

CURSO DE PROGRAMACIÓN FULL-STACK

Python / Guía 8

Ejercicios

Programación Orientada a Objetos I

Ejercicios de aprendizaje

1. Crea una clase Producto con las siguientes características:

Atributos de instancia:

```
código: int
nombre: str
precio: float
stock: int (no es obligatorio. Valor predeterminado: 0)
```

Crea al menos 3 instancias de clases y muestra sus datos.

2. A la clase Producto, agrega las siguientes características:

Atributos de clase:

```
impuestos: float
```

Métodos de instancia:

```
precio_final(impuesto: float) → float
```

Crea al menos 3 instancias de clases y muestra sus datos, pero, cada vez que crees una instancia, se guardará en "precio" el cálculo que hará el método `precio_final`, que consistiría en agregar el impuesto al valor del precio.

3. A la clase Producto, agrega las siguientes características:

Atributo de clase:

```
lista_productos: list[Producto]
```

Si usas la versión 3.10, la anotación de tipo `Producto` dentro de `list`, debe estar entre comillas, si no te dará un error.

```
lista_productos: list["Producto"]
```

En la versión 3.11, simplemente podrás utilizar `Self`, con mayúsculas:

```
lista_productos: list[Self].
```



Métodos de instancia:

comprar(cantidad)

vender(cantidad)

listar_productos

Comprar: si el producto ya está en la lista de productos, aumentar el stock según la cantidad comprada. Si no lo está, agregarlo a la lista de productos. Ir notificando al usuario de los movimientos.

Vender: si el producto está en la lista de productos, quitar del stock según la cantidad vendida. Notificar al usuario si: el producto no está en la lista de productos del negocio; si intenta vender más cantidad de lo que hay en el stock; y si con tal venta se vendió la totalidad del stock. En el último caso, remover de la lista de productos del negocio.



Ver video "[Caso Práctico - Clase producto](#)"

4. Siguiendo el ejercicio anterior, crea dos clases más, con las siguientes características:

Libro

Atributos de instancia:

autor

isbn

Café

Atributos de instancia:

descripción

proveedor

Cada Libro es un Producto. Cada Café es un Producto del negocio.

Compra libros y café para el negocio.

Modifica el método "listar_productos" usando la función isinstance, para que determine si es el producto a mostrar es un libro y así mostrar el autor y isbn, por lo contrario, si es un café, mostrar la descripción y el proveedor.



Ver video "[Caso Práctico - isinstance](#)"

Siguiendo el ejercicio anterior, crea una clase Administrador: será un **mixin**. Lo utilizará la clase Producto para que cada vez que el usuario haga una compra, se requiera la contraseña de administrador para que pueda comprar, si no, no podrá hacerlo.



Ver video "[Caso Práctico - mixin](#)"

5. Crea una clase que sea la estructura de datos de una tarjeta de crédito. Crea otra clase que asocie un titular a la tarjeta de crédito. Crea una clase que simule el comportamiento de un usuario respecto a una tarjeta de crédito.
6. Utiliza la composición para crear una Bicicleta: Ruedas, Pedales, etc. Crea los métodos correspondientes para simular los comportamientos de una bicicleta.
7. Del ejercicio negocio de libros y café:
 - Guarda cada la clase base en un módulo diferente
 - Crea un paquete
 - Utiliza las anotaciones de tipo y corre MyPy
 - Crea pruebas unitarias para cada clase y corre Pytest
 - Crea un menú con funciones, para comprar o vender productos. Reutiliza el código ya usado anteriormente en el ejercicio de "artículos". Puedes intentar emplear la programación orientada a objetos para reducir la cantidad de líneas de código y la legibilidad.
 - ¡No te olvide de documentar clases, métodos y funciones!