

EJERCICIO 2 MODIFICACIÓN Y ANÁLISIS (POO)

Tomas Aristizabal Gomez - T.I 1033181207

Respuestas

- a) Se están definiendo 3 clases, **ComisionJuegoEspectaculos**, **Loteria** y **Apostador**
- b) Sirve para comprobar si ese archivo se está ejecutando como programa principal, en ese caso el código dentro del **if** se ejecutará
- c) Si se ejecuta como programa principal se va a seguir funcionando con normalidad ya que no habrá condicion **if** y en otro caso aunque no da error, si se importa como módulo a otro archivo, va a generar que se ejecute automáticamente el archivo main, lo cual puede no ser deseado
- d) Se crean dos objetos de la clase **Apostador**
- e) Se crea a **apostador1(1, "Juan", 302, "j@gmail.com")** ← parametros, y **apostador2(2, "Ricardo", 548, "r@gmail.com")** ← parametros
- f) **self** hace referencia a la instancia actual de la clase **Apostador**, es decir el objeto de la clase **Apostador** que está llamado al método **play**
- g) Se están creando dos objetos de la clase **Loteria**, uno en cada usuario
- h) El código imprimirá "Necesitas poner más dinero en tu wallet" haciendo referencia a que la cantidad a apostar es mayor que el dinero disponible
- i) El código imprimirá "Has ganado 720.0" ó "Has perdido lo que apostaste" según el caso que caiga en el método **playGame** de la clase **Loteria**
- j) El atributo **self.apostador** hace referencia a un objeto de la clase **Apostador**
- k) Los atributos **self.probability** y **self.value** hacen referencia a tipos primitivos
- l) El codigo quedaria asi:

```
@classmethod  
def changeProbability(cls, nprobability):  
    cls.probability = nprobability
```
- m) Se debería importar la clase **Loteria** a el **Main** y luego llamarlo así, por ejemplo:

```
from Loteria import Loteria
```



```
if __name__ == "__main__":  
    Loteria.changeProbability(0.7)
```
- n) Si, es correcto pero no es lo recomendado, es mejor usar **cls**, que hace referencia a la propia clase que está ejecutando el método, lo que hace más flexible el código a diferencia de usar **Loteria**
- o) La clase **loteria** queda con 5 metodos:
 - **__init__**
 - **payMoney:**
 - **recieveMoney:**
 - **playGame:**
 - **changeProbability**

- p) No, el hecho de que un apostador gane, no significa que ambos lo hagan. Porque son independientes, cada uno tiene su propia **wallet** y su propia **Loteria**, cada ganador tiene las mismas posibilidades pero en distintos casos
- q) Si **probability** se cambia a una constante, se daría por entendido que no se debería modificar escribiéndola en mayúscula, y el método **changeProbability** perdería todo sentido, aunque podría cambiarla igualmente esto no se debería hacer y se consideraría como malas prácticas
- r) Ambos retornan un valor de tipo **float**
- s) La variable **self** hace referencia a la instancia actual de **Loteria**. No podría omitirse, debido a que la clase **ComisionJuegoEspectaculos**, necesita acceder a los atributos de la instancia actual de **Loteria** para poder crear un objeto y calcular la comisión
- t) El parámetro **value** se pasa por valor, ya que es primitivo python pasa una copia de su valor. y el parámetro **self** que hace referencia a la instancia actual de **Apostador**, se pasa como referencia ya que es un objeto y python envía una copia de la su dirección en memoria