

PRUEBA PARCIAL – PROGRAMACIÓN - 3º EVALUACIÓN

SE VALORARÁ POSITIVAMENTE:
1. Los algoritmos y las estructuras de datos.
SE PENALIZARÁ:
1. La no justificación de las decisiones. La no determinación de las excepciones. Descuenta 1 punto.
2. Errores en los requisitos y los algoritmos. Un error leve descuenta 1 punto y un error grave 2 puntos.

Nombre y apellidos:

17/05/2024

Firma:

Sois los responsables de desarrollar una clase llamada **Inventario** que funcione como el sistema central para administrar las plantas en una tienda de plantas. Este sistema debe ser capaz de manejar diversas operaciones que permitan una gestión de las plantas disponibles en la tienda.

Dicho desarrollo debe cumplir con los siguientes requisitos generales:

- La clase planta tiene propiedades como tipo y algunos métodos descritos en el diagrama de clases. El método cumpleaños debe incrementar en uno la edad de la planta en una unidad de mes y el método morir debe cambiar su estado a “Muerta”, teniendo en cuenta que hay dos estados posibles “Viva” o “Muerta”. (0,5 puntos)
 - No todas las plantas son frutales pero sí lo son los melocotoneros y las albaricoqueros. Explica cómo lo resuelves en código java teniendo en cuenta el diagrama de clases. (0,5 puntos)
 - En la tienda existen varios tipos de plantas: ficus, laurel y frutales. En cuanto a frutales disponen de melocotoneros y albaricoqueros, ¿tiene sentido que un cliente compre como planta un frutal? ¿o tiene más sentido que compre específicamente un melocotonero o un albaricoquero? Justifica tu respuesta en el código. (0,5 puntos)
 - Las plantas deben almacenarse en algún tipo de estructura de datos dentro de la clase Inventario. (0,25 puntos)
 - ¿Cuál es el cometido del método **toString()**? ¿Qué sucede exactamente si no se implementa? (0,25 puntos)
 - Para permitir un control del stock deberá implementarse un contador (a modo de auto-incremento) que se utilizará para almacenar el número de plantas que va adquiriendo la tienda, es decir, con cada nueva instancia. Encapsula esta acción en el método **incrementaContador**. (0,5 puntos)
 - Es obligatorio implementar **las demás clases** necesarias para la clase Inventario y son la base del enunciado. Sus métodos y atributos son triviales pero no olvidéis justificar su implementación o existencia. Es decir, tieneInjerto, color, darFruto(), tieneFruto, etc. (1 punto)
 - La clase inventario **debe realizar** las siguientes acciones en la clase principal llamada **ProgramaPrincipal**:
 - Mostrar la lista de plantas. En este caso únicamente el tipo y nombre. (1 punto)
 - Mostrar todos los datos de una planta en concreto. (1 punto)
 - Mostrar todos los datos de todas las plantas. En este caso independientemente a su tipo. (1,5 puntos)
 - Insertar plantas en el inventario. (1 punto)
 - Eliminar plantas del inventario. (1 punto)
 - Vaciar el inventario. (1 punto)
- ◆ Opcionalmente podéis contestar a esta pregunta: ¿es posible hacer uso de algún enumerado en alguna de las clases de este desarrollo? ¿es posible hacer uso del operador *instanceof* en este desarrollo? Implementad, si es posible, ambos escenarios. (0,5 + 0,5 puntos extra)

Recordad que deberéis tomar y justificar las decisiones durante el desarrollo, por ejemplo qué visibilidad y/o modificador de acceso le dais a un determinado método o clase o la creación de constructores, getters o setters, etc. Es por eso que el diagrama de la clases UML presentado a continuación, no contiene más nivel de detalle.

