Preguntas de análisis

a) ¿Cuántas clases se están definiendo en este ejercicio?

R/: Se están definiendo 3 clases Aportador, Loteria, ComisionJuegoEspectaculos.

b) ¿Para qué sirve la línea de código **if __name = “ _main_”**:?

R/: Con este se da a entender a partir de donde se va a ejecutar el programa principal donde se dan todas las ordenes para el correcto flujo del mismo.

c) ¿Qué sucede si retiro la línea de la pregunta anterior en nuestro código?, ¿Este sigue corriendo o hay error? Explique en ambos casos.

R/: El código funcionara, pero además hay que organizar la indentación, y el programa quedara funcionando de dos maneras importándolo desde otro archivo o corriéndolo desde el mismo y esto puede causar el mal flujo del programa

d) ¿Cuántos objetos de la clase **Apostador** se están creando?

R/: Se crean 2 objetos de clase Apostador

e) ¿Cuáles objetos de la clase **Apostador** se están creando?

R/: Se están creando los objetos apostador1 y apostador2

f) ¿A quién está haciendo referencia la variable **self** de la línea **15** de la clase **Apostador** cuando se ejecuta el programa principal?

R/: Se refiere al indicador en cuestión, en el caso de la línea 6 en main.py se refiere al apostador1 y en la línea 12 hará referencia a el apostador2

g) ¿Cuántos objetos de la clase **Lotería** se están creando?

R/: Se están creando dos en **Apostador.py** uno por cada apostador

- En la línea **4** del **main.py** cambiar el apostador1.deposit(500) por apostador1.deposit(300)

h) ¿Qué imprimiría el código por parte del **apostador1**?

R/:

300
Necesitas poner más dinero en tu wallet
300

- En la línea **10** del **main.py** cambiar el apostador2.deposit(500) por apostador2.deposit(400)

i) ¿Qué imprimiría el código por parte del **apostador2**?

R/:

400

Puede imprimir una de estas dos formas ya que es random Has ganado 720.0 o Has perdido lo que apostaste

Puede imprimir una de estas dos formas 720.0 o 0

j) ¿Cuáles atributos de la clase **Lotería** están haciendo referencia a objetos?

R/: El atributo apostador haciendo referencia a el objeto apostador

k) ¿Cuáles atributos de la clase **Lotería** están haciendo referencia a tipos primitivos?

R/: El atributo value, probability

l) ¿Complete las siguientes líneas para que en la clase **Loteria**, se implemente el método de clase changeProbability?

- **@classmethod**
- def changeProbability(**cls**, nprobability):
  cls.probability = nprobability

m) ¿Cómo sería la línea de código para llamar el método changeProbability?

R/: **Loteria.changeProbability(**número al que se quiere cambiar el atributo probability**)**

n) ¿Es correcto si en el método changeProbability que se creó, cambiar lo siguiente? Explique:

Línea Original
- cls.probability = nprobability

Línea Nueva
- Loteria.probability = nprobability

R/: Si, porque se puede llamar al atributo, a través de otro atributo o desde la misma clase

o) ¿Cuántos métodos tiene la clase **Loteria** después de agregarle el nuevo método?

R/: Quedaría con un total de 4 métodos: changeProbability, payMoney, recieveMoney, playGame

p) ¿Si el **apostador1** gana el **apostador2** también? Explique por qué pasa en caso de ser sí o no

R/: Ambos objetos tienen la oportunidad de ganar o perder independiente uno del otro ya que la victoria o derrota dependen de una probabilidad de 0.5

q) ¿Qué sucede si decido cambiar el atributo de clase probability a una constante? ¿Se considera correcto el uso del método changeProbability teniendo en cuanta este nuevo cambio?

R/: Las constantes en Python no existen la única manera de referenciarlas es poner el atributo en mayúscula, por lo tanto, no sucede nada, pero no se considera correcto el uso del método.

r) ¿Cuál es el tipo de retorno de los métodos gain() y commission() de la clase **ComisionJuegoEspectaculos**?

R/: El tipo de retorno es float para ambos metodos

s) ¿A quién está haciendo referencia la variable **self** de la línea **18** de la clase **Loteria** cuando se ejecuta el programa principal? ¿Podría omitirse el uso de la variable **self** en este caso?

R/: Hace referencia al objeto lotería, y no puede omitirse por que no se crearía objeto de clase ComisionJuegoEspectaculo, porque necesita un atributo de referencia para crearse y además a esto saca error en la ejecución.

t) ¿En la línea **15** de la clase apostador vemos como la clase recibe dos parámetros (value, self) especificar cuál de estos pasa por valor y cuál por referencia y por qué?

R/: El parámetro value pasa por valor ya que trae directamente el atributo con el valor que realizara la apuesta y self por referencia porque trae todos los atributos del apostador que aposto.