

# Taller 2 Python Ejercicio 2 Andrés Jacobo Leal Aguirre

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

3 clases

b) ¿Para qué sirve la línea de código `if __name__ == "__main__":`?

Sirve para identificar en donde está el código principal, donde se le dan las instrucciones al programa

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

Teniendo en cuenta la indentación no habría ningún problema en eliminar esta línea de código y se podría ejecutar desde este archivo `main` o desde otro archivo importando al archivo `main`

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

2 objetos

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

`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?

Al objeto por el que se está ejecutando el método, se ejecuta en `apostador1` y `apostador2`

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

2 objetos

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

-- En la línea 4 del `main.py` cambiar el `apostador1.deposit(500)` por `apostador1.deposit(300)`

300

Necesitas poner más dinero en tu wallet

300

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

--En la línea 10 del `main.py` cambiar el `apostador2.deposit(500)` por `apostador2.deposit(400)`

Puede imprimir dos opciones:

400

Has ganado 720.0

720.0

---

400

Has perdido lo que apostaste

0

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

Apostador

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

Value y probability

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

-@classmethod

- def changeProbability(cls, nprobability):

-cls.probability = nprobability

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

lotería.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

El resultado del código no se verá afectado pero hay que tener cuidado con ese cambio por si en algún momento se cambia de clase o se hace una herencia

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

4 metodos

p) ¿Si el apostador1 gana el apostador2 también? Explique por qué pasa en caso de ser sí o

no

No necesariamente porque tienen cierta probabilidad de ganar y de perder sin haber relación entre ellos

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?

No, porque si lo definimos como una constante es porque en el futuro no pensamos cambiarlo

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

`float`

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?

Está haciendo referencia al objeto de tipo `loteria` y no podría omitirse porque el método necesita un parámetro de tipo `loteria`

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

`Value` pasa por valor al ser un dato de tipo primitivo y `self` pasa por referencia al ser un objeto