

Respuestas

- a) 3 clases
- b) Sirve como script principal o clase main.
- c) Sin esa línea el código puede funcionar, pero es menos controlado, ya que ejecutará todo independientemente de si se ejecuta o se importa.
Con esa línea se asegura un comportamiento más controlado, donde el código específico se ejecuta solo cuando el archivo se corre directamente, sin afectar al ser importado.
- d) 2 objetos.
- e) "apostador1" y "apostador2".
- f) A "apostador1"
- g) 1 objeto
- h) Necesitas poner mas dinero en tu wallet
- i) Puede imprimir varias cosas como: Ganaste 720, Perdiste todo tu dinero, etc.
- j) Self.apostador
- k) Self.value
- l) @classmethod
def changeProbability(cls, nprobability):
 cls.probability = nprobability
- m) Loteria.changeProbability(nprobability)
- n) Sí, es correcto, pero cls.probability = nprobability es mejor práctica. Usar cls permite que el método funcione en subclases, mientras que Loteria.probability = nprobability solo afecta a Loteria, sin adaptarse a subclases.
- o) 4 métodos
- p) No, ya que los métodos que usan para jugar, los apostadores, son aleatorios, cada llamado es diferente ya que no comparten atributos de clase.
- q) Si probability se define como constante, elimina o evita usar changeProbability, ya que las constantes no deberían modificarse.
- r) Los dos retornan floats.
- s) Hace referencia al objeto lotería que se creó en la clase play de apostador. Ese necesario el self ya que es un parámetro necesario de la clase "ComisionJuegoEspectaculos", el cual es un objeto lotería referenciado precisamente con ese self.
- t) El parámetro value pasa por valor porque recibe un int que es un primitivo. El parámetro self pasa por referencia porque hace alusión a un objeto apostador.