

Taller 2 Python

Juan Diego Cardenas Acevedo

Grupo 2.

2024-2.

- a) se definen 3 clases, apostador, lotería y comisionjuegospectaculos.
- b) Esta línea verifica si el archivo está siendo ejecutado directamente. Si es así, ejecuta el bloque de código debajo de este condicional.
- c) Si se quita esa línea lo que pasaría es que, si otro archivo importa el main, se ejecutará todo el código.
- d) Se crean dos objetos de la clase apostador.
- e) apostador1 y apostador2.
- f) Ese self hace referencia al apostador, al objeto que invoque el método, de manera similar al this en java.
- g) se crean 2 al usar el método play en el main.
-Si se cambia el monto depositado de 500 a 300, no se puede realizar el método play en el main, devolviendo el mensaje: necesitas poner mas dinero en tu wallet.
- h) El valor del wallet tras jugar lotería depende de un numero generado al azar con un 50% de probabilidades de ganar, en caso de ganar, la ganancia total está definida por $400 - (400 * 20\%) = 400 - 80 = 320$. Esa ganancia se sumaría al valor que tenía antes en la wallet, por lo tanto quedaría con 700 si gana, o 100 si pierde.
-Si el apostador 2 ingresa 400 en lugar de 500, no se vería afectado el funcionamiento del código ya que apuesta 400, lo único que cambia es que de perder el valor del atributo apostador2.wallet pasaría a 0.
- i) por parte del apostador 2 pasa lo mismo que el apostador1, puede ganar o perder, en caso de ganar imprimiría 720 por las condiciones anteriormente mencionadas y en caso de perder 100.
- j) 1, el atributo apostador en la clase lotería hace referencia a objetos de la clase apostador.
- k) Value en la clase lotería hace referencia a un valor de tipo primitivo.

l) @classmethod

```
def changeProbability(cls, nprobability):
```

```
    cls.probability = nprobability
```

m) Loteria.ChangeProbability (nprobability) en el cual nprobability hace referencia al nuevo valor que se desea asignar.

n) Sí, se puede debido a que es un método estático, acceder a él desde la clase o una instancia es equivalente.

o) la clase lotería tiene 5 métodos al agregar el nuevo (contando con el método `__init__`)

p) No, el método playGame dentro de lotería asigna un valor aleatorio a una variable a, esto lo hace cada que se ejecute el método, por lo tanto ambos juegos son independientes, lo que gane el apostador1 no tiene nada que ver con lo que gane o pierda el apostador2.

q) Si se hace que la probabilidad sea una constante el método changeProbability pierde sentido porque está pensado para cambiarle el valor, aunque técnicamente lo más justo es que la probabilidad en la lotería sea siempre la misma.

r) ambos retornos son de tipo entero.

s) El self en ese caso hace referencia a la misma clase loteria para poder acceder a los atributos, y no, no se podría cambiar debido a que no accedería a los objetos de tipo loteria.

t) Value pasa por valor ya que los tipos primitivos pasan por valor en Python, Self pasa por referencia y cualquier cambio a este se verá reflejado.