Santiago Abelardo Salcedo Rodriguez
Programacion Orientada a Objetos
Taller 2 python parte 2
Grupo 1

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

R= 3 clases
b) ¿Para qué sirve la línea de código if __ name = “ _main_ ”:?

R= Es para asegurarse que el archivo si esta siendo ejecutado directamente o esta siendo importando desde otro archivo

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 habría problema con la ejecución precisamente la línea se encuentra en el código principal, si hubiera errores en otras clases que si están siendo importadas se pueden ejecutar líneas de código no deseadas al ser importadas y utilizadas

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= A la instancia en si misma , esta pasando por referencia a si mismo como objeto

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

R= 2 objetos de la clase

- 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= 300
“Necesitarias poner mas dinero en tu wallet”
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= Se ejecutaria exactamente igual que antes,
400
Su wallet si gano o perdio

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

R= el atributo apostador
k) ¿Cuáles atributos de la clase Lotería están haciendo referencia a tipos primitivos?

R= probabilidad y value

l) ¿Complete las siguientes líneas para que en la clase Loteria, se implemente el método de clase changeProbability?
R= Imagino que se quisiera implementar como método de clase
- @classmethod
- def changeProbability(cls, nprobability):
-cls.probability = nprobability

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

R= `Loteria.changeProbability(0.35)`

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= no es correcto ya que por convención en métodos de clase se utiliza el cls para referenciar que se refiere a la propia clase.
o) ¿Cuántos métodos tiene la clase Loteria después de agregarle el nuevo método?

R= 4 metodos mas el método constructor

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

R= Seria solo casualidad ya que la variable a que utiliza el método random.randint al consumirse el método es eliminada y cuando se vuelve a llamar vuelve crearse

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= Pues la estructuración del código es inconsistente, si la cambio a una constante (aunque en Python no existen las constantes) luego al ejecutar el método changeProbability y cambiar el valor dejaría de ser constante y puede afectar al buen funcionamiento del código, pierde sentido

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

R= retornan valores numéricos que interactúan con mas líneas de codigo

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= cuando se ejecuta el programa principal, el apostador 1 juega y crea un objeto Loteria que luego le “pide” que ejecute su método playGame en la línea 18 se refiere a el objeto Loteria creado, similarmente cuando el apostador2 juega crea un objeto Loteria nuevo y ocurre el mismo flujo. Y no no debe omitirse pues comisiónJuegosEspectaculos esta esperando un objeto de tipo Loteria que precisamente con el self se logra

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 al ser un tipo de objeto inmutable y self (apostador) pasa por referencia debido a que es un objeto mutable.