



Ejercicios: POO básicos

Cajita de sorpresas

Clase CajaSorpresa que:

Inicializa con una lista de objetos sorpresa (pueden ser strings: "peluche", "chocolate", "juguete").

Método abrir() que elige uno al azar y lo devuelve/imprime.

Método agregar_sorpresa(objeto) para añadir nuevas sorpresas.

Bonus: si la caja queda vacía, imprime "¡La caja está vacía!".

Semáforo de colores

Clase Semaforo que:

Empieza en rojo.

Método cambiar() que recorre los colores: rojo → verde → amarillo → rojo.

Imprime el color actual.

Bombilla inteligente

Clase Bombilla que:

Tiene estado encendida o apagada.

Método cambiar_estado() que alterna el estado.

Método estado() que devuelve "encendida" o "apagada".

Monedero curioso

Clase Monedero que:

Inicializa con un saldo.

Métodos agregar(monto) y gastar(monto) con mensajes divertidos si no hay suficiente dinero.

Método saldo() que devuelve el saldo actual.

Contador de clics

Clase Contador que:

Inicializa en 0.

Método clic() que incrementa el contador y lo imprime.

Método reiniciar() que vuelve a 0.

Opción: agregar un contador total de todos los objetos creados.

Playlist musical

Clase Reproductor que:

Recibe una lista de canciones.

Método siguiente() que pasa a la próxima canción y vuelve al inicio si llega al final.

Método actual() que imprime la canción que está sonando.



Ejercicios: POO básicos

Fecha mágica

Clase Fecha que:

Inicializa con día, mes y año.

@classmethod desde_texto(cls, texto) que crea una instancia desde un string "dd/mm/yyyy".

Método mostrar() que imprime la fecha.

Robot parlante

Clase Robot que:

Inicializa con un nombre y un tono ("alegre", "serio").

Método saludar() que imprime un mensaje distinto según el tono.

Método cambiar_tono(nuevo_tono).

Semilla que crece

Clase Planta que:

Inicializa con tamaño 0.

Método regar() que aumenta el tamaño y imprime un mensaje: "La planta creció a X cm".

Método estado() que describe si es "pequeña", "mediana" o "grande".

10 Cajero automático mini

Clase Cajero que:

Inicializa con un saldo.

Método depositar(monto).

Método retirar(monto) que verifica si alcanza el saldo.

Método mostrar_saldo().

Bonus: agregar @classmethod que crea un cajero con saldo inicial por defecto