

Parte 2 taller de Python semana 2

Andres David Calderón Jiménez

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

R: 3 clases

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

R: Para verificar que este es el programa principal, el que estamos corriendo directamente.

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: Sigue corriendo, pero correría la función aún si solo importamos código de este archivo en otro, esto podría causar errores en otros momentos.

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

R: Se crean 2.

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

R: Apostador1 y apostador2, con nombres Juan y Ricardo respectivamente.

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 a `apostador1` la primera vez que se ejecuta y a `apostador2` la segunda.

g) ¿Cuántos objetos de la clase Lotería se están creando? - En la línea 4 del `main.py` cambiar el `apostador1.deposit(500)` por `apostador1.deposit(300)`

R: Solo se crea 1 al cambiar la línea, 2 normalmente.

h) ¿Qué imprimiría el código por parte del `apostador1`? - En la línea 10 del `main.py` cambiar el `apostador2.deposit(500)` por `apostador2.deposit(400)`

R: Luego de cambiar la línea del punto anterior se imprime "Necesitas poner mas dinero en tu wallet" pues el valor de la lotería es mayor al depositado."

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

R: Luego de cambiar la línea del punto anterior no cambia nada pues sigue siendo suficiente para jugar la lotería. Lo que imprime es aleatorio y dependerá de el valor de `a` en `Loteria.playGame`.

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

R: `Apostador`, aunque no es un atributo de clase, sino de instancia.

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

R: `probability` y `value`.

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

```
def changeProbability(__, nprobability):  
    _____.probability = nprobability
```

1. @classmethod
2. cls
3. cls

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

R: Loteria.changeProbability(Loteria,nprobability)

- 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, pues para ser útil, cls ya debería ser Lotería. Si fuese una instancia, entonces solo terminaría cambiando para esa lotería en específico. Este sería un método en muchos casos equivalente y tiene más sentido este.

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

R: 5 métodos, solo 1 de clase, los otros 4 de instancia.

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

R: No. No hay relación entre que el apostador1 gane y que el apostador2 gane pues el numero que define el resultado es aleatorio, local a la función playGame, además de que cada vez que un apostador juega, usa una nueva instancia de Loteria.

- 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: Es posible, pero no es recomendable, pues si marcamos probability como constante, entonces otros programadores esperarían que nunca cambiase, y generamos confusión.

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

R: ambos retornan un primitivo float.

- r) ¿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?

Hace referencia a la lotería que esta siendo jugada. No se puede omitir, pues el init de ComisionJuegoEspectaculos necesita una referencia a la lotería.

- s) ¿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?

value pasa por valor pues es un primitivo, y self pasa por referencia pues es el objeto del apostador que esta jugando.