Taller 2 Python

Sergio Morales Martínez

a) ¿Cuántas clases se están definiendo en este ejercicio?
R// 3 clases Apostador, ComisionJuegoEspectaculos y Lotería.

b) ¿Para qué sirve la línea de código if __ name = " _main_ ":
R// Esta línea sirve para verificar si el archivo se esta ejecutando directamente o si esta siendo importado, nos ayuda a evitar que el archivo se ejecute cuando se importe como modulo.

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 sigue corriendo sin ninguna anormalidad, sin embargo, no es conveniente borrar esta línea, debido a que ayuda a prevenir errores.

d) ¿Cuántos objetos de la clase Apostador se están creando?
R// Dos objetos.

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// Self hace referencia a la instancia sobre la cual se está llamando el método.

g) ¿Cuántos objetos de la clase Loteria se están creando? - En la línea 4 del main.py cambiar el apostador1.deposit(500) por apostador1.deposit(300)
R// Dos, ya que cada vez que se llama al método play se esta generando una instancia de loteria.

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// Al cambiar el monto que se introduce al método deposit, no sería posible apostar, ya que en la línea 6 el apostador 1 intenta ingresar más dinero del que tiene, por lo que imprimiría "Necesitas poner mas dinero en tu wallet"

i) ¿Qué imprimiría el código por parte del apostador2?
R// En caso de perder la apuesta, perdería todo lo que tiene, ya que, al modificar este valor, coincidiría con la cantidad que apuesta, por lo que pondría todo en juego.

j) ¿Cuáles atributos de la clase Lotería están haciendo referencia a objetos?
R// self.apostador es el único atributo, pues que este al ser ejecutado en el código hace referencia a una instancia de la clase apostador.

k) ¿Cuáles atributos de la clase Lotería están haciendo referencia a tipos primitivos?
R// self.value y self.probability, uno es de tipo int y el otra de tipo double.

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(valor). También podría ser llamado desde una instancia, sin embargo, no es lo más optimo.

n) ¿Es correcto si en el método changeProbability que se creó, cambiar lo siguiente? Explique:
R// Si, no debería haber inconveniente al cambiarlo, puesto que es un método de clase, y se puede referir a si misma sin problema dentro del método, sin embargo, la manera más conveniente de hacerlo es con el “cls”.

o) ¿Cuántos métodos tiene la clase Loteria después de agregarle el nuevo método?
R// Tiene 5 metodos.

p) ¿Si el apostador1 gana el apostador2 también? Explique por qué pasa en caso de ser sí o no
R// No en todos los casos, sin embargo, en el código actual es posible que ambos ganen, puesto que la probabilidad es un atributo global, y se puede dar el caso de que ambos ganen puesto que estos eventos no son disjuntos, ósea son independientes.

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// No, ya que la intención de que un atributo sea constante es su inmutabilidad, por lo que, al usar un método para cambiar este valor, la constante perdería el propósito para la que fue destinada.

r) ¿Cuál es el tipo de retorno de los métodos gain() y commission() de la clase ComisionJuegoEspectaculos?
R// Ambos retornan un valor del tipo 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 línea 18 es: commi = ComisionJuegoEspectaculos(self)
En esta línea el self hace referencia a la instancia de la clase apostador sobre la cual se está efectuando el método.

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 esta pasando por valor, ya que lo que se pasa es de tipo primitivo (entero) y se estaría pasando una copia. Mientras que self si se pasa por referencia, ya que se esta pasando la referencia de un objeto, y cualquier cambio aquí, afectaría los atributos de este directamente.