

Materia:	Programación II		
Nivel:	2º Cuatrimestre		
Tipo de Examen:	Primer Parcial		
Apellido ⁽¹⁾ :		Fecha:	25 oct 2024
Nombre/s ⁽¹⁾ :		Docente a cargo ⁽²⁾ :	Baus / Bustos
División ⁽¹⁾ :	321-2	Nota ⁽²⁾ :	
DNI ⁽¹⁾ :		Firma ⁽²⁾ :	

(1) Campos a completar solo por el estudiante en caso de imprimir este enunciado en papel.

(2) Campos a completar solo por el docente en caso de imprimir este enunciado en papel.

Sistema Registro de Plantas en un Jardín Botánico

Nos encargan implementar un sistema para gestionar las plantas de un jardín botánico.

En el jardín se registran diferentes tipos de plantas, como árboles, arbustos y flores. Cada planta cuenta con un nombre, su ubicación en el jardín y el clima en que prospera. Los árboles, además, tienen una altura máxima (en metros), los arbustos tienen la densidad de su follaje (medida de 1 a 10), y las flores tienen una temporada de florecimiento (PRIMAVERA, VERANO, OTOÑO, INVIERNO).

Tanto los árboles como los arbustos deben poder ser podados llamando a su método podar().

Funcionalidades requeridas:

- **agregarPlanta(Planta planta):** El botánico debe poder agregar cualquier tipo de planta al sistema. Se deberá lanzar una excepción personalizada si ya existe una planta con el mismo nombre y ubicación en el jardín.
- **mostrarPlantas():** Muestra todas las plantas registradas en el jardín, indicando: nombre, ubicación, clima, y los atributos específicos de cada tipo de planta (altura máxima, densidad del follaje o temporada de florecimiento).
- **podarPlantas():** Permite podar todas las plantas que pueden ser podadas (árboles y arbustos), e informa sobre las plantas que no pueden ser podadas (flores).

A partir del enunciado anterior, se solicita:

1. Realizar el diagrama de clases completo en umletino, mostrando la relación entre Planta, Arbol, Arbusto y Flor y demás.
2. Implementar el código fuente en Java que resuelva las funcionalidades solicitadas.

Detalles adicionales:

1. **Diagrama de Clases:**
 - Debe reflejar la jerarquía de clases y relación entre plantas.
 - Los atributos comunes deben estar en la clase base Planta y los específicos en las clases derivadas.

2. Clases y Herencia:

- Debe haber una clase abstracta Planta que contenga los atributos comunes: nombre, ubicación y clima.
- Las clases Arbol, Arbusto y Flor deben heredar de Planta.
- Los árboles deben tener el atributo alturaMaxima (en metros), los arbustos el atributo densidadFollaje (de 1 a 10), y las flores el atributo temporadaFlorecimiento (un enum con las estaciones).

3. Métodos:

- El método agregarPlanta() debe añadir plantas a una colección y lanzar una excepción si la planta ya existe en esa ubicación.
- El método mostrarPlantas() debe imprimir una lista de todas las plantas y sus atributos.
- El método podarPlantas() debe permitir podar árboles y arbustos, mientras que informa que las flores no pueden podarse.

Criterios de corrección:

1. Aprobación:

- El diagrama de clases debe reflejar correctamente la jerarquía de herencia.
- Las clases y atributos deben estar correctamente implementados.
- Los métodos solicitados deben funcionar adecuadamente.
- El manejo básico de excepciones debe estar presente, especialmente al agregar plantas duplicadas.

2. Promoción:

- El diagrama debe incluir detalles como visibilidad de atributos y métodos, así como relaciones adicionales (por ejemplo, composición).
- El código debe seguir principios de POO, con buen manejo de encapsulamiento y visibilidad de atributos.
- Se espera el uso de un enum para la temporada de florecimiento de las flores.
- El manejo de excepciones debe ser avanzado, con excepciones personalizadas y mensajes claros.
- Se valora el uso de colecciones eficientes.
- Buena modularización del código.

Ejemplos de escenarios:

1. Agregar plantas al sistema:

- Se debe poder agregar un árbol llamado "Roble" en la zona norte del jardín, y al intentar agregar otra planta con el mismo nombre y ubicación, se debe lanzar una excepción.

2. Mostrar plantas registradas:

- El sistema debe listar todas las plantas, mostrando tanto los atributos comunes (nombre, ubicación, clima) como los específicos (altura máxima, densidad del follaje o temporada de florecimiento).

3. Poda de plantas:

- El sistema debe permitir que los árboles y arbustos sean podados, mientras que debe indicar que las flores no requieren poda.