

Taller 2 python ejercicio 2 Jhoneyker Delgado Urbina

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

R: En este ejercicio se están definiendo **3** clases.

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

R: Este sirve para poder especificar el punto de inicio del programa y allí se detallan algunas instrucciones para que el programa puede cumplir con la funcionalidad.

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: Cuando se quita solo esa línea de código, se mostrara error debido a la indentación y si se quita está el código correrá normal, esto se debe a que Python espera que el código este correctamente alineado.

Se puede ejecutar el código desde el mismo archivo `main` o importándolo en otro.

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

R: Se están creando **2** objetos de la clase `apostador`.

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

R: Se crean los objetos:

- **apostador1.**
- **apostador2.**

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

R: El `self` de la línea 15 de la clase `Apostador` hace referencia al **objeto que invoca** el método `play`, en la línea 6 del archivo `main` hace referencia a **apostador1** y en la línea 12 del archivo `main` hace referencia a **apostador2**.

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

R: se están creando **2** objetos de la clase `Loteria`.

- 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: En este caso se imprimirá por parte del `apostador1`:

300

Necesitas poner mas dinero en tu wallet

300

- 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: Imprimirá por parte del `apostador2`:

Puede imprimir: o Puede imprimir:

400

400

Has ganado 720.0

Has perdido lo que apostaste

720.0

0

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

R: En la clase `Lotería` solo el atributo **`apostador`** está haciendo referencia a un objeto.

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

R: En la clase `Lotería` los atributos que están haciendo referencia a tipos primitivos son:

- **`probability`**.
- **`value`**.

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

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

R:

```
- @classmethod  
- def changeProbability(cls, nprobability):  
-     cls.probability = nprobability
```

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

R: Lista.changeProbability(se debe ingresar el número al que se quiere cambiar)

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: Si, es correcto ya que se puede llamar al atributo probability utilizando el nombre de la clase Loteria.

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

R: Quedaria con **4** metodos la clase Loteria.

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

R: **No**, debido a que los dos objetos son independientes y cada uno tiene una probabilidad de 0,5 de ganar o perder, así que **no se garantiza** que si apostador1 gana el apostador2 tambien.

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: En Python no existen constantes pero se puede crear una variable con todas sus letras en MAYÚSCULA y el método changeProbability no se consideraría de uso correcto pero el código si se podría ejecutar.

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

R: El tipo de retorno de los métodos gain() y commission() es **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?

R: self hace referencia al objeto loteria y no se puede omitir porque mostraría error en esa línea ya que se necesita pasarle un valor u objeto para poder crear el objeto ComisionJuegoEspectaculos.

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?

R: El parámetro value se pasa por valor porque este es un parámetro solicitado en el método play y value es un número de tipo int que es la cantidad a jugar y el parámetro self por referencia ya que se pasara el objeto del que se llamo para crear el objeto Loteria.