

Persona01

- Programa persona con nombre y edad, constructor sin y con parámetros. Métodos para pedir datos al usuario, imprimir esos datos y comprobar si es mayor o menor de edad.

Persona02

Haz una clase llamada Persona que siga las siguientes condiciones:

- Sus atributos son: nombre, edad, DNI, sexo (H hombre, M 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

Se implementarán 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()`: calcula si la persona está en su peso ideal (peso en $\text{kg}/(\text{altura}^2 \text{ en m}^2)$), 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.
- `esMayorDeEdad()`: indica si es mayor de edad, devuelve un booleano.
- `comprobarSexo(char 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.

Crea una clase ejecutable que haga lo siguiente:

- 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.

Password

Haz una clase llamada Password donde:

Atributos: longitud y contraseña. Por defecto, la longitud será de 11.

Los constructores:

- Un constructor por defecto.
- Un constructor con la longitud que nosotros le pasemos.

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()`: el usuario genera la contraseña del objeto con la longitud mayor o igual a 11.
- Método `get` para contraseña y longitud. Método `set` para longitud.

Crea un programa en el que existan tres objetos de la clase Password, que el usuario les de valores y comprueba que cada usuario tiene una contraseña fuerte. Obligar al usuario a que la contraseña sea “fuerte”

EJEMPLO de funcionamiento:

Dame una contraseña (Mínimo 11 caracteres)

`asfsagawgsegawgh3`

Dame una contraseña (Mínimo 11 caracteres)

`asgsagsagsaf35435`

Dame una contraseña (Mínimo 11 caracteres)

`ADGqe123456`

La contraseña es `asfsagawgsegawgh3` y su longitud es 17

La contraseña es `asgsagsagsaf35435` y su longitud es 17

La contraseña es `ADGqe123456` y su longitud es 11

La contraseña `asfsagawgsegawgh3` no es fuerte

La contraseña `asgsagsagsaf35435` no es fuerte

La contraseña es `ADGqe123456` es fuerte

Libro

Crear una clase Libro que contenga los siguientes atributos:

- ISBN
- Título
- Autor
- Número de páginas

Crear sus respectivos métodos get y set para cada atributo.

Crear el método toString() para mostrar la información relativa al libro con el siguiente formato: "El libro con ISBN creado por el autor tiene páginas"

La clase Main:

- Crear 2 objetos Libro (los valores que se quieran) y mostrarlos por pantalla.
- Indicar cuál de los 2 tiene más páginas.

Raices01

Realizar una clase llamada Raíces, donde representaremos los valores de una ecuación de 2º grado.

Atributos:

- Tendremos los 3 coeficientes como atributos, llamémosles a, b y c. □ Hay que insertar estos 3 valores para construir el objeto.

Las operaciones que se podrán hacer son las siguientes:

- obtenerRaices(): devuelve las 2 posibles soluciones
- getDiscriminante(): devuelve el valor del discriminante (double), 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.

Formula ecuación 2º grado: $(-b \pm \sqrt{(b^2) - (4*a*c)}) / (2*a)$

Raices02

Igual que raices01, pero de otra forma, retocando parte visual.

Cuadrado01

Realizar una clase llamada Cuadrado01 que pide el valor del lado de un cuadrado y halla el área y el perímetro.

Cuadrado02

Igual que Cuadrado01, pero en lugar de usar métodos que obligan a devolver algo usamos métodos void.

Empleado

Realizar una clase llamada Empleado que pide nombre, DNI y sueldo al usuario y dependiendo del sueldo el programa dice si pagara impuesto o no.

Operaciones01

Programa que pide datos al usuario y halla la suma, resta, multiplicación, división y resto.

Operaciones02

Programa que pide datos al usuario y halla la suma de los pares y los impares, el total de pares y de impares y la media.

Punto

Punto en el plano con lo que lleva 2 coordenadas, calculamos la ecuación de la recta que pasa por dos puntos Donde $A(x_1, y_1)$ y $B(x_2, y_2)$

Triangulo

Programa que pide los 3 lados de un triángulo y determina si es Equilátero (3 lados iguales), Isósceles (2 lados iguales) o Escaleno (ningún lado igual).
También he incluido un método para hallar el área de un rectángulo.