



Preguntas de análisis

- a) ¿Cuántas clases se están definiendo en este ejercicio?
- b) ¿Para qué sirve la línea de código `if __name__ == "__main__":`?
- c) ¿Qué sucede si retiro la línea de la pregunta anterior en nuestro código?, ¿Este sigue corriendo o hay error? Explique en ambos casos.
- d) ¿Cuántos objetos de la clase **Apostador** se están creando?
- e) ¿Cuáles objetos de la clase **Apostador** se están creando?
- f) ¿A quién está haciendo referencia la variable **self** de la línea 15 de la clase **Apostador** cuando se ejecuta el programa principal?
- g) ¿Cuántos objetos de la clase **Lotería** se están creando?
 - En la línea 4 del **main.py** cambiar el `apostador1.deposit(500)` por `apostador1.deposit(300)`
- h) ¿Qué imprimiría el código por parte del **apostador1**?
 - 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**?
- j) ¿Cuáles atributos de la clase **Lotería** están haciendo referencia a objetos?
- k) ¿Cuáles atributos de la clase **Lotería** están haciendo referencia a tipos primitivos?
- l) ¿Complete las siguientes líneas para que en la clase **Lotería**, se implemente el método de clase `changeProbability`?
 - _____
 - `def changeProbability(__, nprobability):`
 - _____`.probability = nprobability`
- m) ¿Cómo sería la línea de código para llamar el método `changeProbability`?
- n) ¿Es correcto si en el método `changeProbability` que se creó, cambiar lo siguiente?
Explique:



Línea Original

- `cls.probability = nprobability`

Línea Nueva

- `Loteria.probability = nprobability`

- o) ¿Cuántos métodos tiene la clase **Loteria** después de agregarle el nuevo método?
- p) ¿Si el **apostador1** gana el **apostador2** también? Explique por qué pasa en caso de ser sí o no
- q) ¿Qué sucede si decido cambiar el atributo de clase `probability` a una constante? ¿Se considera correcto el uso del método `changeProbability` teniendo en cuenta este nuevo cambio?
- r) ¿Cuál es el tipo de retorno de los métodos `gain()` y `commission()` de la clase **ComisionJuegoEspectaculos**?
- 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?
- 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?