

Preguntas de análisis TALLER 2 PYTHON SEMANA 2

- a) Se están definiendo 3 clases en este ejercicio: Apostador, ComisionJuegoEspectaculos y Loteria.
- b) La línea "if __name__ == '__main__':" se utiliza para asegurar que el código dentro de ella solo se ejecute cuando el script se ejecuta directamente, y no cuando se importa como un módulo. Esto es útil para tener un punto de entrada para el programa.
- c) Si se retira esa línea, el código seguirá corriendo sin problemas, ya que no hay nada que impida la ejecución del programa principal. Sin embargo, al tener esa línea, se garantiza que el código dentro de ella solo se ejecute cuando el script se ejecuta directamente.
- d) Se están creando 2 objetos de la clase Apostador.
- e) Los objetos de la clase Apostador que se están creando son apostador1 y apostador2.
- f) La variable self en la línea 15 de la clase Apostador hace referencia al objeto de la clase Apostador que está ejecutando el método init_.
- g) Se está creando 1 objeto de la clase Loteria.
- h) Si se cambia apostador1.deposit(500) por apostador1.deposit(300), el código imprimiría el valor de 300 en la wallet del apostador1.
- i) Si se cambia apostador2.deposit(500) por apostador2.deposit(400), el código imprimiría el valor de 400 en la wallet del apostador2.
- j) Los atributos de la clase Loteria que están haciendo referencia a objetos son self.value y self.apostador.
- k) Los atributos de la clase Loteria que están haciendo referencia a tipos primitivos son self.probability.
- l) Para implementar el método de clase changeProbability en la clase Loteria, la línea de código sería:

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

- m) Para llamar el método changeProbability, se haría de la siguiente manera:

```
```python
Loteria.changeProbability(0.7)
```
```

n) No, no es correcto cambiar la línea `cls.probability = nprobability` por `Loteria.probability = nprobability`. Esto se debe a que al usar `cls` se hace referencia a la clase en sí, mientras que `Loteria` sería una referencia a la instancia de la clase. El uso de `cls` es la forma correcta de acceder a los atributos de clase.

o) La clase `Loteria` tiene 5 métodos: `init_`, `payMoney`, `recieveMoney`, `playGame` y `changeProbability`.

p) Si el apostador1 gana, el apostador2 no ganaría. Cada apostador juega de forma independiente y sus resultados no afectan a los demás.

q) Si se cambia el atributo de clase `probability` a una constante, el método `changeProbability` seguiría siendo correcto, ya que permite modificar el valor de la probabilidad de forma dinámica.

r) Los métodos `gain()` y `commission()` de la clase `ComisionJuegoEspectaculos` tienen como tipo de retorno un número (`float`).

s) La variable `self` en la línea 18 de la clase `Loteria` hace referencia a la instancia de la clase `Loteria` que está ejecutando el método `playGame()`. No se puede omitir el uso de `self` en este caso, ya que es necesario para acceder a los atributos y métodos de la instancia.

t) En la línea 15 de la clase `Apostador`, el parámetro `value` pasa por valor, mientras que `self` pasa por referencia. Esto se debe a que `value` es un tipo primitivo (`int`), por lo que se pasa una copia de su valor, mientras que `self` es una referencia a la instancia de la clase `Apostador`, por lo que se pasa la referencia a la instancia.