Respuestas:

Pregunta a: 3 clases

Pregunta b: Para poder ejecutar las clases.

Pregunta c: El código funciona normalmente porque las clases se ejecutan correctamente en este caso y no lanza error porque esa línea se usa para ejecutar el código que se desea y no las clases al ser importadas.

Pregunta d: 2 objetos

Pregunta e: apostador1 y apostador2

Pregunta f: Está haciendo referencia al objeto que ejecuta el método

Pregunta g: Con el cambio en la línea 4 se crea 1 objeto de la clase Loteria, sin el cambio se crean 2

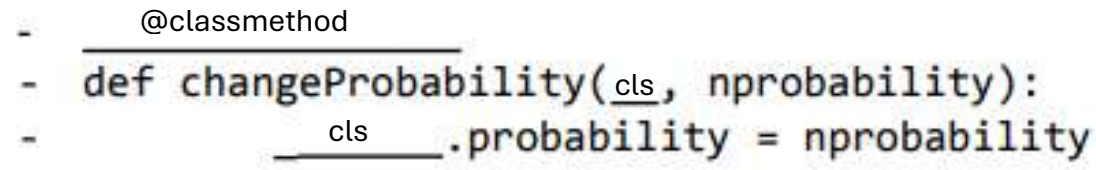
Pregunta h: El código sigue ejecutando el juego de azar, porque las modificaciones en el apostador2 no afectan al apostador1, el código sigue funcionando correctamente

Pregunta i: Imprime el numero en wallet, cuanto gana o pierde, y el dinero que tiene luego de la jugada

Pregunta j: self.apostador

Pregunta k: self.value y probability

Pregunta l:



Pregunta m: Loteria.changeProbability(nprobability)

Pregunta n: No es correcto porque rompe con la convención que está establecida, es decir, estaríamos incurriendo en una mala práctica.

Pregunta o: 4 métodos

Pregunta p: cuando no ganan ambos es porque son objetos diferentes y por lo tanto los métodos que ejecutan son distintos, si sí ganaran ambos, sería por coincidencia o porque ejecutamos el mismo método para ambos objetos.

Pregunta q: Como las constantes no existen en Python, el código funcionaría, pero si la establecimos como tal(en mayúsculas), no debería existir el método changeProbability

Pregunta r: son de tipo double

Pregunta s: Hace referencia al objeto de clase Loteria que está ejecutando ese método, no se puede omitir porque no estaríamos dando el argumento a la clase ComisionJuegoEspectaculos

Pregunta t:value es pasado por valor porque es inmutable y self por referencia porque es mutable