Andrés Felipe Guerra Amaris
cc 1097783416
anguerraa@unal.edu.co

Solución preguntas de codificación y análisis

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

Respuesta: 3, Apostador, Loteria y ComisionJuegoEspectaculos.

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

Respuesta: Para definir el bloque de código que se ejecutará cuando se ejecute el programa y a partir del cual se llaman las demás clases y métodos definidas.

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

Respuesta: sin esa linea de código el codigo seguirá corriendo siempre que se ejecute el módulo main.py directamente. Igualmente lo hará si se importa este módulo a alguno de los otros del código, lo que puede causar problemas.

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

Respuesta: dos.

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

Respuesta: 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?

Respuesta: el self de la linea 15 de clase apostador hace referencia a la instancia desde la cual se está ejecutando el método deposit o play. En el programa principal, cuando se ejecuta apostador1.deposit(500), self hace referencia a la instancia apostador1 de la clase Apostador. De manera similar, cuando se ejecuta apostador2.deposit(500), self hace referencia a la instancia apostador2.

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

Respuesta: Se crean dos, uno por cada llamada al metodo play de la clase apostador.

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

Respuesta: tras la modificación, se crea solo un objeto de la clase Loteria, ya que el apostador1 necesita poner más dinero en su wallet, uno de los objetos de loteria no es creado. el output es: 300

Necesitas poner mas dinero en tu wallet

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

Respuesta: hay dos posibilidades, o gana y queda la wallet con 720.0, o pierde todo y queda en 0.

400
Has perdido lo que apostaste
0
o bien:
400
Has ganado 720.0
720.0

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

Respuesta: el atributo apostador y el self.

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

Andrés Felipe Guerra Amaris
cc 1097783416
anguerraa@unal.edu.co

Respuesta: el atributo value.

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

Respuesta:

```
@classmethod
def changeProbability(__cls__, nprobability):
    cls.probability=nprobability
```

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

Respuesta:

Loteria.changeProbability(nprobability).

donde nprobability es el valor que se desea asignar a la probabilidad.

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

Respuesta: no es correcto o incorrecto, daría el mismo resultado, ya que de ambas maneras se está haciendo referencia a la clase Loteria.

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

Respuesta: 5, el constructor, payMoney, receiveMoney, playGame y changeProbability.

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

Respuesta: no, ya que apostador1 y apostador2 son dos instancias diferentes de la clase Apostador, con su propia wallet e instancia de Loteria, con cada juego siendo aleatorio e individual. Por lo que si uno gana o no no afecta al resultado del 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?

Respuesta: no se consideraría correcto, ya que al declarar un atributo como constante, significa que su valor no debería ser cambiado, por tanto, el metodo para cambiar la probabilidad no seria correcto.

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

Respuesta: de tipo flotante o float.

s) ¿A quién está haciendo referencia la variable self de la línea 18 de la clase Lotería cuando se ejecuta el programa principal? ¿Podría omitirse el uso de la variable self en este caso?

Respuesta: se hace referencia a la instancia de la clase loteria generada por el método play de la clase Apostador. El self no se puede omitir aquí.

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?

Respuesta: value pasa por valor ya que puede ser de tipo int o float, en cambio self es un paso por referencia ya que representa una referencia de la instancia de la clase Apostador que llamó a método play.