a. se están definiendo 3 clases
b. lo que hace esta linea de codigo es prácticamente preguntar si el archivo se está ejecutando directamente o está siendo importado, evitando así que si el archivo es importado se ejecute más cosas de las que se necesitan
c. Si se elimina la linea if __name__ = “__main __” del codigo, el codigo sigue funcionando normalmente, el problema viene si queremos llegar a importar este archivo donde debería estar el if __name__ = “__main __”, ya que al no estar dicha línea provoca que se ejecute todo al importarlo lo que podría no ser lo que se busca
d. se están creando 2 objetos de la clase Apostador
e. se están creando los objetos apostador1 y apostador2
f. hace referencia a una instancia de la clase apostador
g. se crean 2 objetos de la clase loteria
h. 300.
   Necesitas poner más dinero en tu wallet.
   300.
i. 400.
   Aquí no sabría que se imprimiría ya que se corre la función playgame la que da un resultado aleatorio entre “Has ganado X” y “Has perdido lo que apostaste”.
   Aquí nuevamente no se que se imprime ya que depende si gano o no, en dado caso de que haya ganado se imprimirá el valor de X que gano y en caso que no se imprime 0.
j. solo el atributo self.apostador
k. self.value y self.probability
l. -@classmethod
   -def changeProbability(cls, nprobability):
   - cls.probability = nprobability
m. un ejemplo de como seria la linea de codigo para llamar el metodo changeProbability seria Loteria.changeProbability(x), donde x sería la nueva probabilidad que queramos usar por ejemplo 0.8
n. si es correcto y sigue funcionando, el problema está en que si se llega a cambiar el nombre de la clase no funciona y tocaría modificar el nombre de la clase a la que hace referencia
o. 5 métodos
p. no, si el apostador1 gana, el apostador2 no necesariamente tiene que ganar también, al ser diferentes objetos se ejecuta por separado por lo que tienen diferentes resultados como dos apuestas distintas, por lo que puede que uno gane y el otro no, que ambos ganen o que ambos pierdan.
q. si se decide cambiar probability a una constante se escribirá en mayúsculas ya que en python no hay una manera de declarar o establecer constantes y el método changeProbability perdería su propósito ya que al ser probability una constante no debería modificarse su valor, por lo que sería inadecuado su uso
r. ambos métodos tienen un tipo de retorno float
s. la variable self está haciendo referencia a la instancia de la clase loteria, osea hace referencia al apostador y el valor apostado que se creó en esa instancia, no puede omitirse ya que sin esto ComisionJuegoEspectaculos no podria acceder a estos datos.
t. value paso por valor ya que es un tipo de dato primitivo y self pasa por referencia ya que hace referencia a una instancia de la clase.