Preguntas de análisis

a) Se están definiendo 3 clases en este ejercicio: ComisionJuegoEspectaculos, Apostador y Loteria.

b) La línea de código **`if __name__= "__main__":`** sirve para crear y usar las instancias de la clase, ya que por sí mismas estas no pueden ser ejecutadas.

c) Si se retira la línea de la pregunta anterior en el código esto genera un error (unexpected identation), al solucionarlo (eliminando la indentación de las siguientes líneas) el código seguirá funcionando.

d) Se crearon 2 objetos de la clase **`Apostador`**

e) Los objetos de la clase **`Apostador que`** se están creando son: apostador1 ("Juan") y apostador2 ("Ricardo").

f) La variable **`self`** en la línea **15** de la clase **`Apostador`** hace referencia a la instancia apostador1 que hace el llamado a la función play y es asignado como el atributo apostador de la clase Loteria.

g) Se crean dos objetos de la clase **`Loteria:`** en el método play de la clase Apostador se crea una instancia de la clase Loteria por cada jugador que llama el método play, es decir 2.

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

h) El código por parte del **apostador1** imprimirá: "Necesitas poner mas dinero en tu wallet", debido a que en ese caso el atributo wallet (300) tendrá un valor menor al de value (400), se genera ese mensaje de "error".

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

i) El código por parte del **apostador2** imprimirá: "Has ganado"... seguido de la cantidad obtenida por dicho apostador, al cambiar el valor de wallet (400) este es igual al de value (400) lo cual no genera inconveniente alguno en la ejecución del método play ni ningún otro.

j) El atributo de la clase **`Lotería`** que está haciendo referencia a un objeto es apostador, es una instancia de la clase Apostador.

k) Los atributos de la clase **`Lotería que`** están haciendo referencia a tipos primitivos son el atributo de clase probability (float) y el atributo de instancia value (int).

l) Líneas para que en la clase **Loteria**, se implemente el método de clase changeProbability:

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

m) La línea de código para llamar el método changeProbability sería:
`Loteria.probability = Loteria.changeProbability()`

n) Si, sería correcto si en el método changeProbability que se creó, cambiar la línea siguiente, porque la cls hace referencia a la clase que es Lotería, por lo que no habría ningún cambio.

Línea Original

```
- cls.probability = nprobability
```

Línea Nueva

```
- Loteria.probability = nprobability
```

o) La clase **`Loteria`** tiene 4 métodos contando con: changeProbability, payMoney, recieveMoney y playGame.

p) Si el **apostador1** gana, el **apostador2** puede ganar o no, la probabilidad para ambos es de 0.5 y es totalmente independiente al resultado del otro, la posibilidad de ganar recae directamente en un número al azar generado por la librería random.

q) Si se cambiara el atributo de clase probability a una constante eso representaría que ese valor no puede ser cambiado en la ejecución del programa, por lo cual sería incorrecto el uso del método changeProbability teniendo en cuanta este nuevo cambio.

r) El retorno de los métodos `gain()` y `commission()` de la clase **`ComisionJuegoEspectaculos`** es de tipo float, el primero retorna la ganancia obtenida por el apostador correspondiente, la segunda, por su parte es la que hace el llamado a la primera y le da el valor y el porcentaje con el cual deberá hacer los cálculos correspondientes.

s) La variable **`self`** de la línea **18** de la clase **`Loteria`** hace referencia a una instancia de la clase Loteria que se crea cuando el apostador llama al método play, que por su parte llama a playGame en el cual se llama al método ComisionJuegoEspectaculos, con lo cual no podría omitirse el uso de la variable **`self`** en este caso porque es el parámetro que se le está pasando y sin el cual no podrá ser ejecutado dicho método.

t) En la línea **15** de la clase apostador vemos como la clase recibe dos parámetros value y self pasando por valor y por referencia respectivamente, ya que value es un atributo de tipo primitivo, mientras que self hace referencia al apostador asignado (puede ser 1 o 2) que es un objeto de la clase Apostador.

PROGRAMACIÓN ORIENTADA A OBJETOS - UNALMED 2024 -2