



PROGRAMACIÓN ORIENTADA A OBJETOS - UNALMED 2024 -2

Preguntas de análisis

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

R/hay tres clases.

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

R/sirve para escribir código que debería ejecutarse solo cuando el modulo es ejecutado directamente.

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

R/el programa dará el mismo resultado solo que se ejecutadta cada ves que se importe el bloque programa

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

R/se están creando cuatro objetos de clase apostador

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

R/id, name, phone_number, email

f) ¿A quién está haciendo referencia la variable **self** de la línea **15** de la clase **Apostador** cuando se ejecuta el programa principal?

R/se refiere a la clase apostador del objeto apostador

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

R/Se están creando dos objetos

- 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**?

R/ al contrario de lo que seria has perdido lo que apostaste dice hasta ganado 720.0

- 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**?

R/ Has perdido lo que apostaste



PROGRAMACIÓN ORIENTADA A OBJETOS - UNALMED 2024 -2

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

El atributo apostador

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

R/probability y value

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

```
- def changeProbability(self, nprobability):  
-     self.probability = nprobability
```

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

class Loteria:

probability = 0.5

def __init__(self, value, apostador):

self.value = value

self.apostador = apostador

def changeProbability(self, nprobability):

self.probability = nprobability

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
```

R/no es recomendable por que para mantener la claridad y la convención, es recomendable utilizar el enfoque con cls en un método de clase

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

R/tiene un total de cinco metodos

o) ¿Si el **apostador1** gana el **apostador2** también? Explique por qué pasa en caso de ser sí o no

R/no por que si un gana el otro no necesariamente también es es por uqe cada instancia de la clase apostador es independiente una de otra



PROGRAMACIÓN ORIENTADA A OBJETOS - UNALMED 2024 -2

- p) ¿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/su valor no podrá ser modificado después de iniciar

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

R/un tipo float

- r) ¿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?

R/hace referencia a la instancia de lotería y no se podría omitir el self

- s) ¿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?

R/value pasa por valor y self por referencia