

Desarrollo de Preguntas. Taller2-Python Julián Bedoya Palacio-jubedoyap

- a. Se están creando 3 clases
- b. Para asegurarse de que el código del módulo main solo se va a ejecutar dentro de sí mismo, en caso de que se ejecute desde otro módulo no se obtiene nada.
- c. Si retiro dicha línea el archivo importado podría ejecutar su contenido sin desearlo, en cambio con ella me permito importar solo las funciones y clases de ese archivo.
- d. Se crean 2 objetos de la clase apostador
- e. apostador1 y apostador2
- f. Ese self hace referencia a apostador1
- g. Se crean 2 objetos de la clase lotería
- h. Como lo que se ingresa a la wallet (300) es menor que la cantidad con la que se quiere jugar (400) el código imprime "Necesitas poner mas dinero en tu wallet."
- i. Como el valor que se ingresa a la wallet (400) es el mismo con el que se juega (400) (cumpliendo la restricción de self.wallet >= value), podría ocurrir que gane, he imprima un mensaje del tipo "has ganado str(total)" o puede perder e imprime "Has perdido lo que apostaste".
- j. apostador
- k. value y probability
- l. @classmethod

def changeProbability(cls, nprobability):

 cls.probability = nprobability
- m. Loteria.changeProbability(nprobability);
- n. Sí es correcto, porque cls hace referencia a la clase en la que se trabaja, que en este caso es Loteria, por lo tanto hacer cls.probability será lo mismo que hacer Loteria.probability
- o. La clase lotería queda con 5 métodos.
- p. Lo que pasa con un apostador u otro es independiente, es decir si uno gana el otro puede ganar o perder, esto es porque se está jugando con una especie de "ruleta" cuando se hace uso de la librería random dentro de la clase lotería; se puede ver en la línea 16 (a = random.randint(0,1)) y hay dos opciones que lo pueden hacer ganar o perder.
- q. De hecho el cambio es simbólico, es decir, en Python no se pueden crear constantes, la forma para identificarla es escribir su nombre en Mayúscula; por lo tanto, el uso del método changeProbability seguirá funcionando pero no sería algo teóricamente correcto.
- r. Ambos métodos retornan valores de tipo float

s. Está haciendo referencia a un objeto de la clase Lotería y no podría omitirse porque el constructor de la clase ComisionJuegosEspectaculos requiere un argumento explícito cuando es invocado.

t. el parámetro value pasa por referencia al ser un elemento primitivo e inmutable, mientras que self representa un objeto de la clase Apostador y por ser mutable pasa por referencia.