Estructuras de datos. Curso: 2023 - 2do Cuatrimestre.

Trabajo Práctico Nº 2: P.O.O. a Python

Consideraciones de entrega

El trabajo práctico deberá ser:

Realizado en grupos de no más de 3 (tres) alumnos:
Indicando siempre en cada proyecto los integrantes del grupo.

El grupo deberá presentar soluciones para todo el práctico (no se permite hacer diferentes grupos por cada ejercicio).

Publicado en Repl.it.

Entregado antes del día 11 de Septiembre de 2023.

El práctico será evaluado con nota numérica y conceptual (Excelente, Muy Bueno, Bueno, Regular e Insuficiente), teniendo en cuenta la exactitud y claridad de las soluciones propuestas.

Los ejercicios que exijan codificación se valorarán de acuerdo a su exactitud, prolijidad (identación y otras buenas prácticas).

Las soluciones deben ser de autoría propia:

Aquellas que se detecten como idénticas entre diferentes estudiantes serán clasificadas como MAL para todos los involucrados en esta situación.

Las soluciones que se detecten son producto de alguna herramienta/servicio de inteligencia artificial serán clasificadas como MAL.

El estudiante puede agregar cualquier comentario o suposición hecha para la resolución de cada ejercicio.

Ejercicio 2

El comercio Raiden Shoes encarga a la empresa de desarrollo de software para la que usted trabaja un software que permita llevar control de las exitencias de sus productos. A tal fin se le solicita la programación de las clases que mantendrán la información en tiempo de ejecución. Deberá realizar las siguientes tareas:

Modifique las clases Calzado, CalzadoTipo, CalzadoColor y CalzadoTalle implementando:

Constructor parametrizado que inicialice cada una de las variables de instancia con un valor pasado por parámetro.

El método __eq__().

El método __str__().

El método __repr__().

Utilizando decorators programe modificadores de acceso para las propiedades self._cantidad y self._precio.

Utilizando decorators defina una propiedad sólo lectura de nombre total que devuelva el producto: self._cantidad * self._precio.

Modifique la clase Calzado de forma que arroje una excepción de tipo ValueError cuando:

Se intente establecer un valor negativo para el atributo self.sku.
Se intente establecer un valor negativo para el atributo self._cantidad.
Se intente establecer un valor negativo para el atributo self._precio.
Cree el archivo calzado_admin.py y dentro de él la clase CalzadoAdmin que implemente CalzadoAdminAbstract y los métodos abstractos en ella definidos.

Modifique CalzadoAdmin de forma que arroje una excepción de tipo ValueError cuando se intente agregar un elemento a la lista de productos que tenga igual sku que otro ya existente.

En el archivo main.py ponga a prueba los ejercicios anteriores creando al menos 5 instancias de calzado.