



CC201 Programación Orientada a Objetos

PC1-2024-02

Pregunta 1

Un cartón de Bingo consta de 5 columnas de 5 números. Las columnas están etiquetadas con las letras B, I, N, G y O. Hay 15 números que pueden aparecer debajo de cada letra. En particular, los números que pueden aparecer bajo la B van del 1 al 15, los números que pueden aparecer bajo la I van del 16 al 30, los números que pueden aparecer bajo la N van del 31 al 45, y así sucesivamente.

Escribe una función que cree un cartón de Bingo aleatorio y lo almacene en un diccionario. Las llaves serán las letras B, I, N, G y O. Los valores serán las listas de cinco números que aparecen debajo de cada letra, recuerde que en una columna no puede haber números repetidos. Escriba una segunda función que muestre el cartón de Bingo con las columnas etiquetadas correctamente. Utilice estas funciones para escribir un programa que muestre un cartón de Bingo aleatorio.

Es posible que sepa que los cartones de bingo a menudo tienen un espacio "libre" en el medio del cartón. No consideraremos el espacio libre en este ejercicio.

Pregunta 2

Escribe una clase llamada CuentaBancaria that tiene lo siguiente:

1. Un campo llamado **nombre** que almacena el nombre del titular de la cuenta.
2. Un campo llamado **cantidad** que almacena la cantidad de dinero en la cuenta.
3. Un campo llamado **tasaInteres** que almacena la tasa de interés de la cuenta (como un porcentaje).
4. Un **constructor** que simplemente establece los valores de los tres campos anteriores.
5. Un método llamado `aplica_interes()` que no acepta argumentos y aplica el interés a la cuenta. Solo debe modificar el campo de cantidad y no devolver nada. Por ejemplo, si la cuenta tiene \$1000 y la tasa de interés es del 3%, entonces la cantidad variable debe cambiarse a \$1030 (\$1000 + 3% de interés).

A continuación, pruebe la clase, mediante la creación de un nuevo objeto llamado CuentaBancaria Para un usuario nombrado Juan De Arona que tiene \$1000 al 3% de interés. A continuación, haga lo siguiente:

- Utilice el método `aplica_interes()` para aplicar los intereses a la cuenta.
- Imprima cuánto dinero hay ahora en la cuenta después de aplicar los intereses.
- Cambie la tasa de interés de la cuenta a 2%.
- Utilice el método `aplica_interes()` para volver a aplicar los intereses a la cuenta.
- Imprima cuánto dinero hay ahora en la cuenta después de aplicar el interés nuevamente.

Pregunta 3

Escriba una clase llamada `Artículo` que represente un artículo a la venta. Debe tener lo siguiente:

- Campos que representan el nombre y el precio del artículo
- Un constructor que establece esos campos,
- Un método `__str__()` que devuelve una cadena que contiene el nombre del artículo y el precio, con el precio con exactamente 2 decimales

Pruebe la clase creando un nuevo objeto artículo e imprimiéndolo.

Pregunta 4

Escriba una clase llamada `Ticket` que tenga lo siguiente:

- Un campo de costo para el precio del boleto y una hora (como dato de cadena) para la hora de inicio del evento (supongamos que las horas están en formato de 24 horas, como '18:35')
- Un constructor que establece esas variables
- Un método `__str__()` que devuelve una cadena con el formato `Ticket(<costo>, <hora>)`, donde `<costo>` y `<hora>` se sustituyen por los valores de los campos `costo` y `hora`.
- Un método llamado `esNocheHora()` que devuelve **True** o **False**, dependiendo de si la hora se encuentra en el rango de 18:00 a 23:59.
- Un método llamado `totalDescuento()` que toma un número entero n y devuelve el descuento por comprar n boletos. Debería haber un 10% de descuento por comprar de 5 a 9 boletos, y un 20% de descuento por comprar 10 o más. De lo contrario, no hay descuento. Devuelve estos porcentajes como números enteros