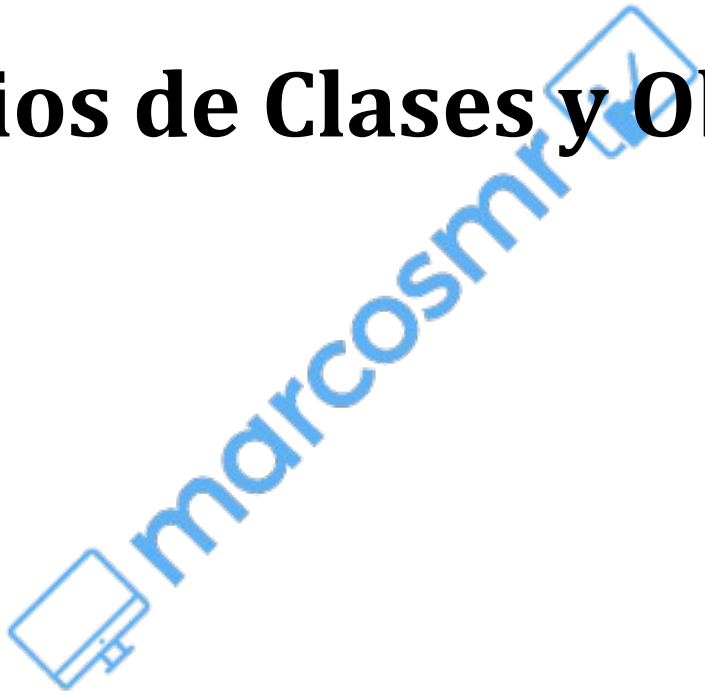
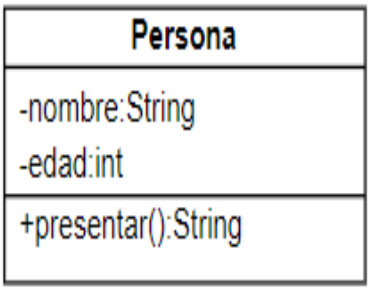
Ejercicios de Clases y Objetos



Relación de ejercicios

Ejercicio 1. Implementa una clase que nos permita modelizar una persona. Las características, propiedades o atributos de una persona vienen determinados por su nombre (cadena de texto) y su edad (número entero). Una persona sabe presentarse lo que se traduce en devolver un texto del estilo "Hola, soy *nombre* y tengo *edad* años".



Ejercicio 2. Amplía la clase Persona anterior para que tenga un constructor que reciba por parámetro el nombre y la edad. Con dichos valores se deben inicializar los atributos de la clase.

Ejercicio 3. Amplía la clase Persona anterior para que tenga un constructor que únicamente reciba la edad por parámetro y asigne como nombre por defecto "Anónimo".

Ejercicio 4. Implementa una clase que cree dos objetos de la clase Persona (invéntate los valores de nombre y edad). Dicha clase debe estar en un paquete diferente a la clase Persona. La primera persona debe ser creada haciendo uso del constructor con dos parámetros de este **ejercicio**. La segunda persona debe hacer uso del constructor al que únicamente se le pasa la edad implementado en este **ejercicio**. Una vez creados los dos objetos, haz que las dos personas se presenten.

Ejercicio 5. Implementa una clase, llamada Producto, que permita modelizar un producto a utilizar en una tienda. Ten en cuenta las siguientes características.

- El producto tiene un nombre y un precio.
- Si no nos indican precio, debemos establecerlo a 1.5 € por defecto.
- Una vez establecido el nombre de un producto, no se debe poder cambiarlo. El precio, sí.
- Debe contar con un método llamado creaEtiqueta que nos devuelva una cadena de texto similar a: `[Plátano: 1.5]`

Una vez creada la clase, implementa un programa que genere 5 productos (¿qué tal si los guardamos en un array?). Los tres primeros tienen que tener el precio que quieras y los otros dos, el precio por defecto. Posteriormente, muestra por pantalla las etiquetas de todos los productos.

Ejercicio 6. Implementa una clase llamada Validador. Dicha clase, que debe estar dentro de un nuevo paquete, debe tener dos métodos estáticos:

a) Uno llamado `estaEntre`, que reciba tres parámetros enteros. El primer parámetro será el número a validar, el segundo será el valor mínimo que puede tener el número pasado como primer parámetro y el tercer parámetro será el valor máximo que puede tener. El método debe devolver si el número está en el intervalo entre mínimo y máximo.

b) Otro llamado `tieneContenido`, que reciba una cadena de texto y diga si tiene contenido (no está vacía y su contenido no son solamente espacios en blanco).

Ejercicio 7. Implementa un programa que permita almacenar el nombre y edad de 10 personas. La información de las personas deberemos pedírsela al usuario. La edad debe estar entre 0 y 99 años. Posteriormente, una vez pedidos los datos de todas las personas, debemos hacer que todas las personas se presenten.

Ejercicio 8. Implementa un programa que permita almacenar el nombre y edad de un grupo de personas (no sabemos el número de personas que vamos a almacenar). La información de las personas deberemos pedírsela al usuario. La edad debe estar entre 0 y 99 años. Posteriormente, una vez pedidos los datos de todas las personas, debemos hacer que todas las personas se presenten. Se permite que dos personas tengan el mismo nombre y edad.

Ejercicio 9. Implementa una clase llamada Entrada. Dicha clase, que debe estar dentro del mismo paquete que la clase Validador, debe tener métodos estáticos que le soliciten al usuario la introducción, por teclado, de números enteros, números reales de doble precisión y cadenas de texto. Implementa un programa que haga uso de la clase Entrada.

Ejercicio 10. Implementa una clase llamada Aleatorio que esté en el mismo paquete de la clase Validador. Dicha clase debe contener tres métodos:

- Un método llamado generaAleatorioEntero que reciba dos números enteros y devuelva un número aleatorio entre ambos.
- Un método llamado generaAleatorioReal que permita generar valores aleatorios de números reales.
- Un método llamado generaAleatorioBooleano que permita generar un verdadero o un falso de forma aleatoria.

Ejercicio 11. Implementa un programa que lance una moneda al aire tantas veces como diga el usuario. Posteriormente debe indicar cuántas veces ha salido cara y cuántas veces cruz.

Ejercicio 12. Crea una clase llamada Meteo que tenga los siguientes atributos enteros: temperatura (entre -10 y 55 °C), humedad (entre 0 y 100%) y presión atmosférica (entre 1013 y 2026 hPa). Dado un objeto Meteo, con sus valores correspondientes, debemos saber si es probable que llueva y si hay probabilidad de fuertes vientos según la siguiente lógica:

- Es probable que llueva si: la temperatura superior a 10 °C, la humedad es superior al 65% y la presión atmosférica es inferior a 1650 hPa.
- Es probable que haya fuertes vientos si: la temperatura está entre 15 y 25°C, la humedad está entre 40 y 80% (excluidos) y la presión es superior a 1050 hPa.
- Si al establecer valor de algún parámetro físico se supera al máximo de ese parámetro, se establecerá el máximo para el mismo. Por ejemplo, si se intenta establecer una humedad del 110, su valor se ajustará a 100 automáticamente.
- Si al establecer valor de algún parámetro físico no alcanza el mínimo valor de ese parámetro, se establecerá el mínimo valor para el mismo. Por ejemplo, si se intenta establecer una humedad del -5%, su valor se ajustará a 0 automáticamente.

Ejercicio 13. Implementa un programa que genere 100 partes meteorológicos (con temperatura, humedad y presión atmosférica generadas de forma aleatoria). Posteriormente, nos debe informar si hay probabilidad de lluvia y/o fuertes vientos en función del parte meteorológico. Un ejemplo de visualización se puede ver en la siguiente imagen.

```
Parte nº24
T: 18   H: 70   P: 1462         ¿Lluvias? Alta probabilidad     ¿Vientos? Alta probabilidad

Parte nº25
T: 52   H: 3    P: 1396         ¿Lluvias? Baja probabilidad     ¿Vientos? Baja probabilidad

Parte nº26
T: -3   H: 7    P: 1334         ¿Lluvias? Baja probabilidad     ¿Vientos? Baja probabilidad

Parte nº27
T: 37   H: 47   P: 1373         ¿Lluvias? Baja probabilidad     ¿Vientos? Baja probabilidad

Parte nº28
T: 28   H: 99   P: 1146         ¿Lluvias? Alta probabilidad     ¿Vientos? Baja probabilidad
```

Ejercicio 14. Implementa una clase para gestionar una línea de un pedido (llamada LineaPedido). Un pedido (que no hay que implementar) debe contener una o más líneas de pedido. Cada línea de pedido consta de un producto (debemos hacer uso de la clase Producto implementada) y un número de unidades compradas de dicho producto. Una vez creada una línea de pedido no se debe poder modificar el producto asociado. Por cada línea de pedido se desea saber el subtotal. Por ejemplo, si compramos 5 unidades de un producto que vale 8, el subtotal de esa línea de pedido es 40. Además, la clase debe generar una cadena de texto con la información similar a la siguiente imagen (unidades, nombre y subtotal). Intenta que los productos y sus precios sean aleatorios. En otro caso, da los valores que desees.

```
5       Fresa           5,78
```

Ejercicio 15. Implementa un programa que cree 5 líneas de pedido y saque por pantalla la información relativa a cada uno de ellos. Para ello, deberás crear antes una serie de productos haciendo uso de la clase Producto creada anteriormente.

```
PEDIDO REALIZADO
-----------------------------------
Uds     Producto        Subtotal

5       Fresa           5,78
2       Naranja         2,49
3       Uva             7,85
10      Pera            28,81
3       Manzana         8,89
```

Ejercicio 16. Implementa una clase que permita modelizar un Pedido. Un pedido está compuesto por una fecha y hora así como varias líneas de pedido. Una vez creado un pedido, podemos agregar una nueva línea de pedido pero no debemos poder modificar ni la fecha ni las líneas de pedido (más allá de poder ir agregando líneas de pedido). Además, nos interesa poder mostrar una factura en la que se muestre: fecha y hora del pedido, líneas de pedido (con su subtotal) y el total de lo que nos costará el pedido. Intenta que los valores sean aleatorios.

Ejercicio 17. Implementa un programa que cree un pedido y muestre toda la información del mismo. Se indica un ejemplo de visualización.

```
FACTURA (08/11/2022 17:24)
**********************************
Uds     Producto        Subtotal

5       Plátano         15,00
3       Melón           5,25
1       Fresa           2,00
----------------------------------
                TOTAL: 22,25
```