

Guía Práctica N°2

Profundizando en Programación Orientada a Objetos (POO)

1. Fracciones:

- Crear una clase Fracción, que cuente con dos atributos: dividendo y divisor, que se asignan en el constructor, y se imprimen como X/Y en el método `__str__`.
- Crear un método sumar que recibe otra fracción y devuelve una nueva fracción con la suma de ambas.
- Crear un método multiplicar que recibe otra fracción y devuelve una nueva fracción con el producto de ambas.
- Crear un método simplificar que modifica la fracción actual de forma que los valores del dividendo y divisor sean los menores posibles.

2. Vectores:

- Crear una clase Vector, que en su constructor reciba una lista de elementos que serán sus coordenadas. En el método `__str__` se imprime su contenido con el formato `[x,y,z]`.
- Crear un método escalar que reciba un número y devuelva un nuevo vector, con los elementos multiplicados por ese número.
- 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 levantar una excepción.

3. Botella y Sacacorchos:

- Escribir una clase Corcho, que contenga un atributo bodega (cadena con el nombre de la bodega).
- Escribir una clase Botella que contenga un atributo corcho con una referencia al corcho que la tapa, o **None** si está destapada.
- Escribir una clase Sacacorchos que tenga un método destapar que le reciba una botella, le saque el corcho y se guarde una referencia al corcho sacado. Debe lanzar un mensaje en el caso en que la botella ya esté destapada, o si el sacacorchos ya contiene un corcho.
- Agregar un método limpiar, que saque el corcho del sacacorchos, o lance un mensaje en el caso en el que no haya un corcho.

4. Modelar una clase Mate que describa el funcionamiento de la conocida bebida tradicional argentina. La clase debe contener como miembros:

- 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).
- El constructor debe recibir como parámetro n, la cantidad máxima de cebadas en base a la cantidad de yerba vertida en el recipiente.
- 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 *¡Cuidado! ¡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 un mensaje que imprima: *¡El mate está vacío!*
- 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.*

5. Juego de Rol:

- Escribir una clase Personaje que contenga los atributos vida, posición y velocidad, y los métodos recibir_ataque, que reduzca la vida según una cantidad recibida y lance un mensaje si la vida pasa a ser menor o igual que cero, y mover que reciba una dirección y se mueva en esa dirección la cantidad indicada por velocidad.
- Escribir una clase Soldado que herede de Personaje, y agregue el atributo ataque y el método atacar, que reciba otro personaje, al que le debe hacer el daño indicado por el atributo ataque.
- Escribir una clase Campesino que herede de Personaje, y agregue el atributo cosecha y el método cosechar, que devuelva la cantidad cosechada.