

Práctica de Clases, objetos, errores y excepciones

Ejercicio 1. Dada la clase

```
class Maquina:
    nombre = ''
    id = ''
    fecha='' # fecha de puesta en marcha
```

Define los métodos:

- parar, que emitirá el mensaje 'ZZZZzzzzzzzzzz'
- marchar, que emitirá el mensaje 'blablablaaaaa'
- contar, que emitirá el mensaje '1,2,3,4,...' (enteros positivos, mayores a 0, hasta el tope que le indiques)
- emitir, que emitirá el nombre dado a la máquina y la edad en función de la fecha de puesta en marcha.

El programa deberá pedir al usuario:

- si desea crear una máquina, en el caso de que su respuesta sea afirmativa, instanciar el objeto.
- que le dé un nombre, id, fecha de puesta en marcha.
- que le indique si quiere que emita su nombre.
- que le indique si quiere que lo ponga en marcha.
- que le indique si quiere que lo pare.
- que le indique si quiere escucharlo contar pero debe decirle hasta qué número.

Ejercicio 2.

Crea la clase Actividad. Esta clase hereda de máquina.

- Define los atributos: nombre y edad.
- Redefine el método emitir para que emita todos los datos.
- Puedes definir una o más actividades o trabajos para la máquina?. Si es así define al menos una.

Ejercicio 3.

Crea la clase Maquinita. Esta clase hereda de Maquina y Actividad

- Define los atributos: nombre y edad # según la fecha de puesta en marcha
- Define los métodos avanzar y retroceder.
- Redefine el método emitir para ésta clase.
- Si la edad es superior a 2 años puede tener una actividad y emitirla, en caso contrario debe emitir una excepción: 'No tiene actividad, todavía está aprendiendo!' (ver pdf errores, excepciones....)

Ejercicio 4.

Crear la clase Superheroe.

- El constructor debe recibir, nombre edad, estado (sin poder, con poder, representado por True o False) y una cantidad (número) máxima de poder.
- Método usar_poder, donde se irá restando el poder utilizado cuando trabaje de superhéroe, si se intenta usar más de lo que tiene como poder, debe emitir una excepción que diga 'Me quedé sin poder!!' buahhhh'.
- Cuando el poder se agote debes revivir al superhéroe.
- Método tener_poder, donde recibirá un valor que determinará la cantidad de poder a reutilizar, si supera la cantidad máxima del atributo debe emitir una excepción que diga 'Desbordamiento de poder!!', este desbordamiento se penalizará y la diferencia se restará al máximo permitido.
- Método revivir, reinicia al superhéroe con poder necesario para pedir tener_poder.
- Método volar, para activarlo debe tener al menos el 20% de poder y cada minuto que vuele, le resta 2% de poder.
- Método perforar, para activarlo debe tener al menos el 20% de poder y cada minuto que perfore, le resta 5% de poder.

Ejercicio 5.

Crear la clase Mate que describa el funcionamiento de la bebida.

- El constructor debe recibir como parámetro n, la cantidad máxima de cebadas en base a la cantidad de yerba vertida en el recipiente que también debe ser un parámetro.
- Un atributo para la cantidad de cebadas restantes hasta que se lava el mate (representada por un número).
- Un atributo para el estado (lleno o vacío).
- Un método cegar, que llena el mate con agua. Si se intenta cegar con el mate lleno, se debe lanzar una excepción que imprima el mensaje: Uhh! Te quemaste!
- Un método beber, que vacía el mate y le resta una cebada disponible. Si se intenta beber un mate vacío se debe lanzar una excepción que imprima el mensaje: El mate está vacío!...;