

5.G. Actividades propuestas.

Sitio: [Formación Profesional a Distancia](#)

Curso: Programación

Libro: 5.G. Actividades propuestas.

Imprimido por: Iván Jiménez Utiel

Día: martes, 7 de enero de 2020, 22:26

Tabla de contenidos

[1. ActividadUT05-1: Clase Libro.](#)

[2. ActividadUT05-2: Clase Libro2.](#)

[3. ActividadUT05-3: Clase Persona.](#)

1. ActividadUT05-1: Clase Libro.

Desarrolla una [clase](#) denominada **Libro** con los siguientes datos:

Miembros dato u atributos:

- **Título**, **autor**, **código** (por ejemplo de tipo numérico) y **prestado** ([booleano](#)).

Miembros [método](#) u operaciones:

- Dos constructores, uno con argumentos y otro por defecto.
- Métodos getters y setters
- [Método](#) **prestar** (devuelve true en caso de préstamo exitoso y false en caso de estar ya prestado)
- [Método](#) **devolver** (devuelve true en caso de poder devolverse, por estar prestado, y false en caso de no poder devolverse por no estar aún prestado)

Incorpora una [clase](#) llamada **PruebaLibro**, en la que se instancien 2 libros usando los 2 posibles constructores. Para el libro instanciado con el constructor por defecto, incorpora sus atributos, leyéndolos de teclado y usando métodos setters.

- Haz pruebas de préstamos y devoluciones de libros.
- Muestra los atributos de los libros con métodos getters.

2. ActividadUT05-2: Clase Libro2.

Desarrolla una [clase](#) denominada Libro con los siguientes datos:

Miembros dato o atributos:

- **Título, autor, código** (por ejemplo de tipo numérico), **número de ejemplares** y **número de ejemplares prestados**.

Nota: en este caso, con respecto al ejercicio anterior, consideramos que de un mismo código de libro, puede haber más de un ejemplar de libro. Por ello, ya no tiene sentido utilizar un booleano, para ver si se ha prestado (ya que tendríamos que tener una array de booleanos de préstamos, uno para cada ejemplar) un libro. Simplemente llevaremos el [control](#) de préstamos de un determinado código de libro, mediante una [variable](#) numérica que indique el número de libros prestados. Cuando todos los libros de un mismo código estén prestados, entonces el número de ejemplares y el de ejemplares prestados será el mismo.

Miembros [método](#) u operaciones:

- Dos constructores, uno con argumentos y otro por defecto.
- Métodos getters y setters
- [Método prestar](#) (devuelve true en caso de préstamo exitoso y false en caso de estar ya prestados todos los libros), que actualiza el número de ejemplares prestados.
- [Método devolver](#) (devuelve true en caso de poder devolverse, por estar prestado algún libro, y false en caso de no poder devolverse por no estar aún prestado ningún libro), que actualiza el número de ejemplares prestados.

Incorpora una [clase](#) llamada **PruebaLibro**, en la que se instancien 2 libros usando los 2 posibles constructores. Para el libro instanciado con el constructor por defecto, incorpora sus atributos, leyéndolos de teclado y usando métodos setters.

Haz pruebas de préstamos y devoluciones de libros.

Muestra los atributos de los libros con métodos getters.

3. ActividadUT05-3: Clase Persona.

Haz una [clase](#) llamada **Persona** que siga las siguientes condiciones:

Miembros **dato o atributos**:

- Sus atributos son: **nombre**, **edad**, **DNI**, **sexo** (H para hombre y M para mujer), **peso** (en kilogramos, admitiéndose decimales) y **altura** (en metros, admitiéndose decimales). Dichos atributos no serán accedidos directamente, por lo que deber tener en cuenta cuál será el modificador de acceso es el más apropiado, así como su tipo.

Miembros [método](#) u operaciones:

- Se incorporarán varios constructores:
 - o Un constructor por defecto.
 - o Un constructor con el nombre, DNI, edad y sexo, el resto por defecto.
 - o Un constructor con todos los atributos como parámetro.
- [Método](#) que devolverá el **índice de masa corporal**, que vendrá dado por el cálculo:
$$(\text{peso en kg})/(\text{altura}^2 \text{ en m})$$
- [Método](#) que devolverá una **valoración** acerca del peso teniendo en cuenta el índice de masa corporal obtenido anteriormente. Devolverá una cadena que indicará si está por debajo del peso normal ($\text{IMC} < 20$), por encima del peso normal ($\text{IMC} \geq 25$), o con el peso dentro de lo considerado normal (los demás casos).
- [Método booleano](#) que indica **si es mayor de edad**.
- [Método](#) con retorno void para **comprobar el sexo** recibiendo como parámetro un carácter. Comprobará que el sexo introducido es correcto ('H' o 'M'). Si no es correcto, lo pondrá a 'H'. No será visible al exterior.
- **toString()**: devuelve toda la información del [objeto](#) (sobreescribiremos el [método](#)). Mostrará entre otras cosas el IMC y la valoración de peso.

Además, crea una [clase](#) ejecutable **PruebaPersona** que haga lo siguiente:

- Pide por teclado el nombre, la edad, DNI, 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.
- Indicar para cada [objeto](#) si es mayor de edad. Para ello crearemos dentro de la [clase](#) **PruebaPersona** un [método estático](#) que reciba como parámetro un [objeto](#) de la [clase](#) **Persona**.
- Por último, mostrar la información de cada [objeto](#), mediante el [método](#) **toString()** mencionado anteriormente.