

Solucion

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

R// Se están definiendo 4 clases

b) ¿Para qué sirve la línea de código `if name == "__main__":`?

R// Sirve para verificar que el metodo se este ejecutando dentro del programa principal y no desde otros modulos

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// En este caso no causa mayor diferencia, ya que al importar apostador, el bloque de codigo se ejecutara inmediatamente, y si exportas main.py se seguira ejecutando igual y no seria restringido

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

R// se están creando 2 objetos de tipo Apostador

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

R// 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// Depende, en la linea 6 hace referencia a apostador1 y en la linea 12 hace referencia a apostador2

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

R// Se crea un objeto por cada vez que se ejecute el metodo play, por tanto al ejecutar el codigo tendremos dos objetos de tipo loteria, uno para apostador1, y otro para apostador2

- 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// Se imprimirá "Necesitas poner mas dinero en tu wallet" ya que queremos jugar 400, cuando solo tenemos 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// Inicialmente ejcutaria el metodo playgame, dependiendo de si gane o pierda imprimira el mensaje correspondiente

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

R// Solamente hay un atributo que referencie un objeto, es `self.apostador`

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

R// ya que no están definidos que tipo de datos son, pero viendo cómo se comportan, podemos decir que `self.value` y `probability` son de tipo primitivo

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("algún valor")

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// Como tal no afecta la funcionalidad del código y seguiría cumpliendo su tarea

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

R// Tendría 5 métodos

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

R// No, esto debido a que el objeto apostador1 y apostador2 son completamente diferentes y no hay nada que los una, así que cada proceso será único para cada objeto, y depende de la suerte que uno u otro gane

q) ¿Qué sucede si decido cambiar el atributo de clase probability a una constante?

¿Se considera correcto el uso del método changeProbability teniendo en cuenta este nuevo cambio?

R// Cambio a constante: Deberías definir probability como una constante, usando mayúsculas, y no permitir que su valor cambie una vez asignado.

Uso del método changeProbability: No sería correcto tener un método changeProbability si probability es una constante, ya que las constantes, por definición, no deben cambiar.

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

R// Ambos métodos (gain() y commission()) retornan valores float.

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// la variable self hace referencia a la instancia actual de la clase Loteria. Específicamente, es la instancia del objeto Loteria que está ejecutando el método playGame(). No podría omitirse porque ComisionJuegoEspectaculos necesita saber de qué instancia de Loteria se trata para poder realizar los cálculos correctos.

- 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// value pasa por valor porque es un tipo inmutable (un número), y cualquier cambio no afectará a la variable fuera de la función.

self pasa por referencia porque es un objeto mutable, por lo que la clase Loteria puede acceder y modificar los atributos del objeto Apostador que se le pasa.