

Valentina Sierra Durango

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

Se están definiendo cuatro clases: Apostador, Loteria, ComisionJuegoEspectaculos, y una clase implícita del módulo random.

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

Esta línea verifica si el archivo está siendo ejecutado como un programa principal. Si es así, se ejecuta el bloque de código bajo esta condición; si el archivo se importa como módulo en otro script, este bloque no se ejecutará.

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

El código sigue corriendo sin error si la línea se elimina, pero el bloque de pruebas en `if __name__ == "__main__":` se ejecutaría cada vez que el archivo se importe, no solo cuando se ejecute directamente.

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

Se crean dos objetos de la clase Apostador: `apostador1` y `apostador2`.

e) ¿Cuáles objetos de la clase Apostador 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?

`self` hace referencia al objeto `apostador1` o `apostador2`, según el objeto que llame al método en ese momento.

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

Se crea un objeto Loteria cada vez que se llama al método `play` en un objeto de Apostador, por lo que se crean dos objetos Loteria, uno para cada apostador.

h) ¿Qué imprimiría el código por parte del `apostador1` si se cambia `apostador1.deposit(500)` por `apostador1.deposit(300)`?

Se imprimirá el saldo inicial de apostador1 como 300 antes de jugar, y después de jugar, dependerá de si gana o pierde (puede mostrar un saldo reducido o incrementado).

i) **¿Qué imprimiría el código por parte del apostador2 si se cambia apostador2.deposit(500) por apostador2.deposit(400)?**

Similar a la respuesta anterior, el saldo inicial de apostador2 sería 400 y cambiaría dependiendo de si gana o pierde.

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

El atributo apostador hace referencia a un objeto de la clase Apostador.

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

El atributo value y probability son de tipo primitivo (float para probability y int para value).

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?**

```
Loteria.changeProbability(0.7)
```

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

Ambas líneas son correctas. cls.probability y Loteria.probability acceden al mismo atributo de clase, pero cls permite flexibilidad en caso de herencia.

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

Cuatro métodos: __init__, payMoney, recieveMoney, playGame, y ahora changeProbability.

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

No, cada jugador juega de forma independiente. Los resultados son aleatorios y no dependen de las partidas de otros jugadores.

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?**

Si probability es una constante, changeProbability no debería usarse ya que estaría en contra del concepto de inmutabilidad de una constante.

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

Ambos métodos retornan un valor de tipo float, ya que calculan un valor de ganancia basado en el porcentaje.

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?**

self en esta línea hace referencia al objeto Loteria actual que está en juego. No puede omitirse porque es necesario para acceder a los atributos y métodos de la instancia.

t) **En la línea 15 de la clase Apostador vemos cómo la clase recibe dos parámetros (value, self) especificar cuál de estos pasa por valor y cuál por referencia y por qué.**

value pasa por valor porque es un entero inmutable, mientras que self pasa por referencia, permitiendo que el objeto se modifique dentro de los métodos de la clase.