Respuestas parte 2 PYTHON

Preguntas de análisis

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

En este ejercicio se definen 3 clases

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

Se utiliza para inicializar el programa y saber cual es el programa principal junto con lo que se va a ejecutar

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

Si se retira sigue corriendo siempre y cuando el archivo se haya ejecutado como el principal, es decir, sirve para es ver si el módulo ha sido ejecutado directamente o no (importado). Si se ha ejecutado como programa principal se ejecuta el código dentro del condicional.

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

2 objetos de la clase Apostador

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

Hace referencia al objeto de clase Apostador que esta ejecutando el método play(), (hace referencia así mismo)

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

Antes del cambio se crean dos objetos de clase Loteria, uno por cada apostador

- En la línea 4 del main.py cambiar el apostador1.deposit(500) por apostador1.deposit(300)

Después del cambio, el "wallet" de apostador1 es menor al "value", es decir no tiene suficiente dinero para apostar, por lo que no se crea un objeto Loteria, pero para el apostador2 si se llega a crear el objeto lotería, por lo que solo se crearía 1 objeto lotería

h) ¿Qué imprimiría el código por parte del apostador1?

Después del cambio del inciso g, imprime: "Necesitas poner más dinero en tu wallet", además del valor atributo "wallet" del objeto apostador1 (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?

Imprime si ha ganado o no la apuesta y cuanto ha ganado, siempre con el valor actualizado de su "wallet"

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

El atributo apostador es de tipo Apostador, aunque suene redundante

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

El atributo value es 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?

Al ser metodo de clase se puede invocar desde la clase o desde los objetos. Ejemplo:

Loteria.changeProbability(nprobablity) Clase

loteria.changeProbability(nprobablity) Objeto

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

Explique:

👩‍💻 PROGRAMACIÓN ORIENTADA A OBJETOS - UNALMED 2024 -2 👩‍💻

Línea Original

- cls.probability = nprobability

Línea Nueva

- Loteria.probability = nprobability

Por notación no seria correcto, ya que entre los paréntesis de los parámetros solo esta cls y nprobability, aunque se refiere a la misma clase, asi que no hay problema

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

Tiene 3 metodos

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

No realmente, es algo que puede suceder pero ya que depende a un numero a (class Loteria, linea16), que es aleatorio (random), que gane un aposatdor no significa que gane el otro

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?

Aunque en Python no existen las constantes, si realmente se respetaran, entonces el método changeProbability no se consideraria, puesto que pretende cambiar el valor de una constante

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

El retorno de ambos métodos es de tipo primitivo, de hecho, son valores numéricos

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?

Es para referirse asi mismo, para ser un atributo de clase loteria del nuevo objeto commi de tipo ComsionJuegosEspectaculos, asi que no podría omitirse

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?

value al ser un valor numérico es de tipo inmutable asi que pasa por valor, en cambio self, que se refiere a un objeto perteneciente a una clase previamente definida es de tipo mutable, por lo que pasa por referencia