

A. Se están definiendo 3 clases: Apostador, ComionJuegoEspectaculos y Loteria

B. Esta línea de código sirve para comprobar si el archivo se está ejecutando como un programa principal o si está siendo importado como un módulo en otro archivo

C. Si se retira esa línea, el código dentro de ese bloque se ejecutará incluso cuando el archivo sea importado como un módulo en otro archivo, el código seguiría corriendo correctamente si se ejecuta directamente, pero podría comportarse de manera inesperada si se importa en otro módulo

D. Dos objetos

E. Los objetos creados son: apostador1 con los atributos id=1, name=" Juan", pone_number=302, email=" j@gmail.com" y

Apostador2 con los atributos id=2, name=" Ricardo", pone_number=548, email=" r@gmail.com".

F. La variable self en la línea 15 de la clase Apostador hace referencia a la instancia actual del objeto Apostador que invoca el método play. Por ejemplo, cuando apostador1.play(400) se ejecuta, self se refiere a apostador1.

G. Se están creando dos objetos de la clase Loteria, uno cada vez que se llama al método play con una apuesta válida

H. Si cambiamos apostador1.deposit(500) por apostador1.deposit(300), apostador1 no tendrá suficiente dinero en la wallet para jugar con el valor de 400. Por lo tanto, la salida será:

300

Necesitas poner mas dinero en tu wallet

300

I. Si cambiamos apostador2.deposit(500) por apostador2.deposit(400), apostador2 tendrá suficiente dinero en la wallet para jugar con el valor de 400. Dependiendo del resultado del juego (ganar o perder), la salida será:

Si gana:

400

Has ganado [total]

[wallet] (wallet with added gain amount)

Si pierde:

400

Has perdido lo que apostaste

0

J. Los atributos de la clase Loteria que hacen referencia a objetos son: apostador

K. Los atributos de la clase Loteria que hacen referencia a tipos primitivos son:

1. value

2. probability (aunque es de clase, sigue siendo un tipo primitivo)

I. @classmethod

```
def changeProbability(cls, nprobability):
```

```
    cls.probability = nprobability
```

M. Para llamar al método changeProbability desde la clase Loteria, la línea de código sería:

```
Loteria.changeProbability(nueva_probabilidad)
```

N. Si, ambas líneas funcionan de manera similar en este contexto porque cls se refiere a la clase desde donde se invoca el método, que en este caso es Loteria.

O. Tiene 4 métodos: __init__, payMoney, recieveMoney, playGame, changeProbability

P. No, si apostador1 gana, eso no afecta a apostador2, cada uno juega su propia instancia de Loteria y los resultados del juego se determinan independientemente para cada instancia basada en la probabilidad.

Q. Si probability se cambia a una constante ya que sería correcto usar chageProbability porque el propósito de este método es cambiar el valor de probability.

R. El tipo de retorno de ambos métodos es float.

S. Hace referencia a la instancia actual de Loteria que invoca el método playGame y en este caso no se puede omitir el uso de self porque es necesario para acceder a los atributos y métodos de la instancia actual de la clase desde dentro de sus métodos.

T. Value es un tipo primitivo (entero) y se pasa por valor

Self es una referencia al objeto Apostador.