Taller Python

Tomas Velasquez Eusse

Respuestas

A. En el ejercicio se definen 3 clases: Apostador, ComisionJuegoEspectaculos y Loteria

B. El if **__name__** == "__ main __" sirve para saber si el archivo se ejecuta como un script o si es importado como un módulo

C. Si retiramos if **__name__** == "__ main __", su bloque de código solo se ejecutara cuando el archivo main se importe como módulo en otro archivo. No generaría un error directamente, pero puede causar comportamientos del programas que no deseamos.

D. Se crean 2 instancias de Apostador.

E. Las instancias de apostador que se crean son: apostador1 y apostador2.

F. El self hace referencia a la instancia actual de la clase Apostador en la que se está ejecutando el metodo play(). por ejemplo cuando en main se usa apostador1.play(), hace referencia a apostador1.

G. Se crean dos instancias de Loteria, cuando se llama al metodo play(), específicamente en las líneas 6 y 12 de main.py.

H. Al hacer el cambio, imprime: 300 y "Necesitas poner más dinero en tu wallet".

I. Al hacer el cambio, imprime 400.

J. En la clase Loteria sólo al atributo apostador hace referencia a un objeto de clase Apostador.

K. En la clase Lotería hay dos atributos que hacen referencia a tipos primitivos: value a un valor numérico y probability a un float (0.5).

L. Para hacer el metodo tendríamos que hacerlo así:

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

M. Para llamar este método haríamos por ejemplo: loteria.changeProbability(0.7)

N. No sería correcto cambiar cls por self, ya que modificamos el atributo probability, el cual es un atributo de clase, y cls es la forma correcta de hacerlo.

O. Ahora la clase Loteria cuenta con 5 metodos: __init__ , payMoney, recieveMoney, playGame, changeProbability.

P. No, que el apostador1 gane no afecta en el resultado del apostador2, ya que cada instancia es independiente de la otra. Además, cada instancia de Lotería que se crea al usar el método play() son independientes la una de la otra.

Q. Si cambiamos el atributo de clase probability a una constante, no sería muy adecuado crear un método changeProbabilty(), ya que por definición una constante tiene la intención de no sufrir cambios ya que fue definida.

R. Ambos métodos, gain() y commission() retornan valores de tipo float.

S. En la línea 18 de Loteria, self hace referencia a la instancia de Loteria la cual ejecuta el método playGame(). No se puede omitir este self ya que este pasa al constructor de ComisionJuegoEspectaculos para que pueda acceder a los atributos y métodos de la instancia en cuestión.

T. El parámetro value pasa por valor, ya que el propio python pasa por valor todos los datos inmutables como enteros, flotantes, cadenas y tuplas.
Por otro lado, el self pasa por referencia, ya que python pasa todas las instancias de clases como referencia.