

Práctica de Clases y objetos

Ejercicio 1. Dada la clase

class Robot:

 nombre = ""

 id = ""

 fecha="" # fecha de puesta en marcha

- a. Define los métodos:
 - dormir, que emitirá el mensaje 'ZZZZzzzzzzzzzz'
 - hablar, que emitirá el mensaje 'bla bla blaaaaa'
 - contar, que emitirá el mensaje '1,2,3,4,...' (números enteros positivos, mayores a 0, hasta el tope que le indiques)
 - emitir, que emitirá el nombre dado al Robot y la edad en función de la fecha de puesta en marcha.
- b. El programa deberá pedir al usuario:
 - a. si desea crear un Robot, en el caso de que su respuesta sea afirmativa, instanciamos el objeto.
 - b. que le dé un nombre, id, fecha de puesta en marcha.
 - c. que le indique si quiere que emita su nombre.
 - d. que le indique si quiere que hable.
 - e. que le indique si quiere que vaya a dormir.
 - f. que le indique si quiere escucharlo contar pero debe decirle hasta qué número.

Ejercicio 2.

- a. Crea la clase Actividad. Esta clase hereda de Robot.
- b. Define los atributos: nombre y edad.
- c. Redefine el método emitir para que emita todos los datos.
- d. Puedes definir una o más actividades o trabajos para el robot?. Si es así define al menos uno.

Ejercicio 3.

- a. Crea la clase HijoDeRobot. Esta clase hereda de Robot y Actividad
- b. Define los atributos: nombre y edad # según la fecha de puesta en marcha
- c. Redefine el método emitir para que emita los datos del hijo
- d. Si la edad es superior a 3 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.

- a. En base a los ejercicios anteriores crea la clase Superheroe.
 - a. El constructor debe recibir, nombre edad, estado (sin poder, con poder, representado por True o False) y una cantidad (número) máxima de poder.
 - b. Método usar_poder, donde se irá restando el poder utilizado cuando trabaje de superhéroe, si se intenta usar más de lo que tiene debe emitir una excepción que diga 'Me quedé sin poder!! buahhhh'.
 - c. Cuando el poder se agote debes revivir al superhéroe.
 - d. Método tener_poder, donde recibirá un valor que determinará la cantidad de poder a utilizar, si supera la cantidad máxima del atributo debe emitir una excepción que diga 'Ya tengo todo el poder!! Jaja! lupiiii!!'
 - e. Método revivir, reinicia al superhéroe con poder.
 - f. En base a lo anterior crea la subclase Superheroeargentino, agrega los atributos que te parezcan convenientes y define (o redefine) los métodos apropiados según el Superheroeargentino que imagines.

Ejercicio 5.

- 5.1. Elige uno de los ejercicios de la práctica de numpy y crea las clases y métodos necesarios.

- 5.2. Elige uno de los ejercicios de la práctica de pandas y crea las clases y métodos necesarios.
- 5.3. Elige uno de los ejercicios de la práctica de networkx y crea las clases y métodos necesarios.

Ejercicios Opcionales

6. Crear una clase Vector, que en su constructor reciba una lista de elementos.
- a. En el método `__str__` se emite su contenido con el formato `[x,y,z]`.
 - b. Crear un método escalar que reciba un número y devuelva un nuevo vector, con los elementos multiplicados por ese número.
 - c. Crear un método sumar que recibe otro vector, verifica si tienen la misma cantidad de elementos y devuelve un nuevo vector con la suma de ambos. Si no tienen la misma cantidad de elementos debe dar una excepción.
7. Crear la clase Mate que describa el funcionamiento de la bebida.
- 7.1. El constructor debe recibir como parámetro `n`, la cantidad máxima de cebadas en base a la cantidad de yerba vertida en el recipiente.
 - 7.2. Un atributo para la cantidad de cebadas restantes hasta que se lava el mate (representada por un número).
 - 7.3. Un atributo para el estado (lleno o vacío).
 - 7.4. Un método `cebar`, que llena el mate con agua. Si se intenta cebar con el mate lleno, se debe lanzar una excepción que imprima el mensaje: `Uhhhhh! Te quemaste!`
 - 7.5. 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!....;`
 - 7.6. Es posible seguir cebando y bebiendo el mate aunque no haya cebadas disponibles. En ese caso la cantidad de cebadas restantes se mantendrá en 0, y cada vez que se intente beber se debe imprimir un mensaje de aviso: `Advertencia: el mate está lavado!, pero no se debe emitir una excepción.`