantes y para que esta destinada (matemáticas, filosofía o física). Piense que más atributos necesita.  
   Un aula para que se pueda dar clase necesita que el profesor esté disponible, que el profesor de la materia correspondiente en el aula correspondiente (un profesor de filosofía no puede dar en un aula de matemáticas) y que haya más del 50% de alumnos.  
   El objetivo es crear un aula de alumnos y un profesor y determinar si puede darse clase, teniendo en cuenta las condiciones antes dichas.  
   Si se puede dar clase mostrar cuantos alumnos y alumnas (por separado) están aprobados de momento (imaginad que les están entregando las notas).