Programación Orientada a objetos - Recuperatorio Parcial 2

Tema 1

19-06-2025

Teoría

1) ¿Cuál es la diferencia entre usar isinstance y type para identificar objetos en una jerarquía de clases? ¿Cuál de las dos formas garantiza la reutilización de código? ¿Por qué?
2) ¿En qué consiste el mecanismo de resolución de conflictos que implementa Python para la herencia múltiple? ¿Qué problema resuelve?

Práctica

Contexto

Una empresa portuaria desea informatizar la gestión de embarcaciones y el control de los alquileres disponibles para sus clientes. Estas embarcaciones pueden ser de diferentes tipos y deben registrarse en un sistema que permita organizar, buscar y mostrar información de cada una de ellas. Las embarcaciones pueden ser alquiladas por periodos de dias, y el sistema debe calcular el costo y evitar que se intente alquilar una embarcación ya
alquilada.

Las embarcaciones con las que cuenta el Puerto son de dos tipos:

Veleros, embarcaciones de vela que se impulsan por el viento. Pueden tener entre 1 y 4 velas, lo cual influye en su velocidad y maniobrabilidad.

Lanchas: embarcaciones a motor, utilizadas para paseos rápidos o pesca deportiva. Su potencia se mide en **HP (caballos de fuerza)**, lo que afecta directamente su consumo y rendimiento.

De todas las embarcaciones se necesita registrar: nombre, eslora (largo en metros), año fabricación, costo diario, disponibilidad (por defecto True), cantidad de días, este atributo tendrá o si la embarcación está disponible.

De un velero se registra, además: cantidad de velas.

Costo diario = $200000

Costo Total (Costo diario + $20000 por vela) cantidad de dias

De una lancha se registra, además: potencia del motor.

Costo diario $300000

Costo Total: (Costo diario + $5000 por HP) cantidad de días.

El analista le solicita a usted que desarrolle una aplicación con las siguientes restricciones.

a) Definir la jerarquía de clases con los métodos correspondientes a cada clase de la narrativa dada.
b) Crear un Gestor de embarcaciones para almacenar los objetos correspondientes a ambas clases. Este Gestor debe basarse en una lista Python.
c) Implementar un programa principal con un menú de opciones que permita testear las siguientes acciones:
    1. Agregar embarcaciones al puerto (validar que el objeto que intenta agregar sea una embarcación, usar isinstance). Lanzar la excepción ValueError, si se intenta agregar una embarcación con un nombre que ya existe.
    2. Leer por teclado el nombre de una embarcación, y alquilarla. (Validar que la embarcación se encuentre disponible).
    3. Mostrar para todas las embarcaciones que se encuentren alquiladas el nombre de la embarcación y el costo total del alquiler. Este método debe ser definido en la superclase. La superclase debe incluir un método abstracto, que luego debe redefinirse en las subclases. Estas implementaciones deben garantizar la reutilización de código (NO DEBEN REPETIRSE SENTENCIAS).
    4. Para todas las embarcaciones que están disponible para alquilar, mostrar: Nombre, Costo diario, tipo de embarcación (usar type), si el tipo es un velero mostrar además la cantidad de velas que contiene, si es una lancha mostrar además la potencia del motor.